FINITE ELEMENT MODELING OF

STREAMFLOW ROUTING

By

IFEANYI LAWRENCE NWAOGAZIE

Bachelor of Science
in Civil Engineering
University of Kansas
Lawrence, Kansas
1979

Master of Science
University of Kansas
Lawrence, Kansas
1980

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
July, 1982

Dedicated to members of my
family, loved ones, and to all those who
appreciate the progress of mankind!

FINITE ELEMENT MODELING OF

STREAMFLOW ROUTING

Thesis Approved:

Avdhesh K. Tyagi

Thesis Adviser

J A Harvey

R. N. DeVries

J P Chandler

Norman N Durham

Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to express my sincere indebtedness to Dr. Avdhesh K. Tyagi, my major adviser, for his support, guidance and constant encouragement throughout this study.

My deep appreciation is also extended to Dr. Richard N. DeVries, Dr. John W. Harvey, and Dr. John P. Chandler, members of the advisory committee, for their invaluable suggestions and assistance in reviewing this manuscript.

A note of thanks is extended to Ralph Hight of the Corps of Engineers, Tulsa and Mr. Phil Weigant of the National Weather Service, River Forecast Section, Tulsa, for their assistance in providinq data on Illinois River, Oklahoma.

Special thanks are extended to Mrs. Cynthia Kesselhuth and Mrs. Oneta Porter of the Civil Engineering school for their assistance in typing this manuscript.

Finally, to my parents in Nigeria for their continuous affectionate concern and encouragement, the author cannot thank them sufficiently and to my colleagues, Mr. M. Hoque, Mr. D. Tee, Mr. S. Sharma, Mr. J. Okere, and many others who have made my working environment a pleasant one.

This research was supported, in part, by the Office of Water Research and Technology, Matching Grant No. B-61-Kansas at the University of Kansas and Allotment Grant No. A-103-OKLA at the Oklahoma State University and the Dean's Research Incentive, College of Engineering at the Oklahoma State University.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

Background

The process of determining the water depths, velocities, and discharges in the channels, rivers, or reservoirs under unsteady conditions arising from flood motions is commonly referred to as flood routing. Interest in flood routing and in part in unsteady flows in water resources stems from the need to plan, design, regulate, and manage our flood prone areas and many other water resource systems.

In a surface water system, runoff, floods, droughts, and stream water quality interact very closely. When excess rainfall occurs over an area, runoff contributes to flooding along rivers whereas drought occurrence due to lack of rainfall results in minimal streamflow, reduced water supply, and less naviagation. In addition, the water quality of the stream becomes poor because of the low flows of the streams. Thus, the occurrence of lack of streamflow or drought affects the management of surface waters in the stream.

The State of Oklahoma experiences runoff ranging from 0.2 inches in the Panhandle to 20 inches in the southeast corner, which reflects the dramatic contrast in precipitation. In the northwestern region an average runoff amounts to about 820,000 acre-feet per year compared to 6,000,000 acre-feet per year in the southeastern region. Annual average runoff for the entire state is approximately 22,000,000 acre-feet (Oklahoma Water Resources Board, 1980).

Flooding has been experienced over the years in Oklahoma. The Water Resources Council estimates that without increased flood management programs average annual flood damages will increase from $2.3 billion in 1975 to $3.6 billion in the year 2000. These damages occur over a flood plain of some 140 to 180 million acres. The Arkansas River Basin and the Red River Basin experienced an estimated $167,000,000 in flood damages in the state between 1955 and 1975, with the majority of that attributed to the Arkansas River (Oklahoma Water Resources Board, 1980).

Some floods occur gradually, as when prolonged steady rainfall saturates a river basin until most of it runs off, creating a greater volume of water than the natural channels and drainage structures can carry. Others are the result of sudden heavy rains occurring in a short time from thunderstorms. The latter is usually experienced in Oklahoma. In either case, floods are considered a problem only when the result is widely spread damage to agriculture and structures or when the normal activities of man are seriously interrupted.

Like other Great Plains States, Oklahoma has scores of extended droughts on an approximately 20-year cycle (Oklahoma Water Resources Board, 1980). An analysis of drought conditions from 1931 to 1971 indicates that drought occurred somewhere in the state about 51% of the time, more frequently in the panhandle and less frequently in northeastern and southcentral areas.

Water quality of Oklahoma's streams is adversely affected by natural and man-made pollution. In the west, natural salt springs and salt flats emit into local streams large quantities of chlorides that are subsequently carried downstream, polluting other major streams as they pass. In central and eastern Oklahoma, municipal and industrial effluents degrade many streams, restricting their beneficial use.

Thus, the interrelationships among runoff from rainfall, floods, droughts, and stream quality are predicted only when mathematical models to simulate the depth of flow and discharge in a stream resulting from rainfall are available.

Study Objectives

The purpose of this study is to evaluate the discharge in the streams under the varying conditions of rainfall. The results of the mathematical models developed using the finite element methods will predict the depth of flow, velocity of flow, and the discharge in the streams.

Three mathematical flow models have been developed in this study. The first two are approximate models while the third is a complete model. They are presented below in the order of increasing complexity:

1. The kinematic flow model, KFM, solved explicitly and implicitly by Galerkin's weighted residual finite element method. The implicit version is implemented using a time weighting factor, and the resulting non-linear system of a tridiagonal matrix equation is solved iteratively by the generalized Newton-Raphson method.
2. The diffusion flow model, DFM, implemented similarly to the implicit kinematic flow model, except that the resulting non-linear system is a bi-tridiagonal matrix equation. Solution is obtained by the Newton-Raphson

technique.

3. The complete flow model, CFM, produces a matrix equation similar to the implicit diffusion flow model and is solved using the same technique.

Model performance is evaluated using two forms of channel geometries. The first is comprised of an artificial stream channel of constant geometry with a hypothetical flood hydrograph imposed at the upstream end of the reach. Simulated flow is compared with the Viessman's solution using the explicit finite difference scheme. The second model test involves flow in the Illinois River, a natural river in Oklahoma. The Illinois River, a tributary of the Arkansas River, originates in northwestern Arkansas as Osage Creek and flows westward into Oklahoma. The flood recorded on April 10, 1979, for Watts and Tahlequah guaging stations, 50.4 miles apart, and that for Flint Creek, a tributary approximately 13.2 miles downstream of the Watts Station are utilized.

The choice of the natural channel is limited due to lack of adequate hydraulic data. Thought the Illinois River seems to exemplify varying channel geometric and hydraulic properties inherent in many other natural rivers in Oklahoma, the availability of data and the excellent flood hydrographs of 1979 record make it the best choice.

The objective of the first test with an idealized river channel is to explore the basic principles and to make some appraisal of the sensitivity of the controlling flow parameters in the mathematical models. The model application to a natural channel checks on the capability of simulating natural floods of long durations for use in the design of hydraulic structures as well as for the flood plain zoning.

CHAPTER II

LITERATURE REVIEW

Hydraulic And Hydrologic Routing Methods

Significant studies of unsteady flow in an open channel date back to the early works of the French mathematicians, Laplace (1775-76) and Lagrange (1783). The Lagrange celerity formula for small waves in shallow water provided the first impetus for subsequent studies. Later the British School of Mathematical Physicists gave some attention to fluid flow problems with contributions being made by Stokes, Kelvin, Rayleigh and Lamb (Water Waves, 1965).

The more advanced mathematical treatment of unsteady flow in an open channel is credited to Barré de Saint Venant (1871), a French mathematician who developed the complete one-dimensional equations of unsteady flow. These are two nonlinear hyperbolic partial differential equations of motion (conservation of mass and conservation of momentum) that very accuately describe the gradually varied flows in open channels. The original form of these equations is:

$$B \frac{\partial y}{\partial t} + A \frac{\partial v}{\partial x} + v \frac{\partial A}{\partial x} = 0 \qquad (2.1)$$

$$\frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} + g(\frac{\partial y}{\partial x} + S_f - S_o) = 0 \qquad (2.2)$$

Where:

A = channel cross-sectional area, ft^2;

B = width of channel water surface, ft;

y = depth of flow, ft;

x = distance along the channel, ft;

t = time, sec;

S_f = friction slope, ft/ft;

v = mean velocity across the section, ft/sec;

S_o = longitudinal bottom channel slope, ft/ft

Two basic techniques for unsteady flow simulation are (1) methods which approximate a solution to the basic equations of unsteady flow (Eq. 2.1 and 2.2), and (2) methods which solve the basic equations. The first methods are sometimes referred to as "hydrologic" routing methods and the second kind, "hydraulic" routing methods (Thomas, 1975).

The importance of the hydraulic routing method has become increasing evident in the light of the modern high-speed digital computer for solutions of unsteady partial differential equations that have no closed form or analytical solution. Numerous unsteady flow phenomena such as surges, effects of tidal fluctuations, backwater resulting from channel junctions or reservoirs, and normal flood waves from excessive rainfall can be analyzed using the hydraulic routing and numerical methods such as finiteelement or finite difference methods.

On the other hand, the early development of the hydrologic routing in the form of a continuity equation is credited to Rippl (1883). In working on reservoir capacity problems, he utilized the concept of successive approximations to routing streams where the data are average daily flows, rather than slope, stage, and velocity measurements. Hydrologic routing is handy when data for hydraulic routing are not available.

The associated continuity equation is:

$$I - 0 = \frac{dS}{dt} \qquad (2.3a)$$

or

$$\tfrac{1}{2}(I_1 + I_2)\ \Delta t - \tfrac{1}{2}(0_1 + 0_2)\ \Delta t = S_2 - S_1 \qquad (2.3b)$$

where, I, 0, t, $(S_2 - S_1)$ are the inflow into a given reach, outflow from that reach, time period for the flow to travel through that reach, and the change in storage during that time period in the reach, respectively. The subscripts 1 and 2 represent conditions at the beginning and end of the routing periods.

The hydrologic routing is sometimes referred to as hydrograph routing because of the graphical relationship established between storage and outflow yields a feasible solution to Equation (2.3b) having two unknowns, 0_2 and S_2. Puls (1928) established a curve of relation between inflow and outflow versus storage for a variety of flood on the Tennessee River.

With some modifications of the continuity Equation (2.3b) to include the local inflows along the channel, Wisler and Brater (1931) presented a revised graphical scheme of Puls. This method was the first to use computed inflow hydrographs from tributaries and unmeasured areas for which no flow records are available. A number of hydrologic routing methods have emerged over the years including various coefficient routing procedures such as the Muskingum technique (McCarthy, 1938). Interested readers are referred to basic texts on hydrology (Chow, 1964; Viessman, 1972).

Numerical Methods in Hydraulic Routing

Finite Difference Methods

A large number of schemes have evolved from the finite difference methods over the years and have been applied with success to equations of unsteady flows and other engineering problems. For instance, the explicit, characteristic, and implicit schemes are the major categories. However, varieties of each group exist, for example the leap frog, diffusion and staggered explicit schemes, method of characteristics with fixed or characteristic grids, and the implicit scheme with weighted four-point or six point (Gunaratnan, 1970; Thomas, 1975; Fread, 1976).

A survey of previous literature indicates that many investigators to date have employed the finite difference schemes in flood routing problems. Isaacson et al. (1954, 1956) investigated flood routing in their peoneering work in the Ohio River. Amein (1966) used the method of characteristics to solve the streamflow problem in an attempt to study the effects of friction on peak flows. Amein and Fang (1969) also used an implicit scheme in solving the streamflow routing problem in natural channels in North Carolina. Pinder and Sauer (1971) employed the explicit method in simulating the flood wave modification due to bank storage effects. Fread (1971, 1973, 1974, 1976, 1978) investigated the routing problems using the implicit four-point and wieghted four-point finite difference schemes. Chaudhry and Contractor (1973), Liggett and Wollhiser (1967), Viessman et al. (1972), and many others have in turn used finite difference methods to solve approximate and complete routing equations.

It is interesting to remark that some of the finite difference schemes have some limitations often associated with convergence and stability problems. The explicit method is subject to a stringent stability condition imposing a limiting value for the time step in relation to distance step (Amein and Fang, 1969). The maximum time step that can be used in the explicit scheme to insure numerical stability when frictional effects are relatively small is computed using the Courant condition (Fread, 1973) as:

$$\Delta t_c \leq \frac{\Delta X_i}{(|v_i| + (g^{A}i/B_i)^{\frac{1}{2}}} \qquad (2.4)$$

where:

A_i and B_i = area of flow and width of the water surface in the i^{th} cross-section, respectively;

ΔX_i = the i^{th} distance step;

Δt_c = the computational time step;

$(A/B)_i$ = the hydraulic depth;

v_i = velocity of flow in the i^{th} cross-section.

Although the explicit scheme would not pose much difficulty for investigation of short time flows, it becomes cumbersome and inefficient for large flood flows in large rivers.

The method of characteristics is highly suitable for rapidly varied flows (Amein, 1966). It can be used for flood studies. However, the scheme is inconvenient in that the results are not obtained at fixed times and locations. A modification of the scheme employing a fixed mesh has been applied by Baltzer and Lai (1968) to tidal flows, but it has no significant advantage over the explicit method for large river flows.

One requirement for the explicit scheme and method of characteristics is the use of equal distance intervals. This appears disadvantageous for rivers with irregular geometry (Fread, 1974). Thus, the development of the implicit schemes arise not only toovercome the equal distance requirement but also as a means of negating the restriction of small time steps imposed on the explicit and characteristic methods for reasons of stavility. The four-point implicit finite difference method appears most advantageous since it can readily be used with unequal distance intervals (Fread, 1973, 1974, 1976).

In the light of the inherent advantages of the implicit four-point scheme, Fread (1973) investigated the influence of the time weighting factor, θ, for spatial variables along with those of the channel parameters, such as the length of the reach, bed slope, roughness coefficient, and surface width, on the numerical distortion (dispersion and attenuation of computed stage hydrographs). The definition of θ is presented in Figure 1. Among other things, the following observations were made. The lower range of allowable θ values minimizes the distortion which results from the use of large time steps in the integration of the implicit difference equation. A value of $\theta = 0.55$ was chosen to minimize distortion while conservatively insuring theoretical stability criteria. The tendency for the stability of the numerical computations to decrease with increasing value of θ exists.

On the other hand, numerical distortion increases when the channel length, L, or the Manning roughness factor, n, increases; and it decreases when the magnitude of the initial depth of flow, y_0, or the

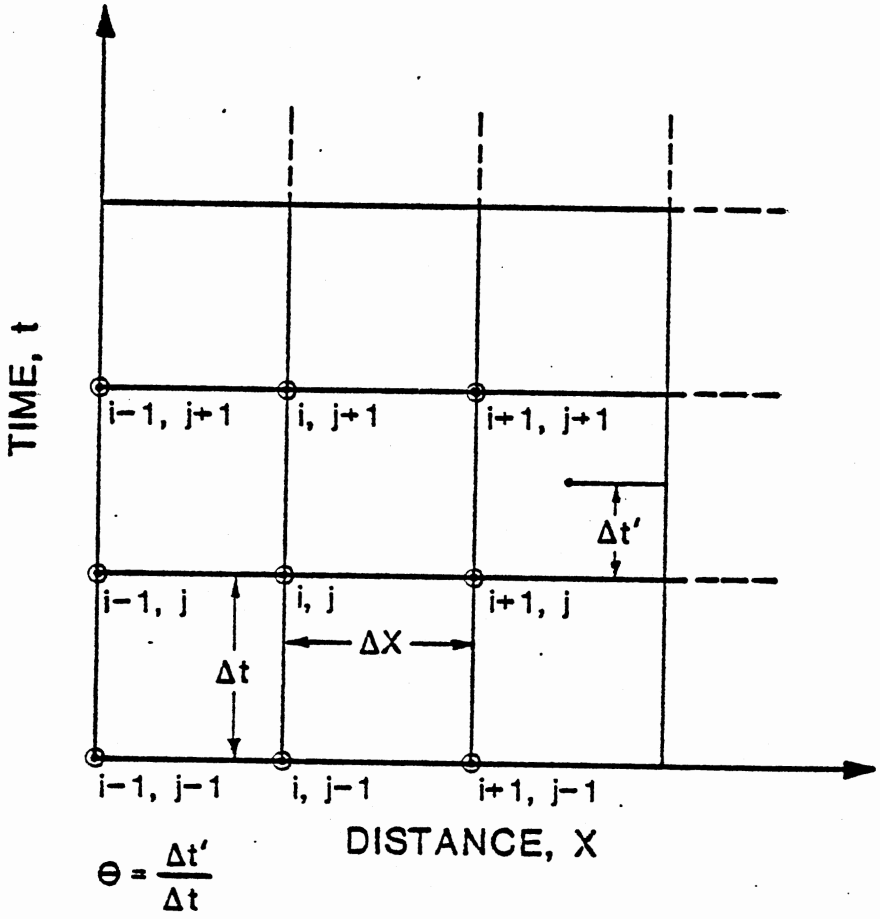


Figure 1. Definition of the Dimensionless Time Weighting Factor, θ.

channel bottom slope, S_0, increases. The channel width, B, was observed to have little or no effect on the magnitude of the numerical distortion.

In general, it is expected that the results obtained by the implicit method would be no different than those obtained by other numerical methods for the solution of the complete equation of unsteady flow in open channels. The main difference is that the implicit method provides the result faster (Amein and Fang, 1969).

Finite Element Methods

Evolution and Extension to Fluid Dynamics

The evolution of the present-day finite element methods has followed a long but imprecise history (Zienkiewicz, 1977). To date, the unified efforts of the early mathematicians and those of engineers mostly in the structural discipline have given rise to a complete picture of finite element methods. The contributions of the mathematicians are seen in the area of formula development (the governing differential equations) for the physical problems and solution techniques such as the variational principle, Gurtin principle, and the weighted residual principles.

On the other hand, the engineers tend to approach the problem by establishing a direct analogy between the real discrete element and finite portions of a continuum domain. As Zienkiewicz (1977) puts it, it is from this "direct analogy" view that the term finite element was born. The existence of a unified treatment of the "standard discrete problems" leads to the first definition of the finite element process as the method of approximation to continuum problems such as:

1. The continuum is divided into a finite number of elements whose behavior is specified by a finite number of parameters, and

2. The solution of the complete system as an assembly of its elements follows precisely the same rules as those applicable to standard discrete problems (p. 3).

For the simple reason that a number of classical mathematical procedures of approximation fall into this category as well as the various direct approximations in engineering, Zienkiewicz (1977) states that the origin of the finite element procedures and the precise moment of its invention are difficult to determine. A supporting point of view is held by Oden (1972) who comments on the piecewise approximations and rudiments of the idea of interpolation supposedly used in ancient Babylonia and Egypt that preceded the calculus over 2000 years ago.

More recently, the practice of representing a structural system by a collection of discrete elements was utilized in the early works of the aircraft structural engineers (Courant, 1943). The formal presentation of the finite element methods together with the direct stiffness method for assembling elements is attributed to Turner, Clough, Martin, and Topp (1956). It was Clough (1960) who first used the term "finite elements" in a later paper devoted to plane elasticity problems.

The application of the finite element method to fluid flows began to assume a degree of importance in the mid-sixties following the early works of Zienkiewicz et al. (1965, 1966), Javandel and Witherspoon (1968), and Tyagi (1971) in porous media flow. For the last decade scores of papers have emerged applying the finite element methods to surface water systems for estuaries, reservoirs and streams, and groundwater systems for flow in saturated and unsaturated zones and groundwater quality (Gallagher, et al., 1974, 1976; Gray, Pinder, and Brebbia, 1976; Ciriani, Maione and Wallis, 1974; Tyagi, 1975a, 1975b, 1975c).

Weighted Residuals Methods, WRM

Elaborate discussion on the basic finite element schemes is found in literature (Zienkiewicz, 1977; Oden, 1972; Finlayson, 1972; Segerlind, 1976; Ames, 1977; Chung, 1978). Norrie and DeVries (1975) presented a bibliography covering over 3800 citations during 1956-1974.

As an approximate method of solving differential equations of initial and/or boundary value problems in engineering and mathematical physics, the finite element can be implemented via variational principle or weighted residual principles. The variational principle is based on the works of Rayliegh (1877) and Ritz (1909). Some classes of problems can not easily be put into variational form, particularly when the governing differential equations are not self-adjoint. Thus, this method has limited application.

The weighted residual methods (WRM), which include the orthogonal collocation, Bubnov-Galerkin, Subdomain, and least-squares, are employed to deal directly with the governing equations of the physical problems (Finlayson, 1972; Ames, 1977). The weighted residuals in general utilize the concept of orthogonal projections of a residual of a differential equation onto a subspace spanned by certain weighting function. Stated differently, the unknown solution in all the WRM is approximated by a set of local basis functions containing adjustable constants or functions (Ames, 1977). These constants or functions are chosen by various criteria to give the "best" approximation for the selected family. For instance, the least-squares method requires higher order interpolation functions in general, even if the physical behavior may be adequately described by lower order (linear) functions (Chung, 1978). This restriction limits its use. The collocation is the simplest (WRM) to apply, but it has a drawback in terms of the number of nodes needed to achieve the same results as with the Galerkin method.

Of special interest in all the WRM is the Bubnov-Galerkin method (1913 and 1915, respectively). The method (often referred to as Galerkin without Bubnov) is the most popular and widely used. Large numbers of non-linear fluid flow systems are easily transformed into "finite element equations" directly. The classical procedures of the Galerkin assume the weighting function and the trial function to be identical (Zienkiewicz, 1977). Like the variational principle, the Galerkin always yields a symmetric matrix equation for linear differential operators.

Approximation of Time Derivatives

The concept of extending the finite element to include the time domain is discussed by Oden (1969, 1972) and Chung (1978). The approach is to regard the basis function as being dependent on time as well as the spatial domain such that:

$$\frac{\partial v(x,t)}{\partial t} = \frac{\partial N_i(x,t)}{\partial t} V_i \qquad (2.5)$$

where:

$v(x,t)$ = dependent variable, v, expressed as a function of space, x, and time, t;

$N_i(x,t)$ = basis function at node, i, as a function of space and time.

Other investigators have extended this idea in many studies in water resources*. Zienkiewicz and Lewis (1973) investigated two linear finite element formulations in the time domain. Grotkop (1973) applied the Galerkin method in the time domain to estuary modeling; Gray and Pinder (1974) conducted a numerical experiment on the use of the Galerkin finite element method to approximate both the time and space derivatives and thereby study the suitability of using higher order basis functions in the time domain for solving the transient groundwater flow equation. Van Genuchten (1977) employed higher basis function (including Hermitean) to a one-dimensional solute transport equation and studied the accuracy of the resulting schemes.

One major disadvantage of finite element approximations in time derivatives is the enormous increase in the computational time and effort. Gray and Pinder (1974) noted the inherent tradeoff between increased accuracy and decreased computational efficiency associated with the finite element time derivatives. The optimum scheme for approximating the time derivative in a groundwater flow problem is dependent on both the behavior of the solution and the method of time step selection.

The second approach to the time derivative approximation is the so called "semidiscrete method" in which the time derivative of a variable at nodes is replaced by a temporal operator (finite difference operator) from the relation, (Chung, 1978):

$$\frac{\partial v(x,t)}{\partial t} = \dot{v}_i(x,t) = N_i(x)\dot{v}_i(t) \qquad (2.6)$$

where:

$\dot{v}_i(t)$ = time derivative of v prescribed at node i

*The author was not able to discover any documentation of FEM in time derivatives applied to unsteady flows in open channel modeling.

Many of the finite element models of transient problems adopt the previously described semidiscrete method. When the time derivatives are approximated with finite differences, either a central in time (Crank-Nicolson) or a backward/forward in time (implicit) scheme can be used (Van Genuchten, 1977). The former results in a second-order accuracy while the latter yields only a first-order accuracy. If higher order basis functions are used, it may be important to obtain a higher order approximation of the time derivatives. This might not be necessary if lower basis functions such as the linear types are employed.

Numerical Properties of FEM for Non-Linear Systems

Numerical properties of the finite element method, such as the stability, convergence, and accuracy, unlike those of the finite difference methods have not been established adequately although many intuitive proofs and conclusions have been stated (Desai and Christian, 1977). The study of the numerical procedures has often been made in a pragmatic manner. When a given scheme is used for a number of problems and it is found satisfactory, it is considered acceptable. The major criticism of this approach is that it may not yield a general scheme (Desai and Christian, 1977).

Error analysis associated with the solution of the non-linear hyperbolic open channel flow equations may be grouped as: finite element approximation errors; temporal approximation errors; and errors due to any iterative non-linear equation solver, such as the Newton-Raphson, predictor-Corrector, and others. At present, no theoretical finite element error estimates are available for the unsteady nonlinear two-variable equations. However, it is possible to perform error analysis due to temporal operators, together with the iterative equation solver in a restricted sense. Adopting a procedure described by Chung (1978, pp. 227), by holding the non-linear terms constant during the iteration cycle, it is then possible to generate an approximate amplification property matrix, a technique reported by Lax-Richtmyer (1956) and Richtmyer-Morton (1967). Every eigenvalue λ_i of the amplification matrix, if made smaller than unity, automatically insures stability. The largest eigenvalue, called the spectral radius of the amplification matrix which governs the stability, and the limiting value of the time step, Δt, can be determined. As the non-linear terms are updated, the amplification changes, thus altering the stability criteria as calculations progress. This increases the difficulty in the stability analysis of an unsteady non-linear system.

A slightly different approach to the error analysis for non-linear hyperbolic equations (Oden and Fost, 1973) requires the finite element basis functions to satisfy the convergence and completeness criteria as for linear elliptic problems. The study yields a stability estimate that is considered to be consistent with the well-known Von Neumann linear stability criterion which requires the discrete system to propagate information at a rate greater than or equal to the speed of propagation of the actual system. The above approach is too narrow in concept, a linearization technique drawn from elliptic type of problems, and limits the use to the special class of hyperbolic equation studied, the one-dimensional homogeneous hyperelastic bodies.

With regard to the finite difference method, Fread (1974) studied the numerical properties of the St. Venant equations for a four-point implicit scheme using the Von-Neumann technique. Since this technique is only applicable to linear differential equations, linearization of the governing equations is adopted with certain terms omitted on the basis of their relatively small magnitude in order to facilitate the stability analysis. On the other hand, the convergence criterion was analyzed by expanding each term in the Taylor series expansion about the point at which the differential equation is computed. The study concludes that the implicit four-point method is unconditionally stable provided the time weighting factor, $\theta \geq 0.5$, and has a second order accuracy since time step, Δt, and distance step, Δx, are quadratic.

Cooley and Moin (1976) studied the numerical properties of the St. Venant equations using the finite element method* and the predictor-corrector iterative solving scheme. They adopted the linearization technique similar to those used by Strelkoff (1970) for stability analysis. Their concluding remarks are identical to those of Fread (1974). Error analyses for other classes of differential equations are reported in literature (Kreig and Key, 1971; Fujii, 1972; Desai, Oden and Johnson, 1975; Desai and Lytton, 1975; and Chung, 1978).

Finite Element Versus Finite Difference

The purpose of resorting to the Numerical Methods is to be able to solve problems either for which there is no analytical solution or for which the analytical solution is too hard to obtain. For the last two decades, attention has been drifting from the finite difference method

* The author observed very astonishingly, the constant use of finite differencing for time derivatives and finite element for space derivatives (mostly linear basis functions). This seems to explain the formidable difficulties associated with a complete finite element error analysis of unsteady non-linear hyperbolic equations.

to finite element method in hydrology and water resources. The search for the most, efficient and accurate simulation model has continued to be the center of inspiration for this change of attention. The question often raised is "Is there any real reason for this change of attention?" Review of major studies in fluid dynamics involving the FDM and FEM sheds some light on answering the question.

In a very vigorous classification of trial functions, Zienkiewics (1974) states that within a broad definition, the finite difference technique falls into a "subclass" of the general finite element methodology, which indeed embraces many other classical approximation procedures. Nevertheless, both techniques can be considered as distinct in a much narrower perspective. For instance, the two methods differ in a manner in which the element equations are generated from the governing equation. While the two adopt the principle of discretization as the initial step in the numerical procedure, the way the concept of discretization is implemented varies. In the FDM, the governing equation is discretized whereas in the FEM, the region or continuum of the system is discretized. In other words, in the FEM the problem is formulated as an integral to be minimized, and we use a numerical approximation of the integral to obtain a solution. This step is necessary regardless of the kind of FEM adopted --variational principle or method of weighted residuals (Myers, 1971).

Another distinguishing feature of the FEM is the difference in the grid and element numbering system. For instance, a typical element, e, is the interval between nodal points, i and j. This numbering scheme is slightly different from the FDM where nodal-point number is also used to designate the region surrounding the nodal points. In the FEM, the numbering of the nodal points is entirely separate from the numbering of the elements.

Though it has not been proved technically that any one method is superior to the other, what seems obvious is that the finite element method may prove more advantageous for some classes of problems--those with extremely complex geometry--than FDM. A supporting viewpoint can be drawn from Myers (1971). In a one-dimensional steady state heat transfer problem for a thin rod, solution is sought by the variational principle (FEM) and FDM using the same number of nodal points. It was observed that the FEM solution falls below the exact nodal values by about the same amount that the FEM are above the exact values. This was explained by the fact that FEM was generated by minimizing the integral (Myers, 1971).

Some numerical studies performed by Pinder and Gray (1976) using the equation governing the convective -diffusion transport of a conservative contaminant help to illustrate the relationship between FEM and FDM. By using the Galerkin approximation of the space derivatives and the finite difference approximation of the time derivatives, they observed that the FEM can be expressed in terms of weighted average finite difference approximations. However, this observation had been reproted by others earlier (Myers, 1971; Finlayson, 1972).

Advantages of one method over the other in terms of the numerical properties such as convergence and stability may depend on the nature of the problem as well as the solution technique adopted, Newton-Raphson or Predictor-Corrector method for non-linear problems. For simulation of floods of long duration, a stable algorithm with large distance and time steps is needed. As Cooley and Moin (1976) and Manan et al. (1977) indicated, the FEM has some answer.

CHAPTER III

MATHEMATICAL STATEMENT

Introduction

The mathematical expressions of the unsteady gradually varied streamflow hydraulics are afforded by the well-known "Saint Venant Equations," named after Barré de Saint Venant (1871) who first derived them. The original forms of these equations as presented in Chapter II have been modified to include the lateral flow term. These equations are one-dimensional non-linear hyperbolic, initial as well as boundary value partial differential equations, which may be derived from the laws of conservation of mass and momentum.

No attempt is made to re-derive these equations herein, rather interested readers are referred to any basic text on open channel hydraulics, such as Chow (1959), Henderson (1966), Viessman et al. (1972), Wylie and Streeter (1978) and many others.

Governing Differential Equations

The distribution of depth of flow and velocity of flow and discharge in a stream are represented in Figure 2, following. The mathematical model that predicts the flow on a space and time basis can be represented by the following equations, Viessman et al. (1972):

Equation for conservation of mass

$$\frac{\partial y}{\partial t} + y \frac{\partial v}{\partial x} + v \frac{\partial y}{\partial x} - q(x,t) = 0 \tag{3.1}$$

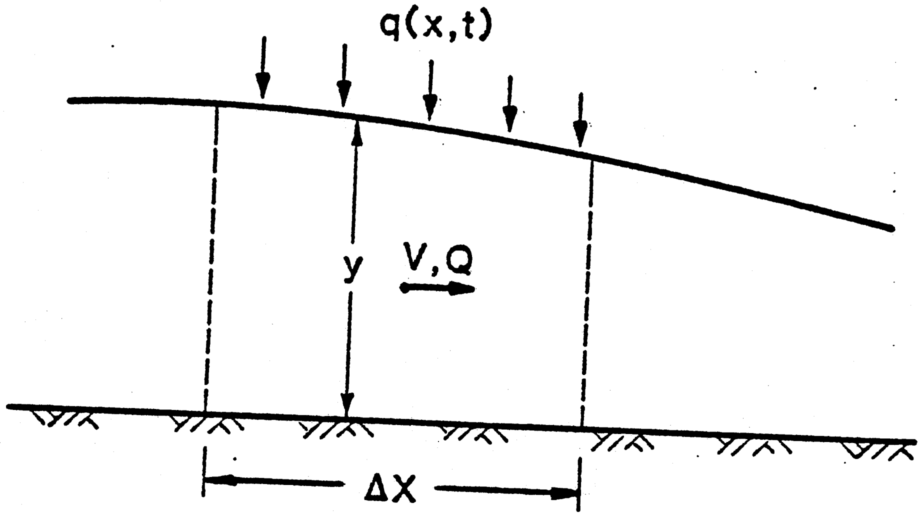


Figure 2. Streamflow Element

Equation for conservation of momentum

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} + \frac{v}{y}q(x,t) + g\ (\frac{\partial y}{\partial x} + S_f - S_o) = 0 \qquad (3.2)$$

where:

q = lateral inflow in the channel reach, Δx, ft per sec;

$S_f = \frac{n1^2 v^2}{2.2082R^{4/3}}$ (friction slope, derived from Manning's EQ.), ft/ft;

n1 = Manning's roughness factor, sec per ft $^{-1/3}$;

R = hydraulic radius, ft;

Other terms are as defined for EQ. (2.1) and (2.2).

The two dependent variables in Equations (3.1) and (3.2) are the depth of flow, y(x,t), and the velocity of flow, v(x,t). The channel geometry is specified by the area of flow, A(x), the hydraulic width, B(x), (where $A(x) = B(x) \cdot \partial y/\partial x$) and the slope, $S_o = S_o(x)$. The lateral inflow q(x,t) has about three possible sources of contribution, namely, the rainfall on the stream, overland flow, and the subsurface inflow.

The conservation of mass and momentum equations presented above are classified as one-dimensional in the sense that flow characteristics such as depth and velocity are considered to vary only in the longitudinal X-direction of the channel. Other simplifying assumptions inherent in their derivation are as follows: (1) the velocity is constant and the water surface is horizontal across any section perpendicular to the longitudinal axis; (2) the flow is gradually varied with hydrostatic pressure prevailing at all points in the flow such that the vertical acceleration of water particles may be neglected; (3) the longitudinal axis of the channel can be approximated by a straight line; (4) the bottom slope of the channel is small; (5) the bed of the channel is fixed, i.e. no scouring or deposition is assumed to occur; (6) the resistance coefficient for steady uniform turbulent flow is considered applicable, and an empirical resistance equation such as the Manning equation describes the resistance effects; and (7) the flow is incompressible and homogeneous in density (Fread, 1976; Freeze, 1972).

Once the velocity of flow and depth of flow are computed from Equations (3.1) and (3.2), the discharge can be computed from the following equation:

$$Q = vy \tag{3.3}$$

where Q = the streamflow volumetric flow rate; cubic feet per sec per channel width of flow.

Initial and Boundary Conditions

The essential requirements to initiate any hydraulic routing, be it open channel or overland flows, are the initial and boundary conditions. The distinction between initial and boundary conditions is merely one of position on the x plane at the commencement of the solution procedure, (Viessman et al., 1972). The initial condition on one hand describes the flow depth, velocity, or discharge at all points in space at time, t = 0. If flow is assumed uniform and steady before any flood wave reaches the point of interest upstream of the entire channel, then either the Manning or Chézy's equation is employed to calculate the initial flow parameters.

On the other hand, the boundary condition refers to the depth, velocity, or discharge at the up- and down-stream points or other point(s) of interest on the river reach at all times, t > 0. Examples of boundary conditions are discussed elegantly by Fread (1976) and summarized in the following equation:

$$My + Nv = P \tag{3.4}$$

where

M, N and P = known functions of either y or v or both. Either M or N is zero at the upstream boundary, and M, N, and P are segments of a rating curve for the downstream boundary.

Important points to keep in mind in boundary condition specification are as follows: (1) if discharge hydrographs are used for both the upstream and downstream boundary conditions, any error in the initial conditions (the initial depth of flow and velocities at all computational nodes along the stream between the up- and down-stream boundaries when the simulation is started) will be perpetuated in the computations (Fread, 1976). This is not the case when other possible combinations of the boundary conditions (specified depths or discharges upstream and rating curve downstream) are used. (2) Associated with the channel hydraulics are two interacting phenomena, namely, the state of flow (subcritical, critical, and supercritical) and the boundary conditions (Viessman et al., 1972). For subcritical flow, boundary conditions are required at both up- and down-stream of the river reach whereas only two upstream boundary conditions are necessary in supercritical flow. This is because downstream effects can not be propagated backward.

Simplified Models

The solution of the complete one-dimensional unsteady flow Equations (3.1) and (3.2) oftentimes results in enormous computer time and storage, particularly for floods of long durations. In essence, this has attracted significant interest in the use of simplified models, such as the kinematic and diffusion flow models. The mathematical justification in the use of these simplified models is provided by the slope approximation analysis and Froude number order of magnitude analysis (Henderson, 1966).

While the continuity equation is completely retained, the simplifying assumptions are made in the momentum equation. If Equation (3.2) is re-arranged with the friction slope, s_f, being the subject of the formula and letting q(x,t) equal zero, the resulting equation is:

$$s_f \quad = \quad s_o - \frac{\partial y}{\partial x} - \frac{v}{g}\frac{\partial v}{\partial x} - \frac{1}{g}\frac{\partial v}{\partial t} \tag{3.5}$$

steady uniform flow
(kinematic flow model)

steady non-uniform flow, I
(diffusion flow model)

steady non-uniform flow, II

unsteady non-uniform flow
(complete flow model)

The volumetric flow rate, Q, is obtained by combining Equation (3.5) and Manning's formula as:

$$Q = \frac{1.486}{nl} AR^{2/3} \sqrt{s_o - \frac{\partial y}{\partial x} - \frac{v}{g}\frac{\partial v}{\partial x} - \frac{1}{g}\frac{\partial v}{\partial t}} \tag{3.6}$$

From Equation (3.6), if the last three slope terms are small compared with s_o, the discharge, Q, can be computed as in uniform flow, and it is dependent on depth only. The resulting relationship is known as the kinematic model, and its momentum equation is expressed as:

$$s_f = s_o = \frac{nl^2 v^2}{2.22R^{4/3}}$$

or (3.7)

$$Q = \frac{1.486}{nl} AR^{2/3} \; S_o^{\frac{1}{2}}$$

The kinematic model has been successfully applied in simulating flows in natural floods in steep river slopes of the order of 10 feet per mile or more, overland flows, and slow-rising hydrographs (Henderson, 1966).

If the longitudinal streambed slope, S_o, is very flat, the $\partial y/\partial x$ term in Equation (3.6) may well be of the same order as S_o. In this case the Froude number±, F, will be very low, so that the third term in Equation (3.6) will be negligible. In fact the third and fourth terms can be shown to be of the same order of magnitude. Details of the mathematical proof are discussed by Henderson (1966). However, for $F^2 << 1$, the terms $v/g\ \partial v/\partial x$ and $1/g\ \partial v/\partial t$ are of the same order of magnitude. This flow condition yields the diffusion flow model. The momentum equation yields:

$$S_f = S_o - \frac{\partial y}{\partial x} \qquad (3.8)$$

Indeed, the kinematic and diffusion flow models are two extreme cases of slopes--steep and flat--which are frequently encountered in overland flow on watersheds and natural routing of a flood wave in streams. Conceivably, there are possible intermediate values of slope for which all the four slope terms in Equation (3.6) would be appreciable. This is a case where the complete flow model is employed.

± Froude number is a dimensionless flow parameter utilized to characterize the state of flow. If the Froude number is less, equal, or greater than unity, the flow is subcritical, critical, and supercritical, respectively.

CHAPTER IV

FINITE ELEMENT FORMULATION

Introduction

The finite element method selected here is the Galerkin's weighted residual principle. This is an excellent choice for the solutions of the unsteady open channel flow equations that are characterized by the nonlinear hyperbolic behavior. This class of equations cannot easily be expressed in the variational form because the governing differential equations are not self-adjoint. Thus, the weighted residual principle, such as the Galerkin's principle, is employed to solve the governing equations of unsteady flow. In its final form, the method generates a system of ordinary differential equations in time for transient problems.

The weighted residuals utilize the concept of orthogonal projections of a residual of a differential equation onto a subspace spanned by certain weighting function. A discussion as it applies to some finite element problems is given by Chung (1978), Norrie and De Vries (1973), Martin and Carey (1975), and Zienkiewicz (1977). The implementation of the finite element formulation of the flow equations is carried out in four basic steps--(1) channel discretization and selection of approximation functions, (2) derivation of element equations, (3) assembly of element equations, (4) transient solution of the system of equations. For sake of clarity, the continuity Equation (3.1) is chosen to illustrate these steps.

Channel Discretization and Selection Of Approximation Function

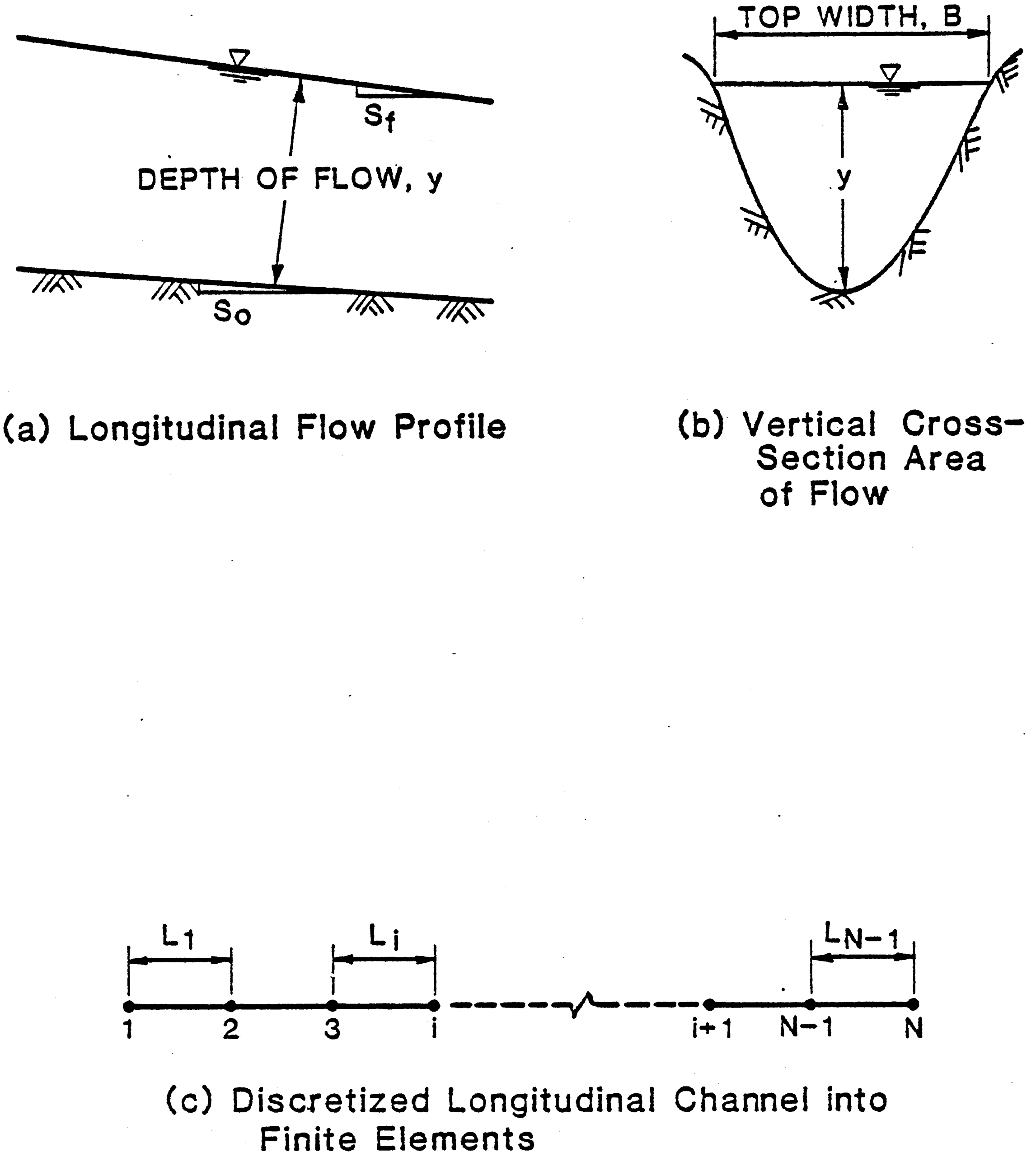
The natural channel shown in Figures 3a and 3b is idealized as a straight line as presented in Figure 3c because the flow equations are one-dimensional. The channel is divided into (N-1) small segments called elements or reaches where N is the total number of nodes for which the solution of the dependent variables is sought. Each element will be modeled with the same flow equation but with different channel geometry and hydraulic properties. The element equations are later assembled into global matrix equations for solution.

To initiate the element equations, the approximation of the dependent variables, such as the velocity of flow, v(x,t), and the depth of flow, y(x,t), that form continuous functions over the infinite distance into discrete variables for a finite distance is necessary in the finite element method. Approximation functions, also known as shape or basis functions, include linear, quadratic, higher order polynomials or spline functions. The linear shape function is utilized to keep calculations simple.

It is important to note that a single function approximating the entire flow domain is difficult to find. The finite element method simplifies the procedure by breaking down or discretizing the function and domain into the elements shown in Figure 4. The characteristics of a shape function are summarized as follows:

1. Each function denoted as N_k^e is zero, except within the element e, and k must be a node of e.

2. The function N_k^e is defined as a continuous function of the independent variable x_k over the element e in such a manner that the value



(a) Longitudinal Flow Profile

(b) Vertical Cross-Section Area of Flow

(c) Discretized Longitudinal Channel into Finite Elements

Figure 3. Natural - Idealized Flow Sections.

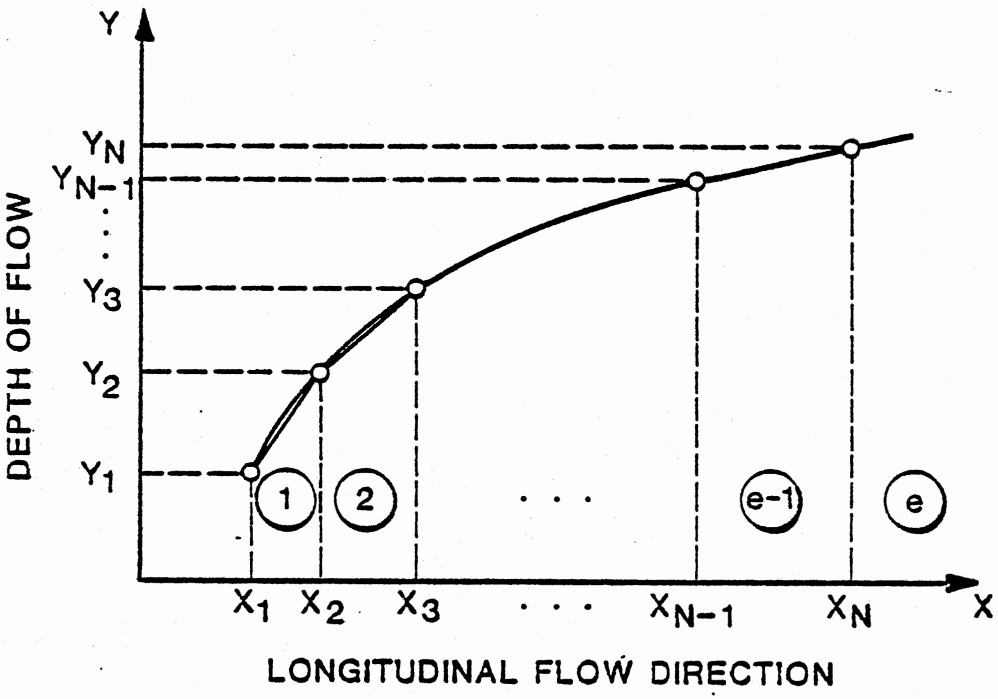


Figure 4. Approximation of Flow Domain by a Piecewise Continuous Function.

at the nodal point k is unity, and the values at the other nodal points of the element are zero.

3. The function y(x,t), or y for simplicity (depth of flow), is allowed to vary linearly in each element:

$$y = A + Bx \tag{4.1}$$

where

A and B = constants

For determining the values of A and B, consider Figure 5 and Equation (4.1). Two simultaneous equations are generated by substituting into Equation (4.1) the corresponding values at points x_k and x_{k+1} respectively. These equations are:

$$y_k = A + Bx_k \tag{4.2}$$

$$y_{k+1} = A + Bx_{k+1} \tag{4.3}$$

Solve for A and B:

$$y_{k+1} - y_k = B(x_{k+1} - x_k)$$

$$B = y_{k+1}/x_{k+1} - x_k$$

$$y_k = A + \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \; x_k$$

$$A = \frac{y_k(x_{k+1} - x_k)}{x_{k+1} - x_k} - \frac{x_k(y_{k+1} - y_k)}{x_{k+1} - x_k}$$

$$A = \frac{x_{k+1}h_k - x_k h_{k+1}}{x_{k+1} - x_k}$$

Thus, the linear shape function becomes:

$$y = \frac{x_{k+1}y_k - x_k y_{k+1}}{x_{k+1} - x_k} + \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \; x \tag{4.4a}$$

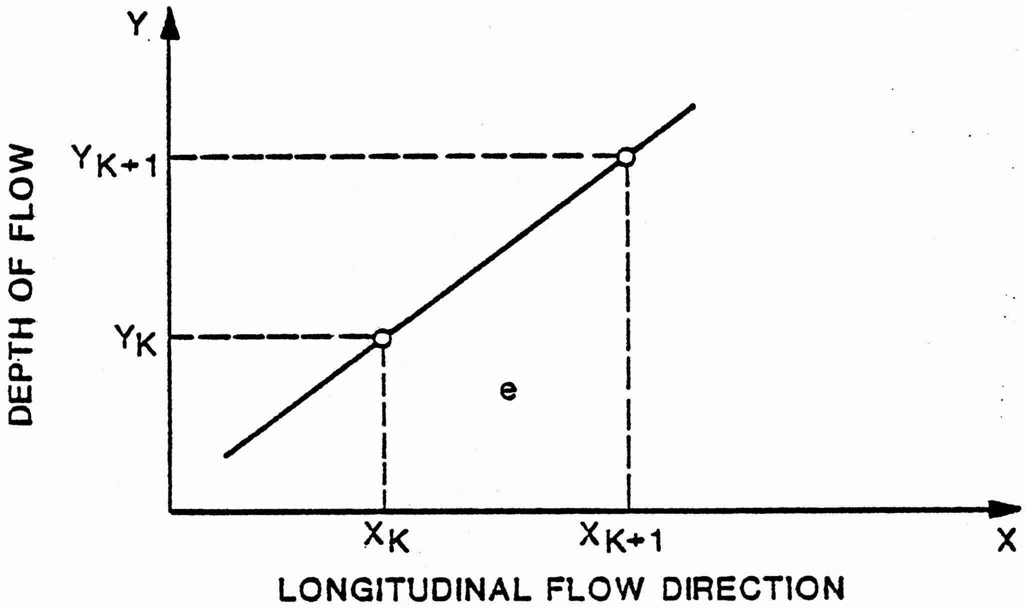


Figure 5. Linear Shape Function.

$$= \frac{x_{k+1} - x}{x_{k+1} - x_k} y_k + \frac{x - x_k}{x_{k+1} - x_k} y_{k+1} \qquad (4.4b)$$

$$= N_k^e \; y_k + N_{k+1}^e \; y_{k+1} \qquad (4.4c)$$

By adjusting the coordinate system in Figure 5 such that the origin is at x_k and the distance from the new origin to x_{k+1} is L, Equation (4.4b) reduces to

$$y = (1 - s)y_k + sy_{k+1} \qquad (4.5)$$

where

$$N_k^e = (1 - s), \; N_{k+1}^e = s \text{ and } s = x/L$$

when $x = x_k$, $N_k^e = 1$ and $N_{k+1}^e = 0$

$x = x_{k+1}$, $N_k^e = 0$ and $N_{k+1}^e = 1$

as required part of the characterisitics of the shape function.

Derivation of Element Equations

The Galerkin's weighted residual method is the basis of the element derivation equations. The method requires that errors or residual between the approximate solution and the true solution be orthogonal to the functions used in the approximation. The principle is expressed mathematically by Segerlind (1976):

$$\int_R N_\beta \; L(\phi)dR = 0 \quad \beta = i, j, k, \ldots \qquad (4.6)$$

where

N_β = shape function;

ϕ = unknown parameter and is approximated by

$$\phi = [N_i, N_j, N_k, \ldots]\{\phi\}; \qquad (4.7)$$

$L(\phi)$ = differential equation governing ϕ; and

R = region of interest.

Equation (4.6) implies that the shape function N_β must be orthogonal to the residual between the approximate solution and the true solution over the region R. Inserting the continuity Equation (3.1) into Equation (4.6) yields:

$$\sum_{1}^{k-1} \int_{x_k}^{x_{k+1}} N^T\left(\frac{\partial y}{\partial t} + y\frac{\partial v}{\partial x} + v\frac{\partial y}{\partial x} - q(x,t)\right)dx = 0 \qquad (4.8)$$

where

$\sum_{1}^{k-1}$ = expression for summing individual element equations from 1 to (k-1) elements;

N^T = transpose to the shape function, and other terms are as defined previously.

Using the shape function, Equation (4.5) into Equation (4.8) gives

$$\begin{array}{l} \text{Term (1)} \quad \text{(2)} \quad \text{(3)} \qquad \text{(4)} \\ \sum_{1}^{k-1} \int_{0}^{1} N^T\left(\frac{\partial y}{\partial t} + y\frac{\partial v}{\partial x} + v\frac{\partial y}{\partial x} - q(x,t)\right)Lds = 0 \end{array} \qquad (4.9)$$

Contribution of terms from left to right in Equation (4.9) is given below:

Term (1):

$$\int_0^1 N^T\left(\frac{\partial y}{\partial t}\right)Lds = \int_0^1 \begin{bmatrix}(1-s)\\ s\end{bmatrix} \begin{bmatrix}(1-s) & s\end{bmatrix} \begin{Bmatrix}\dot{y}_1\\ \dot{y}_2\end{Bmatrix} Lds$$

$$= \frac{L}{6} \begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix} \begin{Bmatrix}\dot{y}_1\\ \dot{y}_2\end{Bmatrix}$$

where

$\dot{y} = \frac{\partial y}{\partial t}$, time derivative of y.

Term (2):

For the second term, first consider the following analysis:

Let $v = N_k V_k + N_{k+1} V_{k+1} \quad = \quad [N]\{V\}$

then $\frac{\partial v}{\partial x} = \frac{\partial}{\partial x}\left([N]\{V\}\right)$

$$= \begin{bmatrix} \frac{\partial N_k}{\partial x} & \frac{\partial N_{k+1}}{\partial x} \end{bmatrix} \begin{Bmatrix} V_k \\ V_{k+1} \end{Bmatrix}$$

Thus

$$\int_0^1 N^T\left(y\,\frac{\partial v}{\partial x}\right)Lds = \int_0^1 \begin{bmatrix} (1-s) \\ s \end{bmatrix} \begin{bmatrix} (1-s) & s \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix}$$

$$= 1/6 \begin{bmatrix} (2y_1 + y_2)(v_2 - v_1) \\ (v_1 + 2v_2)(v_2 - v_1) \end{bmatrix} \qquad (4.9b)$$

Term (3):

The third term is

$$\int_0^1 N^T\left(v\frac{\partial y}{\partial x}\right)Lds = 1/6 \begin{bmatrix} (2v_1 + v_2)(y_2 - y_1) \\ (v_1 + 2v_2)(y_2 - y_1) \end{bmatrix} \qquad (4.9c)$$

Term (4):

The last term is

$$\int_0^1 N^T(q(x,t))Lds = q(x,t)\int_0^1 \begin{bmatrix} (1-s) \\ s \end{bmatrix} Lds$$

$$= \frac{L}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} q(x,t) \qquad (4.9d)$$

Combining each of the evaluated terms yields the following element equation:

$$\frac{L}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{Bmatrix}\dot{y}_1\\ \dot{y}_2\end{Bmatrix} + 1/6\begin{bmatrix}(2y_1 + y_2)(v_2 - v_1)\\ (y_1 + 2y_2)(v_2 - v_1)\end{bmatrix} + 1/6\begin{bmatrix}(2v_1+v_2)(y_2-y_1)\\ (v_1+2v_2)(y_2-y_1)\end{bmatrix}$$

$$- \frac{9L}{2}\begin{Bmatrix}1\\ 1\end{Bmatrix} = 0 \qquad (4.10)$$

Multiplying the Equation (4.10) by a factor of 6 and adding up the two middle terms, we obtain:

$$L\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{Bmatrix}\dot{y}_1\\ \dot{y}_2\end{Bmatrix} + \begin{bmatrix}v_2-4v_1 & 2v_2+v_1\\ -v_2-2v_1 & 4v_2-v_1\end{bmatrix}\begin{Bmatrix}y_1\\ y_2\end{Bmatrix} - Lq\begin{Bmatrix}3\\ 3\end{Bmatrix} = 0 \qquad (4.11)$$

In a manner analogous to the above procedure, the momentum Equation (3.2) for an element can be derived as:

$$\frac{L}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{Bmatrix}\dot{v}_1\\ \dot{v}_2\end{Bmatrix} + 1/12\begin{bmatrix}-2v_1-v_2 & -v_1-2v_2\\ 2v_1+v_2 & v_1+2v_2\end{bmatrix}\begin{Bmatrix}v_1\\ v_2\end{Bmatrix} + 9/2\begin{bmatrix}-1 & 1\\ -1 & 1\end{bmatrix}\begin{Bmatrix}y_1\\ y_2\end{Bmatrix}$$

$$+ \frac{Lq}{2}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{Bmatrix}(v/y)_1\\ (v/y)_2\end{Bmatrix} + \frac{gL}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{Bmatrix}s_{f1}\\ s_{f2}\end{Bmatrix} - \frac{gS_oL}{2}\begin{Bmatrix}1\\ 1\end{Bmatrix} = 0$$

$$(4.12)$$

Assembly of Element Equations

The element properties originally expressed in local coordinates need to be transformed into global coordinates before solution algorithm is initiated. Based on the node-to-node relationship (Figure 3c), it is possible to generate an overall element property matrix for the entire domain, a process called assembling of element equations.

The concept of discretization employed earlier is based on the fact that a domain with varying geometric and hydraulic properties can be treated independently as subdomains but systematically from one subdomain to another. Assuming that the elements are of variable lengths and that there are N nodes, the assembled global matrix equation for the continuity Equation (4.11) becomes:

$$\begin{bmatrix} 2L_1 & L_1 & 0 & 0 & 0 & 0 & 0 \\ L_1 & 2(L_1+L_2) & L_2 & 0 & 0 & 0 & 0 \\ 0 & L_2 & 2(L_2+L_3) & L_3 & 0 & 0 & 0 \\ \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} \\ 0 & 0 & 0 & L_i & 2(L_i+L_{i+1}) & L_{i+1} & 0 \\ \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} \\ 0 & 0 & 0 & 0 & 0 & L_{N-1} & L_{N-1} \end{bmatrix} \begin{Bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \text{--} \\ \dot{y}_i \\ \text{--} \\ \dot{y}_N \end{Bmatrix} +$$

$$\begin{bmatrix} v_2-4v_1 & 2v_2+v_1 & 0 & 0 & 0 & 0 \\ -v_2-2v_1 & v_3-v_1 & 2v_3-v_2 & 0 & 0 & 0 \\ 0 & -v_3-2v_2 & v_4-v_2 & 2v_4-v_3 & 0 & 0 \\ \text{----} & \text{----} & \text{---} & \text{----} & \text{----} & \text{----} \\ 0 & 0 & -v_{i+1}-2v_i & v_{i+2}-v_i & 2v_{i+2}+v_{i+1} & 0 \\ \text{----} & \text{----} & \text{---} & \text{----} & \text{----} & \text{----} \\ 0 & 0 & 0 & 0 & -v_N-2v_{N-1} & 4v_N-v_{N-1} \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ \text{--} \\ y_i \\ \text{--} \\ y_N \end{Bmatrix} -$$

$$3 \begin{Bmatrix} q_1L_1 \\ q_1L_1+q_2L_2 \\ q_2L_2+q_3L_3 \\ \text{-------} \\ q_iL_i+q_{i+1}L_{i+1} \\ \text{-------} \\ q_{N-1}L_{N-1} \end{Bmatrix} = 0 \qquad (4.13)$$

The general form of the above assembled global continuity equation can be expressed as:

$$[A]\{\dot{y}\} + [B]\{y\} - \{C\} = 0 \tag{4.14}$$

where

A, B are matrices and C is a column vector;

$\dot{y}$ is the time derivative;

y and v are dependent variables.

The momentum equation follows the same pattern of assembly.

Transient Solution Approach

The solution to the time-dependent global matrix Equation (4.13) is sought through a "semi-discrete" approach. This approach requires the time derivative of the dependent variable at each node to be replaced by a finite difference scheme in time domain. A simple illustration of the semi-discrete approach can be demonstrated by considering Equation (4.14). The time derivative, $\dot{y}$, will be replaced by a finite difference scheme, such as the forward, backward, and central difference. These are respectively given below:

Forward Difference $$\dot{y} = \frac{y^{K+1}-y^{K}}{\Delta t} \tag{4.15a}$$

Backward Difference $$\dot{y} = \frac{y^{K}-y^{K-1}}{\Delta t} \tag{4.15b}$$

Central Difference $$\dot{y} = \frac{y^{K+1}-y^{K-1}}{2\Delta t} \tag{4.15c}$$

where:

K = time level.

Substitution of Equation (4.15a) into Equation (4.14) yields

$$[A]\ \{\frac{y^{K+1}-y^{K}}{\Delta t}\} + [B]\{y^{K}\} - \{C\} = 0 \tag{4.16}$$

An implicit equation can be generated from Equation (4.16) with the aid of the time weighting factor. This subject is discussed elegantly in Chapter V.

CHAPTER V

NUMERICAL FLOW MODELS

Introduction

Three distinct deterministic streamflow routing models are investigated and are discussed in this chapter in their order of increasing complexity: (1) The kinematic flow model comprises (a) the simplified version of the momentum Equation (3.2) that neglects pressure and inertia terms as compared to friction and gravity terms (see Equation 3.7) and (b) the complete form of continuity Equation (3.1); (2) the diffusion flow model combines (c) the simplified momentum equation that accounts only for pressure, friction,and gravity terms, Equation (3.8), and (d) the complete form of continuity Equation (3.1); and (3) the complete flow model comprises the complete forms of both continuity Equation (3.1) and momentum Equation (3.2).

The kinematic flow model is investigated in both an explicit and implicit sense. The explicit kinematic flow model leads to linear equations. They are solved using a direct method similar to the tridiagonal matrix algorithm set-up by Varga (1962). Solution proceeds by matrix reduction similar to Gaussian elimination. In contrast to the explicit model, the weighted implicit kinematic model yields a set of non-linear tridiagonal matrix equations which are solved by the functional Newton-Raphson iterative method. This method is known as implicit because the set of equations are solved by an indirect method.

The diffusion flow model, as well as the complete flow model each results in a non-linear bi-tridiagonal matrix equation. The functional Newton-Raphson's method, along with the direct solution algorithm*, triangular decomposition technique that yields a recursion algorithm (Douglas et al., 1959; Von Resenberg, 1975), is utilized to predict depth and velocity of flow for each model.

Explicit Kinematic Finite Element Model, EKFEM

The non-linear continuity Equation (3.1) is easily converted to linear form by use of geometric and flow relations:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} - q(x,t) = 0 \tag{5.1a}$$

where

A = area of flow, ft^2;

Q = volumetric flow rate, ft^3/sec.

The appropriate simplified momentum equation for coupling with the continuity Equation (5.1a) has been obtained and is presented below:

$$S_f = S_o = n1^2v^2/2.22R^{4/3}$$

or (5.1b)

$$Q = \frac{1.486}{n1} AR^{2/3}So^{1/3}$$

Applying the Galerkin's weighted residual method to Equation (5.1a) results in the following linear first order ordinary differential equation (see Equation 4.11):

* Direct solution algorithm for a linearized bi-tridiagonal matrix equation stored in compact (2Nx6) matrix, where N is total number of nodes, was originally developed by Douglas et al. (1959) and is presented in Appendix B.

$$\frac{L}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{Bmatrix}\dot{A}_1\\ \dot{A}_2\end{Bmatrix} + 1/2\begin{bmatrix}-1 & 1\\ -1 & 1\end{bmatrix}\begin{Bmatrix}Q_1\\ Q_2\end{Bmatrix} - qL/2\begin{Bmatrix}1\\ 1\end{Bmatrix} = 0 \qquad (5.2)$$

For the entire channel reach the assembled matrix equation becomes:

$$1/6\left[\begin{array}{ccccccc}2L_1 & L_1 & & & & \bigcirc & \\ L_2 & 2(L_1+L_2) & L_2 & & & & \\ & L_2 & 2(L_2+L_3) & L_3 & & & \\ & -- & -- & -- & -- & -- & -- \\ & & L_i & 2(L_i+L_{i+1}) & L_{i+1} & & \\ \bigcirc & & -- & -- & -- & -- & -- \\ & & & & L_{N-1} & 2L_{N-1} & \end{array}\right]\begin{Bmatrix}\dot{A}_1\\ \dot{A}_2\\ \dot{A}_3\\ --\\ \dot{A}_i\\ --\\ \dot{A}_N\end{Bmatrix} +$$

$$\begin{Bmatrix}Q_2-Q_1\\ Q_3-Q_1\\ Q_4-Q_2\\ -----\\ Q_{i+1}-Q_{i-1}\\ -----\\ Q_N-Q_{N-1}\end{Bmatrix} - 1/2\begin{Bmatrix}L_1q_1\\ L_1q_1 + L_2q_2\\ L_2q_2 + L_3q_3\\ ----------\\ L_iq_i + L_{i+1}q_{i+1}\\ ----------\\ L_{N-1}q_{N-1}\end{Bmatrix} = 0 \qquad (5.3)$$

Equation (5.3) is equivalently expressed in a matrix form:

$$[K]\{\dot{A}\} + \{D\} - \{F\} = 0 \qquad (5.4)$$

The time solution of Equation (5.4) is possible upon implementation of the forward differencing in time domain.

$$[K]\{A\}^{n+1} = [K]\{A\}^n + \Delta t\{F\}^n - \Delta t\{D\}^n \qquad (5.5)$$

The solution of the area of flow at various nodes proceeds forward in time with the right hand side evaluated at a previous time level, n. Thus, the Equation (5.5) can be expressed in a more compact form:

$$[K]\{A\}^{n+1} = \{X\}^{n} \tag{5.6}$$

where

X = known column vector at previous time level.

The matrix, K, is a linear and tridiagonal type that easily leads to a direct solution algorithm. The computer program solving Equation (5.6) is facilitated by the use of the compact tridiagonal algorithm proposed by Varga (1962). The computed area of flow at current time level, n+1, is used to update the volumetric flow rate, Q, Equation (5.1b). The solution cycle is repeated as new time level is reached. The coded explicit finite element scheme exhibits dynamic stability due to restrictions on time step. This drawback inherent in explicit numerical schemes is expected regardless of the finite element approach. However, the stability problem is corrected in the weighted implicit flow model.

Weighted Implicit Kinematic Finite Element Model, WIKFEM

The implicit kinematic flow model begins by combining the non-linear continuity matrix Equation (4.14) with the modified momentum Equation (5.8) (with velocity the subject of the formulation rather than the volumetric flow rate). The introduction of the dimensionless time weighting factor, θ, Figure 1, and the forward differencing to Equation (4.14) yeilds the following:

$$[A+\Delta t\theta B]\{y\}^{n+1} - \theta\Delta t\{C\}^{n+1} = [A+\Delta t(1-\theta)B]\{y\}^{n}+\Delta t(1-\theta)\{C\}^{n} \tag{5.7}$$

And the modified momentum equation is repeated here for convenience as:

$$v = \frac{1.486}{n1} R^{2/3} S_0{}^{1/2} \qquad (5.8)$$

where all terms are as previously defined for Equations (3.1), (3.2), and (4.14).

The expanded form of Equation (5.7) for the upstream, interior, and downstream nodes, respectively, are given below:

$$\begin{aligned} F_1 \equiv\ & [(2L_1+\Delta t\theta(v_2-4v_1))y_1 + (L_1+\Delta t\theta(2v_2+v_1))y_2]^{n+1} - [3\Delta t\theta L_1 q_1]^{n+1} \\ & + [(-2L_1+\Delta t(1-\theta)(v_2-4v_1))y_1 + (-L_1+\Delta t(1-\theta)(2v_2+v_1))y_2]^n \\ & - [3\Delta t(1-\theta)L_1 q_1]^n = 0 \end{aligned} \qquad (5.9a)$$

$$\begin{aligned} F_i \equiv\ & [(L_{i-1}+\Delta t\theta(-v_i-2v_{i-1}))y_{i-1} + (2(L_{i-1}+L_i)+\Delta t\theta(v_{i+1}-v_{i-1}))y_i \\ & + (L_i+\Delta t\theta(2v_{i+1}+v_i))y_{i+1}]^{n+1} - 3\Delta t\theta[q_{i-1}L_{i-1}+q_iL_i]^{n+1} \\ & + [(-L_{i-1}+\Delta t(1-\theta)(-v_i-2v_{i-1}))y_{i-1} + (-2(L_{i-1}+L_i)+\Delta t(1-\theta)(v_{i+1}-v_{i-1}))y_i \\ & + (-L_i+\Delta t(1-\theta)(2v_{i+1}+v_i))y_{i+1}]^n - 3\Delta t(1-\theta)[q_{i-1}L_{i-1}+q_iL_i]^n = 0 \end{aligned}$$

$$(5.9b)$$

$$\begin{aligned} F_N \equiv\ & [(L_{N-1}-\Delta t\theta(v_N+2v_{N-1}))y_{N-1} + (2L_{N-1}+\Delta t\theta(4v_N-v_{N-1}))y_N]^{n+1} \\ & - [3\Delta t\theta L_{N-1}q_{N-1}]^{n+1} + [(-L_{N-1}+\Delta t(1-\theta)(-v_N-2v_{N-1}))y_{N-1} \\ & + (-2L_{N-1}+\Delta t(1-\theta)(4v_N-v_{N-1}))y_N]^n - [3\Delta t(1-\theta)L_{N-1}q_{N-1}]^n = 0 \end{aligned} \qquad (5.9c)$$

The solution of Equation (5.9) is obtained through the generalized functional iterative method known as the Newton-Raphson method, first used by Amein and Fang (1969) and later by Fread (1971, 1976). Equation (5.9) expressed in functional form is as follows:

$$F_1\ (y_1,\ y_2) = 0 \tag{5.10a}$$

$$F_i\ (y_{i-1},\ y_i,\ y_{i+1}) = 0 \tag{5.10b}$$

$$F_N\ (y_{N-1},\ y_N) = 0 \tag{5.10c}$$

The y terms in the parentheses are the depth of flow, the dependent variable to be solved. The subscript associated with each y denotes the nodal location. The computed values of y are utilized in Equation (5.8) to generate the corresponding values of velocity of flow, v. For the system of N non-linear equations with N unknowns, computation is initiated by assigning trial values to the N unknowns. The substitution of the trial values into the system of non-linear equations yields a set of N residuals. In fact, the residual is the value of the right-hand side of the equation after the trial values are substituted in Equation (5.10). The final solution is obtained when the residuals are reduced to a suitable tolerance level.

If it is assumed that the computations have been carried through the j^{th} iteration, in other words the values of the unknowns have been approximated through the j^{th} iteration, then is possible to estimate the value of the residual as follows:

$$F_1(y_1^{\,j},\ y_2^{\,j}) = R_1^{\,j} \tag{5.11a}$$

$$F_i(y_{i-1}^{\,j},\ y_i^{\,j},\ y_{i+1}^{\,j}) = R_i^{\,j} \tag{5.11b}$$

$$F_N(y_{N-1}^{\,j},\ y_N^{\,j}) = R_N^{\,j} \tag{5.11c}$$

where $R_i^{\,j}$ is the residual at the j^{th} iteration cycle for the i^{th} node. The Newton-Raphson algorithm ties up the residual and partial derivatives of the system of Equations (5.11) in the following manner:

$$\begin{bmatrix} \frac{\partial F_1}{\partial y_1} & \frac{\partial F_1}{\partial y_2} & & & \\ \frac{\partial F_2}{\partial y_1} & \frac{\partial F_2}{\partial y_2} & \frac{\partial F_2}{\partial y_3} & \bigcirc & \\ & \frac{\partial F_3}{\partial y_2} & \frac{\partial F_3}{\partial y_3} & \frac{\partial F_2}{\partial y_4} & \\ & --- & --- & --- & --- \\ & & \frac{\partial F_i}{\partial y_{i-1}} & \frac{\partial F_i}{\partial y_i} & \frac{\partial F_i}{\partial y_{i+1}} \\ \bigcirc & & --- & --- & --- \\ & & & \frac{\partial F_N}{\partial y_{N-1}} & \frac{\partial F_N}{\partial y_N} \end{bmatrix} \begin{Bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ --- \\ \Delta y_{i-1} \\ \Delta y_i \\ \Delta y_{i+1} \\ --- \\ \Delta y_{N-1} \\ \Delta y_N \end{Bmatrix}^j = \begin{Bmatrix} R_1 \\ R_3 \\ R_3 \\ --- \\ R_i \\ --- \\ R_N \end{Bmatrix}^j \tag{5.12}$$

where $\Delta y = y^{j+1} - y^j$ (the difference between current and previous iterates of y).

The matrix to the left-hand side of Equation (5.12) is the tridiagonal Jacobian matrix of size (NxN). It is possible to store the matrix in a compact (Nx3) form as shown in Equation (5.13) following:

$$\begin{bmatrix} \bigcirc & \frac{\partial F_1}{\partial y_1} & \frac{\partial F_1}{\partial y_2} \\ \frac{\partial F_2}{\partial y_1} & \frac{\partial F_2}{\partial y_2} & \frac{\partial F_2}{\partial y_3} \\ \frac{\partial F_3}{\partial y_2} & \frac{\partial F_3}{\partial y_3} & \frac{\partial F_3}{\partial y_4} \\ --- & --- & --- \\ \frac{\partial F_i}{\partial y_{i-1}} & \frac{\partial F_i}{\partial y_i} & \frac{\partial F_i}{\partial y_{i+1}} \\ --- & --- & --- \\ \frac{\partial F_N}{\partial y_{N-1}} & \frac{\partial F_N}{\partial y_N} & \bigcirc \end{bmatrix} \begin{Bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ --- \\ \Delta y_{i-1} \\ \Delta y_i \\ \Delta y_{i+1} \\ --- \\ \Delta y_{N-1} \\ \Delta y_N \end{Bmatrix} = \begin{Bmatrix} R_1 \\ R_2 \\ R_3 \\ --- \\ R_i \\ --- \\ R_N \end{Bmatrix} \tag{5.13}$$

The right-hand side of Equation (5.13) comprises the column vector generated upon substitution of the trial values of the unknowns into Equation (5.9). The individual terms of the Jacobian are generated from Equation (5.9) and written below as:

$$\frac{\partial F_1}{\partial y_1} = 2L_1 + FAC(v_2 - 4v_1)$$

$$\frac{\partial F_1}{\partial y_2} = L_1 + FAC(2v_2 + v_1)$$

$$\frac{\partial F_i}{\partial y_{i-1}} = L_{i-1} + FAC(-v_i - 2v_{i-1})$$

$$\frac{\partial F_i}{\partial y_i} = 2(L_{i-1} + L_i) + FAC(v_{i+1} - v_i) \qquad (5.14)$$

$$\frac{\partial F_i}{\partial y_{i+1}} = L_i + FAC(2v_{i+1} + v_i)$$

$$\frac{\partial F_N}{\partial y_{N-1}} = L_{N-1} + FAC(-v_N - 2v_{N-1})$$

$$\frac{\partial F_N}{\partial y_N} = 2L_{N-1} + FAC(4v_N - v_{N-1})$$

where FAC = $\theta\Delta t$.

Equation (5.13) is a linearized form of the non-linear weighted implicit kinematic model similar to Equation (5.6) and is solved in the same manner. The computer program for Equation (5.13) does not require the Jacobian matrix to be up-dated for every iteration, rather after every three iterations. The approach seems reasonable in terms of minimizing the computer time because convergence is achieved with relatively few iterations for most time steps employed. For the guess values of the dependent variable, y, required to initiate the iterative Newton-Raphson equation solver, the initial uniform flow depths are utilized. The initial depths of flow prior to the flood into the channel are the best guess to use. Proper upstream and downstream boundary conditions, such as discharge hydrograph and loop-rating curve, are incorporated in the model. The solution is then sought for a prescribed convergence error criterion.

The effectiveness of the weighted implicit model as compared to the explicit version, along with the other two flow models, is discussed in Chapter VI.

Weighted Implicit Diffusion Finite Element Model, WIDFEM

Another simplified model is the diffusion flow model. The model is developed by coupling the continuity Equation (3.1) and the simplified momentum Equation (3.8). The finite element transformation procedure for Equation (3.1) is given in Equations (4.8) through (4.14). The same principles are applied to Equation (3.8), resulting in the following element equation:

$$1/2 \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} + \frac{L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} S_{f1} \\ S_{f2} \end{Bmatrix} - \frac{L}{2} S_o \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} = 0 \qquad (5.15)$$

where

$S_f = FR\ v^2/R^{4/3}$ and

$FR = n1^2/2.2082.$

The assembled matrix equation of Equation (5.15), along with the dimensionless time weighting factor and the forward time differencing, becomes:

$$\begin{aligned}&\Delta t\theta[F]\{y^{n+1}\} + \Delta t\theta \frac{FR}{(R^{n+1})^{4/3}} [D]\{v^{n+1}\}^2 - \Delta t\theta\{M^{n+1}\}\\ &= -\Delta t(1-\theta)[F]\{y^n\} - \Delta t(1-\theta)\frac{FR}{(R^n)^{4/3}}[D]\{v^n\}^2 + t(1-)\{M^n\}\end{aligned} \tag{5.16}$$

where the superscripts (n+1) and (n) for the variables y, v, and R are the current and previous iterations, respectively. The expanded forms of Equation (5.16) similar to Equation (5.9) for the upstream, interior, and downstream nodes are respectively given as:

$$\begin{aligned}G_1 \equiv\ & L_1(FR)\Delta t\theta[2v_1{}^2/R_1{}^{4/3} + v_2{}^2/R_2{}^{4/3}]^{n+1} + 3\Delta t\theta[y_2-y_1]^{n+1}\\ &- 3\Delta t\theta[S_oL_1]^{n+1} + L_1(FR)\Delta t(1-\theta)[2v_1{}^2/R_1{}^{4/3} + v_2{}^2/R_2^{4/3}]^n\\ &+ 3\Delta t(1-\theta)[y_2-y_1]^n - 3\Delta t(1-\theta)[S_oL_1]^n\end{aligned} \tag{5.17a}$$

$$\begin{aligned}G_i \equiv\ & \Delta t\theta(FR)[L_{i-1}v_{i-1}^2/R_{i-1}^{4/3} + 2(L_{i-1}+L_i)v_i{}^2/R_i{}^{4/3} + L_iv_{i+1}^2/R_{i+1}^{4/3}]^{n+1}\\ &+ 3\Delta t\theta[y_{i+1}-y_{i-1}]^{n+1} - 3\Delta t\theta[(L_{i-1}+L_i)S_o]^{n+1}\\ &+ \Delta t(1-\theta)(FR)[L_{i-1}v_{i-1}^2/R_{i-1}^{4/3} + 2(L_{i-1}+L_i)v_i{}^2/R_i{}^{4/3}+L_iv_{i+1}^2/R_{i+1}^{4/3}]^n\\ &+ 3\Delta t(1-\theta)[y_{i+1}-y_{i-1}]^n - 3\Delta t(1-\theta)[(L_{i-1}+L_i)S_o]^n\end{aligned} \tag{5.17b}$$

$$\begin{aligned}G_N \equiv\ & L_{N-1}(FR)\Delta t\theta[v_{N-1}^2/R_{N-1}^{4/3} + 2v_N{}^2/R_N{}^{4/3}]^{n+1} + 3\Delta t\theta[y_N-y_{N-1}]^{n+1}\\ &- 3\Delta t\theta[S_oL_{N-1}]^{n+1} + L_{N-1}(FR)\Delta t(1-\theta)[v_{N-1}^2/R_{N-1}^{4/3}]^n + 3\Delta t(1-\theta)[y_N-y_{N-1}]^n\\ &- 3\Delta t(1-\theta)[S_oL_{N-1}]^n\end{aligned} \tag{5.17c}$$

The simultaneous solution of Equations (5.9) and (15.17) is possible using the generalized functional iterative method, known as the Newton-Raphson method discussed earlier in the implicit kinematic flow model. The functional representations of Equations (5.9) and (15.17) are as follows:

$$\begin{array}{c}\text{upstream}\\\text{nodes}\end{array}\quad \begin{aligned} F_1\ (y_1, v_1, y_2, v_2) &= 0 \\ G_1\ (y_1, v_1, y_2, v_2) &= 0 \end{aligned} \tag{5.18a}$$

$$\begin{array}{c}\text{interior}\\\text{nodes}\end{array}\quad \begin{aligned} F_i\ (y_{i-1}, v_{i-1}, y_i, v_i, y_{i+1}, v_{i+1}) &= 0 \\ G_i\ (y_{i-1}, v_{i-1}, y_i, v_i, y_{i+1}, v_{i+1}) &= 0 \end{aligned} \tag{5.18b}$$

$$\begin{array}{c}\text{downstream}\\\text{nodes}\end{array}\quad \begin{aligned} F_N\ (y_{N-1}, v_{N-1}, y_N, v_N) &= 0 \\ G_N\ (y_{N-1}, v_{N-1}, y_N, v_N) &= 0 \end{aligned} \tag{5.18c}$$

Similar to the implicit kinematic model, the substitution of the trial values for v and y into the system of non-linear Equations (5.18) yields a set of 2N residuals. Furthermore, the computations are carried through the j^{th} iteration cycle; then the estimates of the residuals are as follows:

$$\begin{aligned}
&F_1\ (y_1^j, v_1^j, y_2^j, v_2^j) = R_{F_1}^j \\
&G_1\ (y_1^j, v_1^j, y_2^j, v_2^j) = R_{G_1}^j \\
&F_i\ (y_{i-1}^j, v_{i-1}^j, y_i^j, v_i^j, y_{i+1}^j, v_{i+1}^j) = R_{F_i}^j \\
&G_i\ (y_{i-1}^j, v_{i-1}^j, y_i^j, v_i^j, y_{i+1}^j, v_{i+1}^j)\ R_{G_i}^j \\
&F_N\ (y_{N-1}^j, v_{N-1}^j, y_N^j, v_N^j) = R_{F_N}^j \\
&G_N\ (y_{N-1}^j, v_{N-1}^j, y_N^j, v_N^j) = R_{G_N}^j
\end{aligned} \tag{5.19}$$

Where $R^j_{F_i}$ and $R^j_{G_i}$ are the residuals at the j^{th} interation cycle for the continuity and momentum equations, respectively, at i^{th} node.

The Newton-Raphson algorithm couples the residuals and the partial derivatives of the systems of Equation (15.19) in the following form:

$$\left[\begin{array}{cccccccc}
\frac{\partial F_1}{\partial y_1} & \frac{\partial F_1}{\partial v_1} & \frac{\partial F_1}{\partial y_2} & \frac{\partial F_1}{\partial v_2} & & & & \\
\frac{\partial G_1}{\partial y_1} & \frac{\partial G_1}{\partial v_1} & \frac{\partial G_1}{\partial y_2} & \frac{\partial G_1}{\partial v_2} & & \bigcirc & & \\
\hdashline
 & \frac{\partial F_i}{\partial y_{i-1}} & \frac{\partial F_i}{\partial v_{i-1}} & \frac{\partial F_i}{\partial y_i} & \frac{\partial F_i}{\partial v_i} & \frac{\partial F_i}{\partial y_{i+1}} & \frac{\partial F_i}{\partial v_{i+1}} & \\
 & \frac{\partial G_i}{\partial y_{i-1}} & \frac{\partial G_i}{\partial v_{i-1}} & \frac{\partial G_i}{\partial y_i} & \frac{\partial G_i}{\partial v_i} & \frac{\partial G_i}{\partial y_{i+1}} & \frac{\partial G_i}{\partial v_{i+1}} & \\
\hdashline
 & & & \frac{\partial F_N}{\partial y_{N-1}} & \frac{\partial F_N}{\partial v_{N-1}} & \frac{\partial F_N}{\partial y_N} & \frac{\partial F_N}{\partial v_N} & \\
 & \bigcirc & & \frac{\partial G_N}{\partial y_{N-1}} & \frac{\partial G_N}{\partial v_{N-1}} & \frac{\partial G_N}{\partial y_N} & \frac{\partial G_N}{\partial v_N} &
\end{array}\right]
\left\{\begin{array}{c}
\Delta y_1 \\ \Delta v_1 \\ \Delta y_2 \\ \Delta v_2 \\ ---- \\ \Delta y_{i-1} \\ \Delta v_{i-1} \\ \Delta y_i \\ \Delta v_i \\ \Delta y_{i+1} \\ \Delta v_{i+1} \\ ---- \\ \Delta y_{N-1} \\ \Delta v_{N-1} \\ \Delta y_N \\ \Delta v_N
\end{array}\right\}^j
=
\left\{\begin{array}{c}
R_{F_1} \\ R_{G_1} \\ ---- \\ R_{F_i} \\ R_{G_i} \\ ---- \\ R_{F_N} \\ R_{G_N}
\end{array}\right\}^j \tag{5.20}$$

where $\Delta y = y^{j+1} - y^j$ and $\Delta v = v^{j+1} - v^j$.

The matrix to the left-hand side of Equation (5.20) containing the partial derivatives of the functions F and G is the bi-tridiagonal Jacobian matrix of size (2N x2N). The maximum non-zero elements in any single row is six. Thus, the Jacobian matrix is stored in a compact (2N x 6) matrix. The individual terms of the Jacobian matrix for the function G are given below whereas those of F are noted in Equation (5.14).

$$\frac{\partial G_1}{\partial y_1} = -FAC\left[\frac{8L_1(FR){v_1}^2}{3{R_1}^{7/3}}\;\frac{\partial R_1}{\partial y_1} + 3.0\right]$$

$$\frac{\partial G_1}{\partial y_2} = 4L_1(FR)(FAC)v_1/{R_1}^{4/3} \tag{5.21a}$$

$$\frac{\partial G_1}{\partial y_2} = -FAC\left[\frac{4L_1(FR){v_2}^2}{3{R_2}^{7/3}}\;\frac{\partial R_2}{\partial y_2} - 3.0\right]$$

$$\frac{\partial G_1}{\partial v_2} = 2L_1(FR)(FAC)v_2/{R_2}^{4/3}$$

$$\frac{\partial G_i}{\partial y_{i-1}} = -FAC\left[\frac{-4L_{i-1}(FR)v_{i-1}^2}{3R_{i-1}^{7/3}}\;\frac{\partial R_{i-1}}{\partial y_{i-1}} + 3.0\right]$$

$$\frac{\partial G_i}{\partial v_{i-1}} = 2L_{i-1}(FR)(FAC)v_{i-1}/R_{i-1}^{4/3}$$

$$\frac{\partial G_i}{\partial y_i} = -FAC\left[\frac{8(L_{i-1}+L_i)(FR){v_i}^2}{3{R_i}^{7/3}}\;\frac{\partial R_i}{\partial y_i}\right] \tag{5.21b}$$

$$\frac{\partial G_i}{\partial v_i} = 4(L_{i-1}+L_i)(FR)(FAC)v_i/{R_i}^{4/3}$$

$$\frac{\partial G_i}{\partial y_{i+1}} = -FAC\left[\frac{4L_i(FR)v_{i+1}^2}{3R_{i+1}^{7/3}}\;\frac{\partial R_{i+1}}{\partial y_{i+1}} - 3.0\right]$$

$$\frac{\partial G_i}{\partial v_{i+1}} = 2L_i(FR)(FAC)v_{i+1}/{R_{i+1}}^{4/3}$$

$$\frac{\partial G_N}{\partial y_{N-1}} = -FAC\left[\frac{4L_{N-1}(FR)v_{N-1}^2}{3R_{N-1}^{7/3}}\;\frac{\partial R_{N-1}}{\partial y_{N-1}} + 3.0\right] \tag{5.21c}$$

$$\frac{\partial G_N}{\partial v_{N-1}} = 2L_{N-1}(FR)(FAC)v_{N-1}/R_{N-1}^{4/3}$$

$$\frac{\partial G_N}{\partial y_N} = -FAC\left[\frac{8L_{N-1}(FR)v_N^{\ 2}}{3R_{N-1}^{7/3}}\ \frac{\partial R_N}{\partial y_N}\ -\ 3.0\right] \qquad (5.21c)$$

$$\frac{\partial G_N}{\partial v_N} = 4L_{N-1}(FR)(FAC)v_N/R_N^{\ 4/3}$$

where

$FAC = \theta\Delta t$;

$\frac{\partial R}{\partial y}$ = rate of change of hydraulic radius with depth.

For a natural channel, $R = A/P$ (area, A, divided by the wetted perimeter, P), then

$$\partial R/\partial y = \frac{(P\frac{\partial A}{\partial \cancel{R}y} - A\frac{\partial P}{\partial \cancel{A}y})}{P^2}$$

For a rectangular channel, $R = By/B+2y$, and

$$\frac{\partial R}{\partial y} = \frac{B^2}{(B+2y)^2}$$

The solution of Equation (5.20) is initiated iteratively by evaluating the right hand column vector and the Jacobian matrix using the prevous nodal values of depth, y, and velocity, v. Good starting values for nodal depths and velocities are those of the uniform flow before the flood wave arrives at the upstream section of the channel.

At the upstream boundary node, if the flood discharge hydrograph is imposed as a known condition, then the corresponding upstream velocity, v_1, at any time level is evaluated as:

$$v_1 = Q_1/By_1 \qquad (5.22)$$

On the other hand, the simplified momentum equation adopted for the diffusion model adequately describes the downstream boundary condition as a loop rating curve. Thus, no further modification is necessary.

Weighted Implicit Complete Finite Element Model, WICFEM

Solution of the complete flow model follows the same basic steps as the implicit diffusion flow model. The significant difference between the two models resides in the total number of terms in the momentum Equation (3.2). For this reason, only the manipulation of the complete momentum equation deserves further discussion. The finite element transformed version of the complete momentum Equation (3.2) is presented in Chapter IV as Equation (4.12).

The assembled matrix equation (Equation 4.12), together with the dimensionless time weighting factor, θ, and forward time differencing yields:

$$
\begin{aligned}
&[D + (\theta\Delta t)E]\{v^{n+1}\} + \theta\Delta t[F]\{y^{n+1}\} + \theta\Delta t[G]\{\frac{v^{n+1}}{y^{n+1}}\} + \theta\Delta t[D]\{S_f{}^{n+1}\} \\
&- \theta\Delta t\{M^{n+1}\} = [D - \Delta t(1-\theta)E]\{v^n\} - \Delta t(1-\theta)[F]\{y^n\} - \Delta t(1-\theta)[G]\{\frac{v^n}{y^n}\} \\
&- \Delta t(1-\theta)[D]\{S_f{}^n\} + \Delta t(1-\theta)\{M^n\} \qquad (5.23)
\end{aligned}
$$

where

D, E, F and D = assembled matrices of Eq. 4.12 from left to right, respectively;

M = assembled vector of Eq. 4.12.

Letting $S_f = FR\ v^2/R^{4/3}$, where $FR = g\ n1^2/2.2082$,

then Equation (5.23) becomes:

$$[D + (\theta\Delta t)E]\{v^{n+1}\} + \theta\Delta t[F]\{y^{n+1}\} + \theta\Delta t[G]\{\frac{v^{n+1}}{y^{n+1}}\} + \theta\Delta t\left(\frac{FR}{(R^{n+1})^{4/3}}\right)[D]$$

$$\{v^{n+1}\}^2 - \theta\Delta t\{M^{n+1}\} = [D - \Delta t(1-\theta)E]\{v^n\} - \Delta t(1-\theta)[F]\{y^n\} - \Delta t(1-\theta)[G]$$

$$\{\frac{v^n}{y^n}\} - \Delta t(1-\theta)\,\frac{FR}{(R^n)^{4/3}}\,[D]\{v^n\}^2 + \Delta t(1-\theta)\{M^n\} \tag{5.24}$$

The expanded forms of Equation (5.24) for the upstream, interior and downstream nodes are given as:

$$G_1 \equiv \Big[(4L_1+\theta\Delta t(-2v_1-v_2))v_1 + (2L_1+\theta\Delta t(-v_1-2v_2))v_2\Big]^{n+1}$$

$$+ 2\theta\Delta t\left[2L_1q_1\frac{v_1}{y_1} + L_1q_1\frac{v_2}{y_2}\right]^{n+1} + 2\theta\Delta t\left[2L_1\,\frac{FR}{R_1^{4/3}}\,v_1^{\,2} + L_1\,\frac{FR}{R_2^{4/3}}\,v_2^{\,2}\right]^{n+1}$$

$$+ \theta\Delta t\Big[6g(y_2-y_1)\Big]^{n+1} - \theta\Delta t\Big[6gS_oL_1\Big]^{n+1} + \Big[(-4L_1+\Delta t(1-\theta)(-2v_1-v_2))v_1$$

$$+ (-2L_1+\Delta t(1-\theta)(-v_1-2v_2))v_2\Big]^n + 2\Delta t(1-\theta)\left[2L_1q_1\frac{v_1}{y_1} + L_1q_1\frac{v_2}{y_2}\right]^n$$

$$+ 2\Delta t(1-\theta)\left[2L_1\,\frac{RF}{R_1^{4/3}}\,v_1^{\,2} + L_1\,\frac{FR}{R_2^{4/3}}\,v_2^{\,2}\right]^n + \Delta t(1-\theta)\Big[6g(y_2-y_1)\Big]^n$$

$$- \Delta t(1-\theta)\Big[6gS_oL_1\Big]^n = 0 \tag{5.25a}$$

$$G_i \equiv \Big[(2L_{i-1}+\theta\Delta t(2v_{i-1}+v_i))v_{i-1} + (4(L_{i-1}+L_i)+\theta\Delta t(v_{i-1}-v_{i+1}))v_i$$

$$+ (2L_i+\theta\Delta t(-v_i-2v_{i+1}))v_{i+1}\Big]^{n+1} + 2\theta\Delta t\Big[q_{i-1}L_{i-1}\,\frac{v_{i-1}}{y_{i-1}} + 2(L_{i-1}q_{i-1}+L_iq_i)$$

$$\frac{v_i}{y_i} + q_iL_i\,\frac{v_{i+1}}{y_{i+1}}\Big]^{n+1} + 2\theta\Delta t\Big[L_{i-1}\,\frac{FR}{R_{i-1}^{4/3}}\,v_{i-1}^{2} + 2(L_{i-1}+L_i)\,\frac{FR}{R_i^{4/3}}\,v_i^{\,2}$$

$$+ L_i\,\frac{FR}{R_{i+1}^{4/3}}\,v_{i+1}^{2}\Big]^{n+1} + \theta\Delta t\Big[6g(y_{i+1}-y_{i-1})\Big]^{n+1} - \theta\Delta t\Big[6gS_o(L_{i-1}+L_i)\Big]^{n+1}$$

$$+ \Big[(-2L_{i-1}+\Delta t(1-\theta)(2v_{i-1}+v_i))v_{i-1} + (-4(L_{i-1}+L_i)+\Delta t(1-\theta)(v_{i-1}-v_{i+1}))v_i$$

$$+ (-2L_i+\Delta t(1-\theta)(-v_i-2v_{i+1}))v_{i+1}\Big]^n + 2\Delta t(1-\theta)\Big[q_{i-1}L_{i-1}\,\frac{v_{i-1}}{y_{i-1}}$$

$$+ 2(L_{i-1}q_{i-1}+L_iq_i)\,\frac{v_i}{y_i} + q_iL_i\,\frac{v_{i+1}}{y_{i+1}}\Big]^n + 2\Delta t(1-\theta)\Big[L_{i-1}\frac{FR}{R_{i-1}^{4/3}}\,v_{i-1}^2$$

$$+ 2(L_{i-1}+L_i)\frac{FR}{R_i^{4/3}}\,v_i{}^2 + L_i\frac{FR}{R_{i+1}^{4/3}}\,v_{i+1}^2\Big]^n + \Delta t(1-\theta)\Big[6g(y_{i+1}-y_{i-1})\Big]^{n+1}$$

$$- \Delta t(1-\theta)\Big[6gS_o(L_{i-1}+L_i)\Big]^n = 0 \qquad (5.25b)$$

$$G_N \quad \Big[(2L_{N-1}+\theta\Delta t(2v_{N-1}+v_N))v_{n-1} + (4L_{N-1}+\theta\Delta t(v_{N-1}+2v_N))v_N\Big]^{n+1}$$

$$+ 2\theta\Delta t\Big[L_{N-1}q_{N-1}\,\frac{v_{N-1}}{y_{N-1}} + 2L_{N-1}q_{N-1}\,\frac{v_N}{y_N}\Big]^{n+1} + 2\theta\Delta t\Big[L_{N-1}\frac{FR}{R_{N-1}^{4/3}}\,v_{N-1}^2$$

$$+ 2L_{N-1}\frac{FR}{R_N^{4/3}}\,v_N{}^2\Big]^{n+1} + \theta\Delta t\Big[6g(y_N-y_{N-1})\Big]^{n+1} - \theta\Delta t\Big[6gS_oL_{N-1}\Big]^{n+1}$$

$$+ \Big[(-2L_{N-1}+\Delta t(1-\theta)(2v_{N-1}+v_N))v_{N-1} + (-4L_{N-1}+\Delta t(1-\theta)(v_{N-1}+2v_N))v_N\Big]^n$$

$$+ 2\Delta t(1-\theta)\Big[L_{N-1}q_{N-1}\frac{v_{N-1}}{y_{N-1}} + 2L_{N-1}q_{N-1}\frac{v_N}{y_N}\Big]^n + 2\Delta t(1-\theta)\Big[L_{N-1}\frac{FR}{R_{N-1}^{4/3}}\,v_{N-1}^2$$

$$+ 2L_{N-1}\frac{FR}{R_N^{4/3}}\,v_N{}^2\Big]^n + \Delta t(1-\theta)\Big[6g(y_N-y_{N-1})\Big]^n - \Delta t(1-\theta)\Big[6gS_oL_N-1\Big]^n \qquad (5.25c)$$

By replacing all the Jacobian terms associated with the momentum equation for the diffusion flow model in Equation (5.20) with those of the complete flow model, the solution thereafter follows the same routine. However, it is possible to modify the downstream boundary condition for the momentum equation similar to the diffusion flow model as an adequate loop rating curve, (Equation 15.17c). The upstream momentum Equation (5.25a) needs no modification if Equation (5.22) is employed to update the upstream velocity, v_1.

The required Jacobian-momentum terms to replace those in Equation (5.20) are as follows:

$$\begin{aligned}
\frac{\partial G_1}{\partial y_1} &= -FAC\left[\frac{16(FR)L_1v_1^{\,2}}{3R_1^{\,7/3}}\;\frac{\partial R_1}{\partial y_1} + \frac{4L_1q_1v}{y_1^{\,2}} + 6g\right] \\
\frac{\partial G_1}{\partial v_1} &= 4L_1 + FAC\left[(-4v_1-2v_2) + \frac{8L_1(FR)v_1}{R_1^{\,4/3}} + \frac{4L_1q_1}{y_1}\right] \\
\frac{\partial G_1}{\partial y_2} &= -FAC\left[\frac{8(FR)L_1v_2^{\,2}}{3R_2^{\,7/3}}\;\frac{\partial R_2}{\partial y_2} + \frac{2L_1q_1v_2}{y_2^{\,2}} + 6g\right] \\
\frac{\partial G_1}{\partial v_2} &= 2L_1 + FAC\left[(-2v_1-4v_2) + \frac{4L_1(FR)v_2}{R_2^{\,4/3}} + \frac{2L_1q_1}{y_2}\right]
\end{aligned} \tag{5.26a}$$

$$\begin{aligned}
\frac{\partial G_i}{\partial y_{i-1}} &= -FAC\left[\frac{8(FR)L_{i-1}v_{i-1}^{\,2}}{3R_1^{\,7/3}}\;\frac{\partial R_{i-1}}{\partial y_{i-1}} + \frac{2L_{i-1}q_{i-1}v_{i-1}}{y_{i-1}^{\,2}} + 6g\right] \\
\frac{\partial G_i}{\partial v_{i-1}} &= 2L_{i-1} + FAC\left[(4v_{i-1}+2v_i) + \frac{4L_{i-1}(FR)v_{i-1}}{R_{i-1}^{4/3}} + \frac{2L_{i-1}q_{i-1}}{y_{i-1}}\right] \\
\frac{\partial G_i}{\partial y_i} &= -FAC\left[\frac{16(FR)(L_{i-1}+L_i)v_i^{\,2}}{3R_i^{\,7/3}}\;\frac{\partial R_i}{\partial y_i} + \frac{4(L_{i-1}q_{i-1}+L_iq_i)v_i}{y_i^{\,2}}\right] \\
\frac{\partial G_i}{\partial v_i} &= 4(L_{i-1}+L_i) + FAC\left[2(v_{i-1}-v_{i-1}) + \frac{8(L_{i-1}+L_i)(FR)v_i}{R_i^{\,4/3}} + \frac{4(L_{i-1}q_{i-1}+L_iq_i)}{y_i}\right] \\
\frac{\partial G_i}{\partial y_{i+1}} &= -FAC\left[\frac{8(FR)L_iv_{i+1}^{\,2}}{3R_{i+1}^{7/3}}\;\frac{\partial R_{i+1}}{\partial y_{i+1}} + \frac{2L_iq_iv_{i+1}}{y_{i+1}^{\,2}} - 6g\right] \\
\frac{\partial G_i}{\partial v_{i+1}} &= 2L_i+FAC\left[(-2v_i-4v_{i+1})+\frac{4L_i(FR)v_{i+1}}{R_{i+1}^{4/3}} + \frac{2L_iq_i}{y_{i+1}}\right]
\end{aligned} \tag{5.26b}$$

$$\frac{\partial G_N}{\partial y_{N-1}} = -FAC\left[\frac{8(FR)L_{N-1}v_{N-1}^{\,2}}{3R_{N-1}^{7/3}}\;\frac{\partial R_{N-1}}{\partial y_{N-1}} + \frac{2L_{N-1}q_{N-1}v_{N-1}}{y_{N-1}^{\,2}} + 6g\right] \tag{5.26c}$$

$$\frac{\partial G_N}{\partial v_{N-1}} = 2L_{N-1} + \text{FAC}\left[(4v_{N-1}+2v_N)+\frac{4L_{N-1}(\text{FR})v_{N-1}}{R_{N-1}^{4/3}}+\frac{2L_{N-1}q_{N-1}}{y_{N-1}}\right]$$

$$\frac{\partial G_N}{\partial y_N} = -\text{FAC}\left[\frac{16(\text{FR})L_{N-1}v_N^{2}}{3R_N^{7/3}}\ \frac{\partial R_N}{\partial y_N^{2}}+\frac{4L_{N-1}q_{N-1}v_N}{y_N^{2}}-6g\right] \tag{5.26c}$$

$$\frac{\partial G_N}{\partial v_N} = 4L_{N-1}+\text{FAC}\left[(2v_{N-1}+4v_N)+\frac{8L_{N-1}(\text{FR})v_N}{R_N^{4/3}}+\frac{4L_{N-1}q_{N-1}}{y_N}\right]$$

Similar to the weighted implicit kinematic and diffusion flow models, the Jacobian terms of Equation (5.26) are up-dated after every three iterations. The numerical performance of the complete flow model and simplified models in predicting the depth of flow and velocity of flow is the subject of Chapters VI and VII.

CHAPTER VI

VERIFICATION OF MODELS

Introduction

Although the use of numerical methods for unsteady flow investigations has increased tremendously in recent years, most of the investigations are still exploratory, and serious attempts at making them accessible to the users in the field have not yet been made. The responsibility for developing an efficient numerical model that needs minimal or no modification, except the insertion of input data, is partly the objective of this study. However, further testing of the models with different problems and with a variety of boundary conditions is necessary for general use.

The performance of each model, particularly the weighted implicit diffusion and complete flow models in predicting flows in a natural channel, was assessed by comparing simulated and observed hydrographs. Possible discrepancies between simulated and observed flows are attributable to the following sources: errors in field measurements of the flows, survey errors in the measurement of channel sections, errors in estimating resistance coefficients, and, most importantly, changes in the channel properties before and during the unsteady flow event (Amein and Fang, 1969). Other sources of errors are associated with the numerical method itself, namely: finite element approximation errors, temporal approximation errors, and errors due to any iterative non-linear equation solver (Chung, 1978). To eliminate the contribution of the first kind of sources of errors, it was necessary to verify the models in two parts.

The first part involves simulation of flow in an idealized channel of rectangular geometry. Simulated hydrographs were compared with similar results from those predicted using an explicit finite difference scheme (Viessman et al., 1972). This approach helps to explore the basic principles in the numerical development of the individual models. As a result, any discrepancies observable on application to natural channels should not be all blamed on the mathematical model development. The second part involves simulation of a flood in a natural channel, the Illinois River located in Oklahoma. Limited data in other major streams in Oklahoma, mostly cross sections, roughness coefficients and recorded floods made the Illinois River the best choice.

Application to Idealized Channel

The computer programs of the explicit, weighted implicit kinematic, diffusion, and complete flow models have been written in FORTRAN IV for an IBM 360 model 75. The models were applied separately to simulate the hypothetical flood in a rectangular channel presented by Viessman et al. (1972) using the explicit finite difference scheme. The example problem considers a 2-mile long and 2-ft wide rectangular channel having a depth of flow of 6 ft. It is subjected to an upstream increase in flow to 2000 cfs in a period of 20 minutes, and then it decreases uniformly to the initial depth of flow in an additional period of 40 minutes. The channel has a bottom slope of 0.0015 ft/ft and an estimated Manning coefficient, n, of 0.02.

Similar to the example of Viessman et al. (1972), a distance step of 528 ft was used in the simulation, although the four models can accept variable distance steps.. Also, the weighted implicit kinematic model has a built-in option to route the flood in a trapezoidal, triangular, or rectangular channel. For the first two geometries, the right- and left-side slopes captioned as ZRS and ZLS should have assigned values other than zeros, except for a rectangular channel. The triangular geometry will have zero width for input value.

Hydrographs from EKFEM and WIKFEM

The flow hydrographs for upstream, midreach, and downstream sections are predicted by the explict and weighted implict kinematic models. These hydrographs are plotted along with those predicted by Viessman et al. (1972), shown in Figure 6. It should be noted that the explicit difference scheme of Viessman et al. (1972) solves the continuity and momentum equations completely. The kinematic flow models depict attenuations in the peak flows at midreach as well as the downstream section. This performance is acceptable since the longitudinal channel slope utilized for the simulation falls within 10 feet per mile (10.3%) for which the use of kinematic approximation is justified. Details of the slope approximation for use of simplified models are discussed by Henderson (1966).

While the explicit kinematic finite element model is limited to a time step of 2 seconds because of stability considerations, the weighted implicit scheme appears to be unconditionally stable. The influence of the time weighting factor, θ, on the numerical distortion (dispersion and

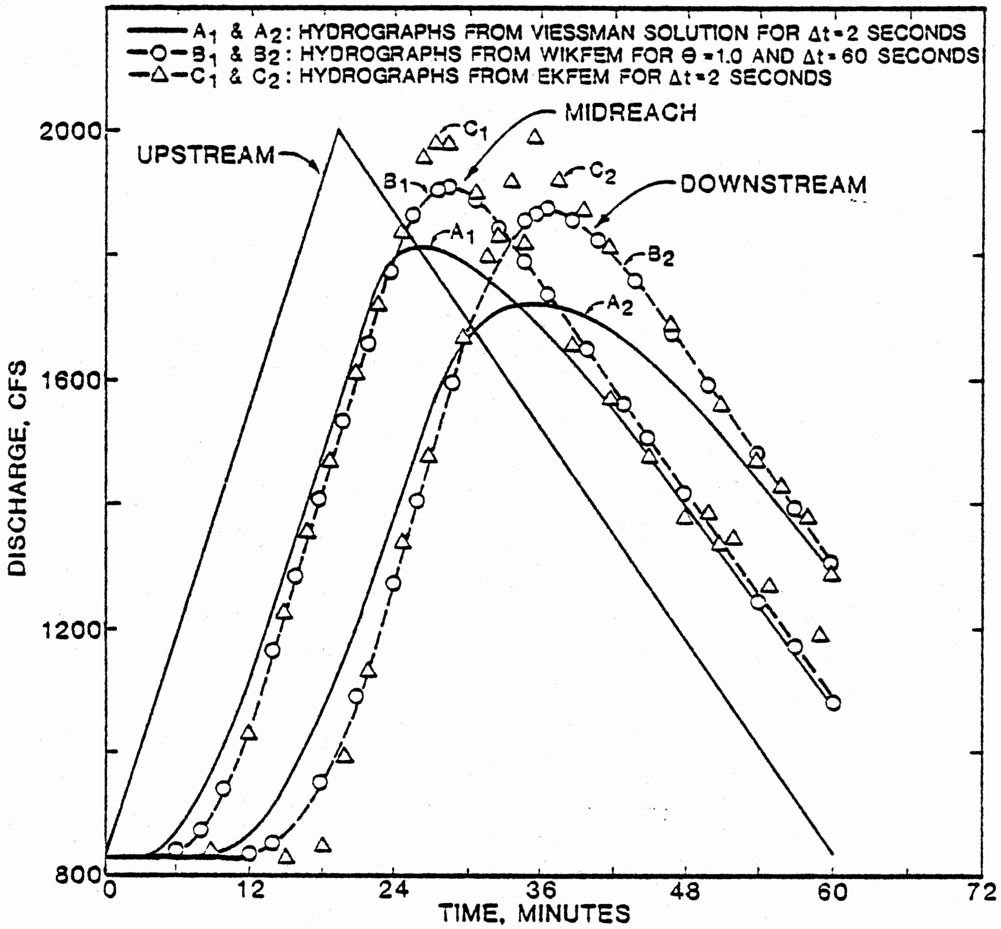


Figure 6. Comparison of Dishcarge Hydrographs for EKFEM and WIDFEM and Viessman's Solution.

attenuation of computed stage or discharge hydrographs) is shown in Figure 7. The plotted discharge hydrograph indicates that the lower range of the allowable θ values, such as 0.75, as compared to the upper limiting value of 1.00 minimized the attenuation of the peak flowwhich results from the use of a large time step, Δt of 300 seconds. This observation is not unique but confirms that of Fread (1973). The weighted implicit kinematic flow model was run for Δt = 180, 300, and 600 seconds with various values of the weighting factor θ, such as 0.55, 0.75, and 1.0, respectively. In all time steps, fastest convergence was obtained with θ = 1.0, and it is recommended for use with this routine. It is not surprise that θ = 1.0 affords rapid convergence because the scheme becomes fully implicit. However, instability results in WIKFEM with θ < 0.55. Thus, the allowable range of the time weighting factor, θ, is $0.55 \leq \theta \leq 1.0$.

Hydrographs from WIDFEM

Simulated discharge hydrographs for a time step of 60 seconds and time weighting factors of 0.55 and 1.0, respectively, along with those of Viessman et al. (1972) are compared in Figure 8. The predicted hydrographs denoted as plots B and C in the figure are in close agreement with those of Viessman et al. (1972). However, the slight influence of the time weighting factor in the predicted peak flows at mid-reach and down-stream locations can be observed.

Though the difference in peak flows with θ values of 0.55 and 1.0 is minimal for a time step of 60 seconds, significant differences for larger time steps such as 300 secondsor more are apparent. Figure 9 illustrates very clearly the iteractive effect of the time weighting

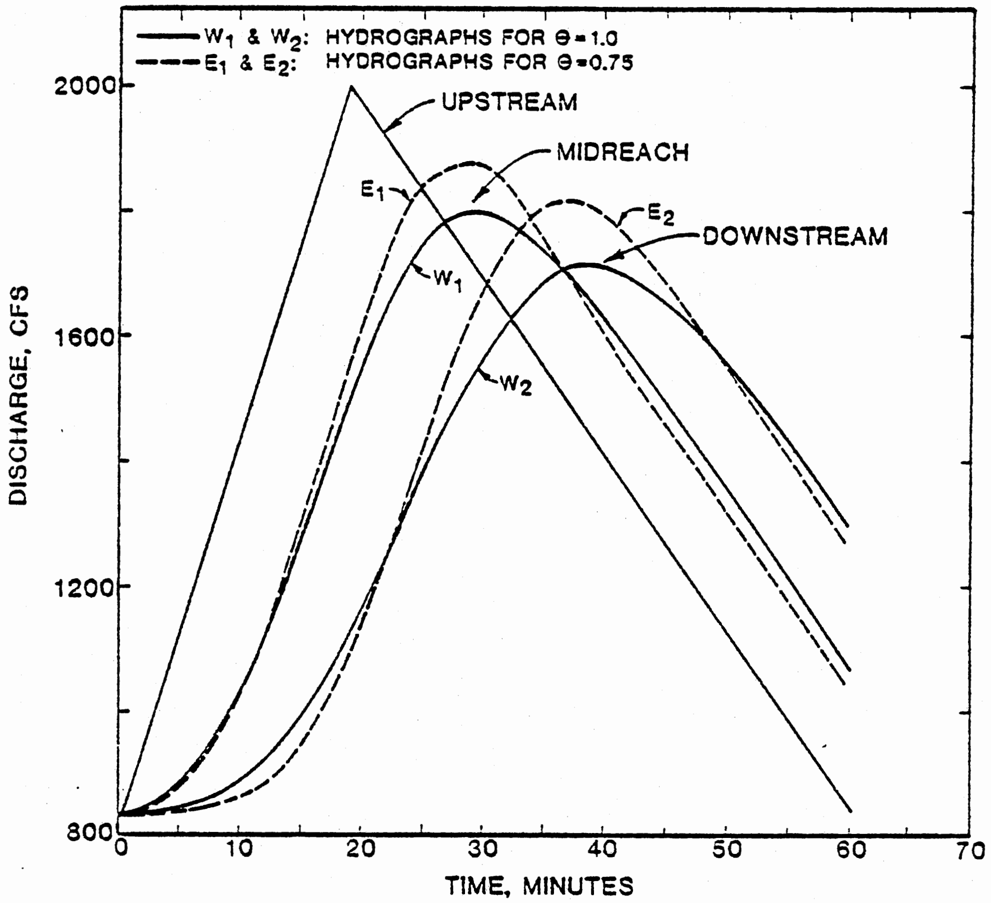


Figure 7. Weighted Implicit Kinematic Finite Element Model Simulation at 300 Seconds for θ of 0.75 and 1.00.

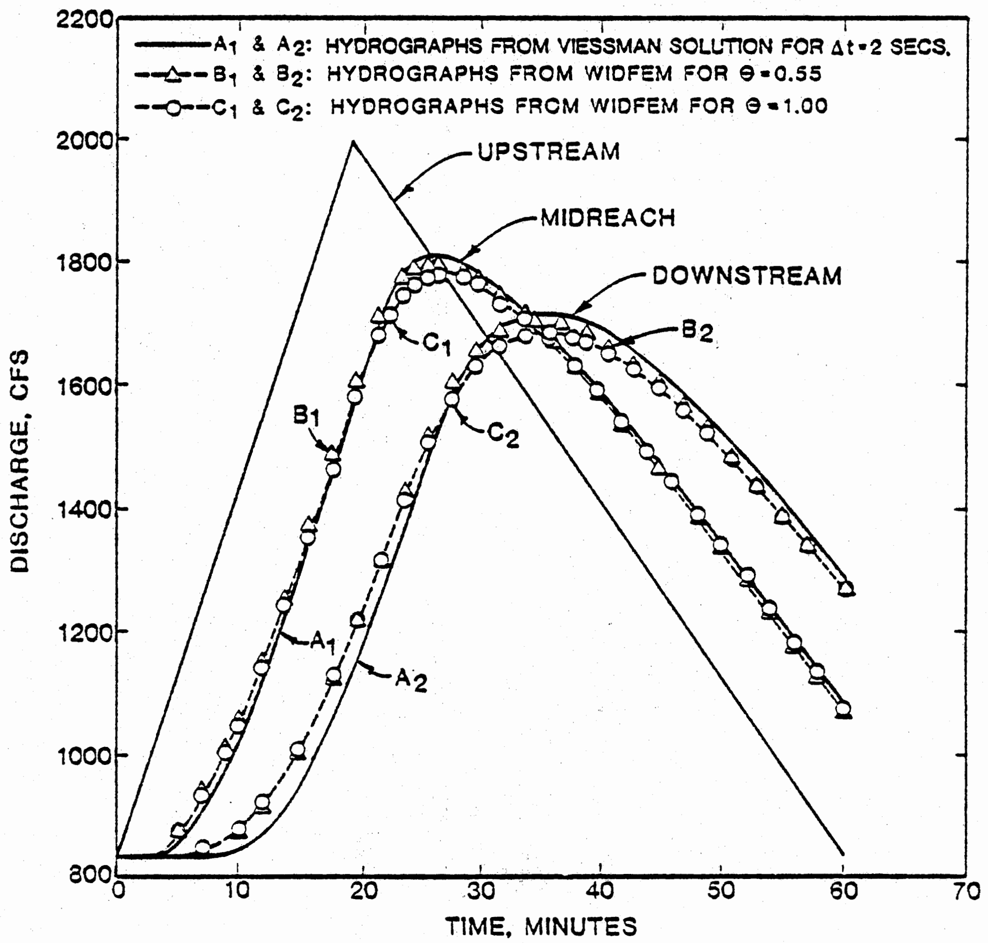


Figure 8. Comparison of Hydrographs from WIDFEM at Δt of 60 Seconds and θ of 0.55 and 1.00 and Viessman's Solution.

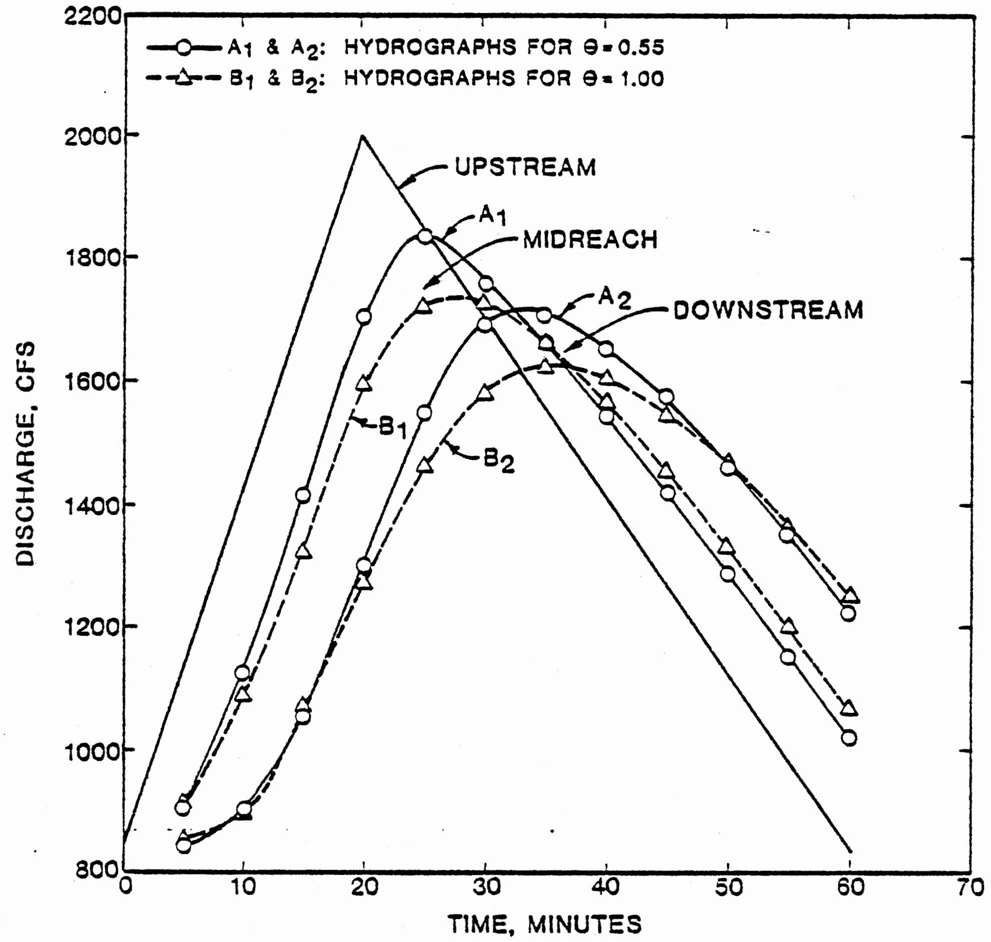


Figure 9. Weighted Implicit Diffusion Finite Element Model Simulation at 300 Seconds for θ of 0.55 and 1.00.

factor, θ, and the numerical dispersion resulting from use of large time steps. Values of θ greater than 0.55 tend to attenuate the peak discharge. This observation equally validates that of the weighted implicit kinematic finite element model discussed earlier.

Like the WIKFEM, the weighted implicit diffusion is unconditionally stable for the time weighting factor in the range of $0.55 \leq \theta \leq 1.0$. In spite of the numerical distortion associated with the use of large time steps, only a θ value of 0.55 predicted hydrographs identical to Viessman et al. (1972). As a result, subsequent simulations of the WIDFEM for time steps large than 60 seconds were executed with a θ value of 0.55.

Hydrographs from WICFEM

Applications of the weighted implicit complete finite element model to the idealized channel using a time step of 60 seconds and a θ value of 0.55 predicted the discharge hydrographs shown in Figure 10. Hydrograph results are identical to those of Viessman et al. (1972) on the rising limbs but differ slightly on the receding limbs. On the average this difference is insignificant. The WICFEM affords an unconditionally stable solution for the time weighting factor in the range of $0.55 \leq \theta \leq 1.0$. Also the model shares the same basic characteristic as the WIDFEM discussed earlier.

Flow Simulation in a Natural Channel

The second test analyzed flow through a natural river channel. The Illinois River between Watts and Tahlequah gaging stations (Sta. 1955 and 1965, respectively) in Oklahoma, shown in Figure 11, was chosen.

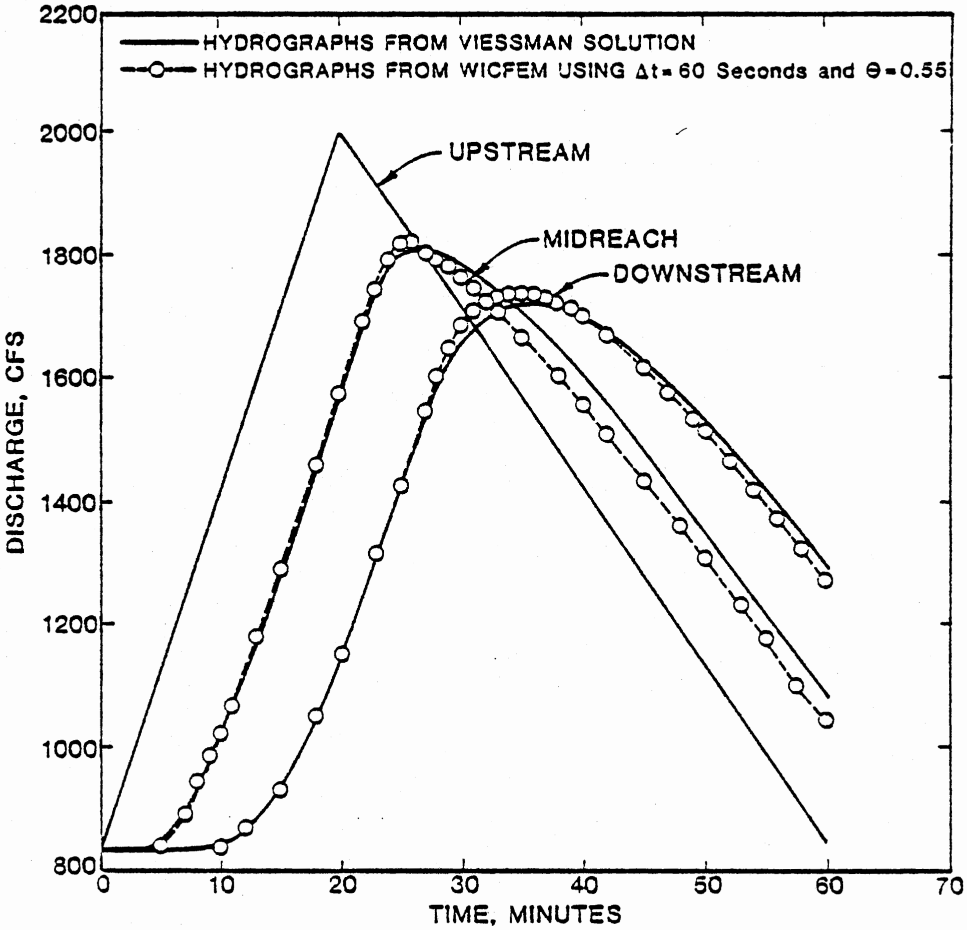


Figure 10. Comparison of Hydrographs from WICFEM at Δt of 60 Seconds and θ of 0.55 and Viessman's Solution.

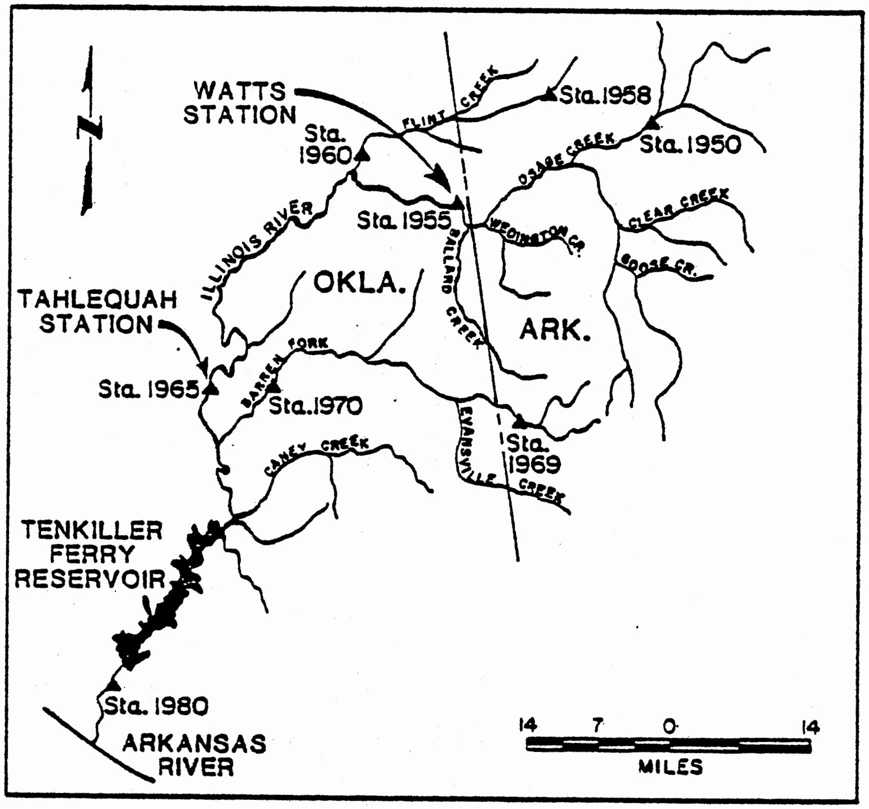


Figure 11. Map of Illinois River Basin.

Geometric cross-sectional data (developed from topographic maps) were collected from Weigant (1982) along with the flood data of April 10, 1979, and were used in predicting the flow hydrograph at the Tahlequah station. Figure 11 shows the Illinois River and locations of the stations.

Owing to the nature of the available topographic data, simulation was executed using a composite channel section. The changes from section to section in some locations are significant enough that smaller distance steps are necessary to adequately represent them in the model. Thus, the channel sections were averaged with a single longitudinal bottom slope of 4.5 feet per mile (Weigant, 1982).

Initial and Boundary Conditions

Initial depths of flow were generated by backwater calculation starting from a downstream depth. Discharge values at intermediate nodes were estimated by linear interpolation applied to the two initial discharges at up-and down-stream locations. Nodal velocities corresponding to initial depths are calculated by dividing the nodal discharge by corresponding cross section. At the upstream point, the discharge was prescribed as a function of time. At the downstream boundary, a loop rating curve was imposed.

For the 1979 flood, the initial discharge values are given by the unsteady nonuniform flow of 482 cfs at the Watts station and 596 cfs at the Tahlequah station at time $t = o$. The discharge hydrograph at the Watts Station increased from 482 to 22980 cfs in 28 hr and then decreased to 1722 cfs in additional 68 hr. Figures 12 and 13 show the observed discharge hydrographs at Watts and Tahlequah and the rating curves at the

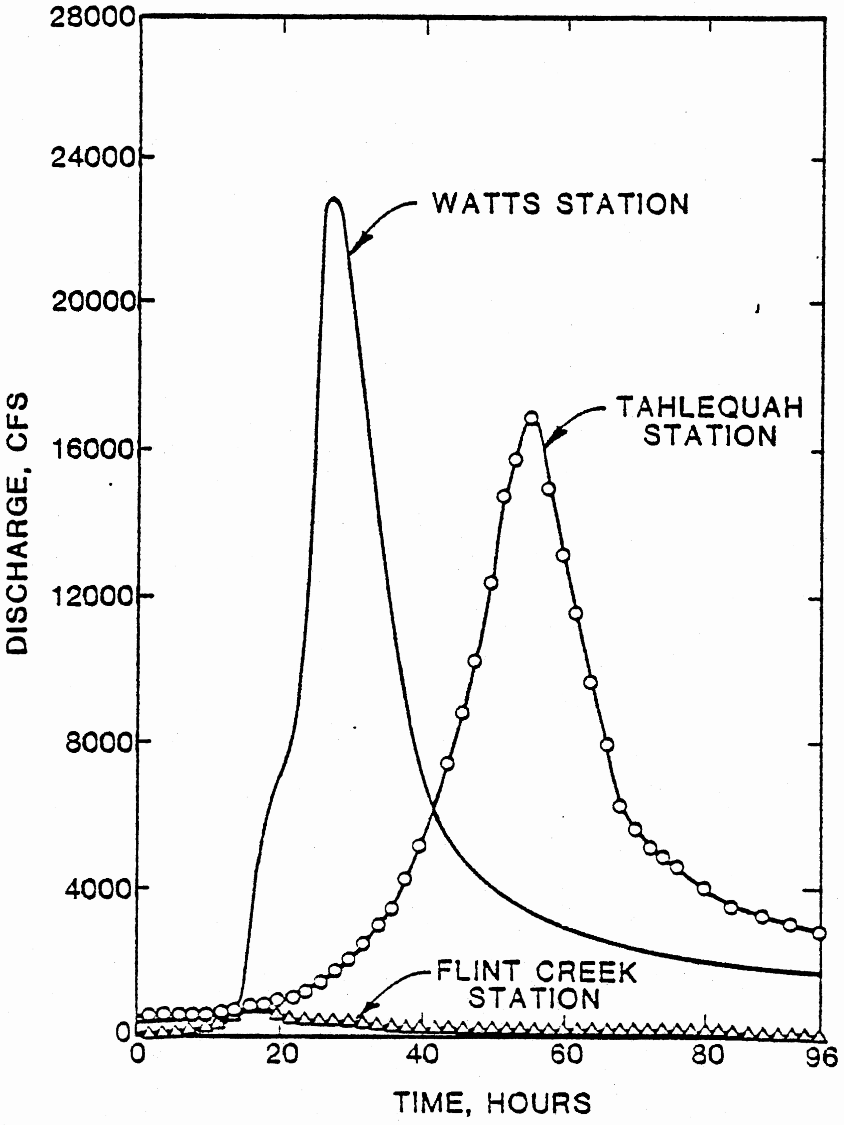


Figure 12. Observed Discharge Hydrographs at Watts, Tahlequah, and Flint Creek Stations for April 10, 1979, Flood, Illinois River, Oklahoma.

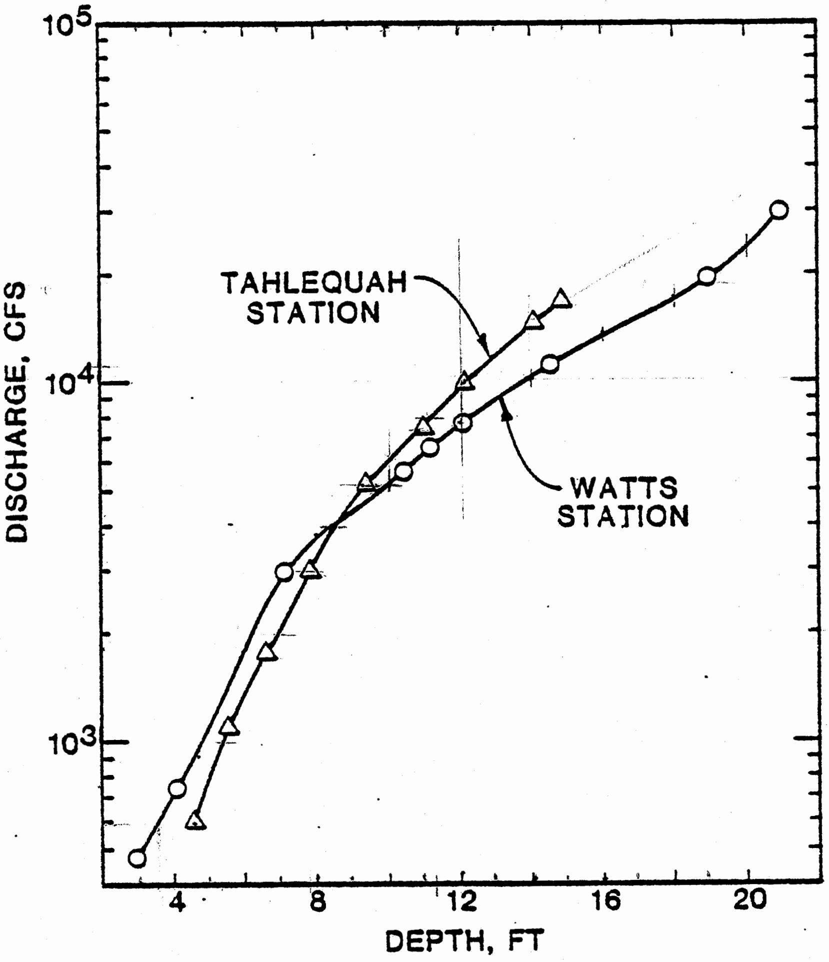


Figure 13. Rating Curves at Watts and Tahlequah, Illinois River, Oklahoma, for April 10, 1979.

stations, respectively. Computed flow at Tahlequah, 50.4 miles from Watts, was compared to the observed flow at the same station.

Determination of Flow Parameters

The flow parameters necessary for simulation in natural channels are the channel cross sections, A; top widths, B; Manning's roughness coefficient, n; and lateral inflow, q.

Average cross-sectional and top width data were utilized to generate a fourth-order polynomial equation, using a least square fitting program (Davis, 1973). A fourth-order polynomial yielded the best fit from the analysis of variance. By increasing the order of the polynomial beyond fourth, it was necessary to see if the increase in the degree of the polynomial significantly improved the fit of the regression. Such statistics as the sum of square due to deviation defined as the difference between total sum of square (SS_T) and sum of square due to regression (SS_R) and the goodness-of-fit defined as SS_R/SS_T were used for assessment. The general form of the equation adopted to model the averaged cross section areas and top width^s is represented as:

$$A(Y) = b_o + b_1 Y + b_2 Y^2 + b_3 Y^3 + b_4 Y^4 \qquad (6.1a)$$

$$B(Y) = C_o + C_1 Y + C_2 Y^2 + C_3 Y^3 + C_4 Y^4 \qquad (6.1b)$$

where A(Y) and B(Y) implies that the area and top width are functions of depth of flow only. Figure 14 illustrates a typical cross section geometry of the Illinois River as given in Equation (6.1). Results are included in the computer sample output.

The initial estimated Manning's roughness coefficient variation

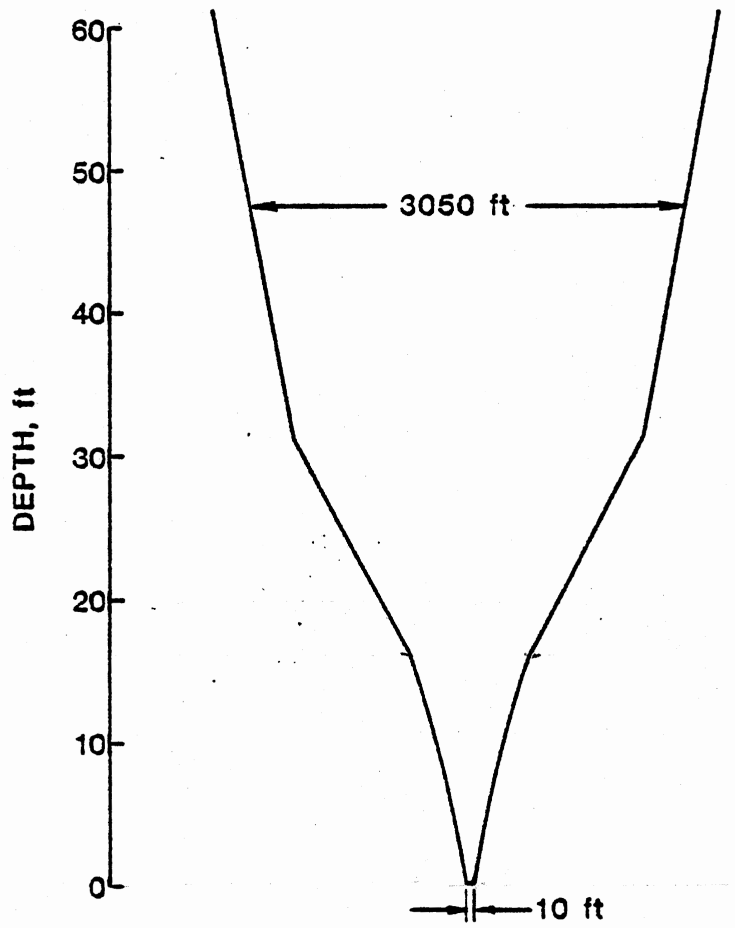


Figure 14. Typical Cross-section of Illinois River, Oklahoma.

against discharge for the Illinois River as provided is plotted in Figure 15. Also shown are the fitted third-order polynomial regression equations of the initial estimated roughness coefficient and of the modified coefficient values. The fitted regression curves are necessary because the plot of the initial estimates of roughness coefficient versus discharge depicts shape variations in some adjoining corners. Thus, a smooth curve was deemed necessary to better represent actual roughness coefficient variations Equation (6.2).

$$n = 0.03713 + 0.14097E\text{-}05Q + 0.41739E\text{-}10Q^2 - 0.23004E\text{-}14Q^3 \qquad (6.2)$$

The modified initial estimates of the roughness coefficient variation are represented as:

$$n = 0.02615 + 0.42801E\text{-}05Q \ -0.21618E\text{-}09Q^2 + 0.39355E\text{-}14Q^3 \qquad (6.3)$$

The lateral inflow hydrograph at Flint Creek, a tributary of the Illinois River 13.2 miles downstream of the Watts Station, recorded during the same date, was imposed as a function of time, Figure 12. The main channel reach corresponding to 13.2 miles from Watts was allowed to receive the lateral inflow from Flint Creek. The inflow is represented in cubic feet per second per area of reach.

Hydrographs from WIDFEM and WICFEM

Application of the flow models to the Illinois River was limited to WIDFEM and WICFEM because of the inherent flat slope of the channel. Use of the kinematic flow models would not be adequately justified in this particular example based on the slope approximation analysis (Henderson, 1966).

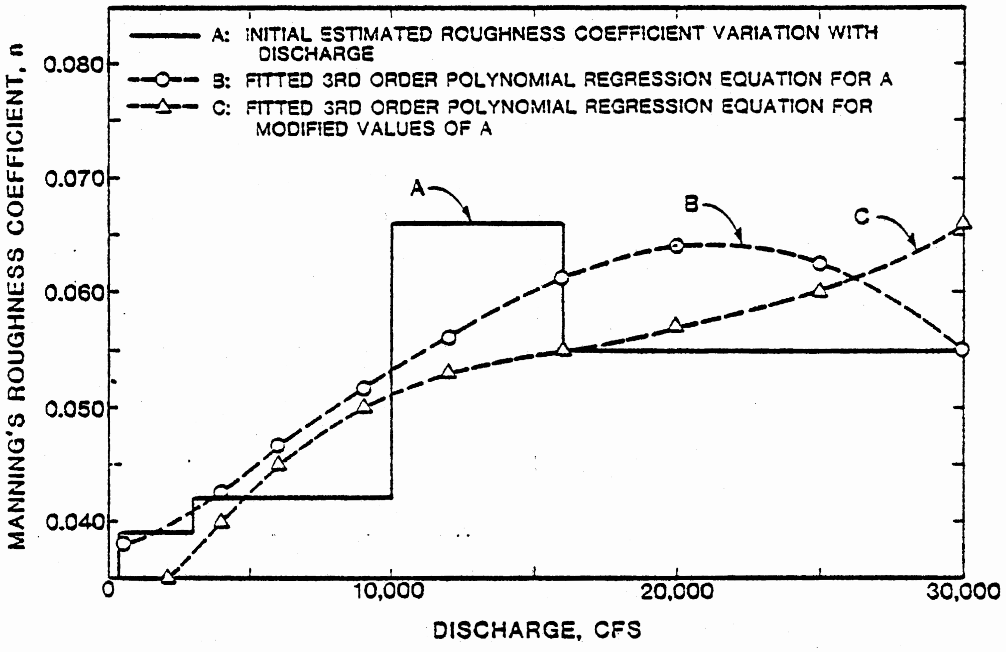


Figure 15. Estimated and Fitted Manning's Roughness Coefficient Variation with Discharge, Illinois River, Oklahoma.

The depth of flow at the Tahlequah Station for the WICFEM using a time step of 30 minutes and a time weighting factor of 0.55 is predicted and compared with that measured in Figure 16. Simulated results are in excellent agreement with observed flow. The marginal difference between computed and actual depths at the early portion of the rising limb and at the tailing edge of the hydrograph reflects the uncertainity of the input data. Among other things, the models are sensitive to variations of the Manning's roughness coefficients in predicting flows. Higer roughness coefficients imply reduced flows and vice versa. Thus, close predictions are possible as long as the roughness coefficient and other input data are accurate.

Figure 16 also illustrates the response of the WICFEM to a modified Manning's roughness coefficient regression Equation (6.3) in predicting depths of flow. The predicted stage hydrograph indicates a slightly high peak, at six hours earlier than the previous prediction using equation (6.2). Indeed, the simulated depths of flow using Equation (6.2) yielded the time of the peak that are more similar to those observed than to those from Equation (6.3). Thus, Equation (6.2) is more representative of the actual roughness variation in the Illinois River. Figure 17 shows the depth of flow predicted using the weighted implicit diffusion finite element model, WIDFEM. The same observations are valid as discussed above using WICFEM. Comparison of the computed flows from WIDFEM and WICFEM using Manning's regression Equation (6.2) against the observed records at the Tahlequah Station is provided in Figures 18 and 19. Discharge hydrographs depict a compounded error of the computed depth of flow and velocity of flow for a given location in the stream. For instance, the difference in peak flows as indicated in Figure 16 is

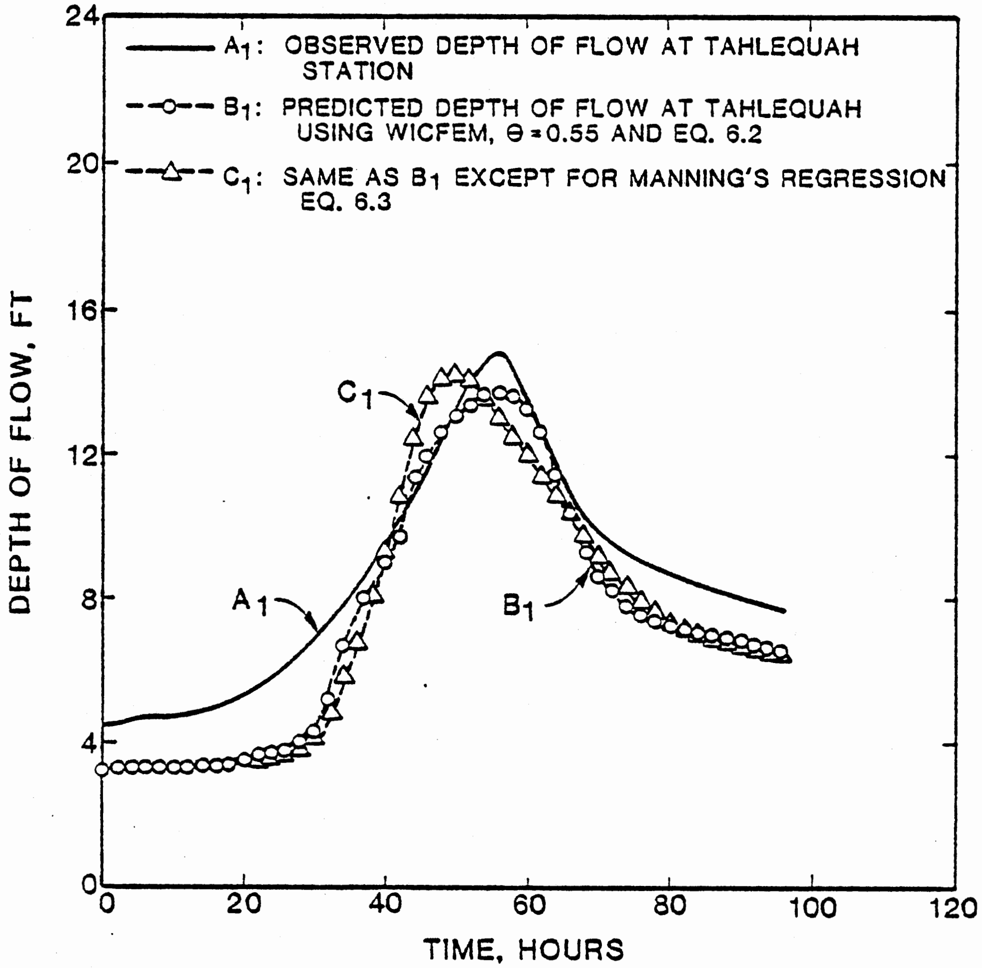


Figure 16. Observed and Predicted Stage Hydrographs at Tahlequah Station from Weighted Implicit Complete Finite Element Model for Δt of 1800 Seconds.

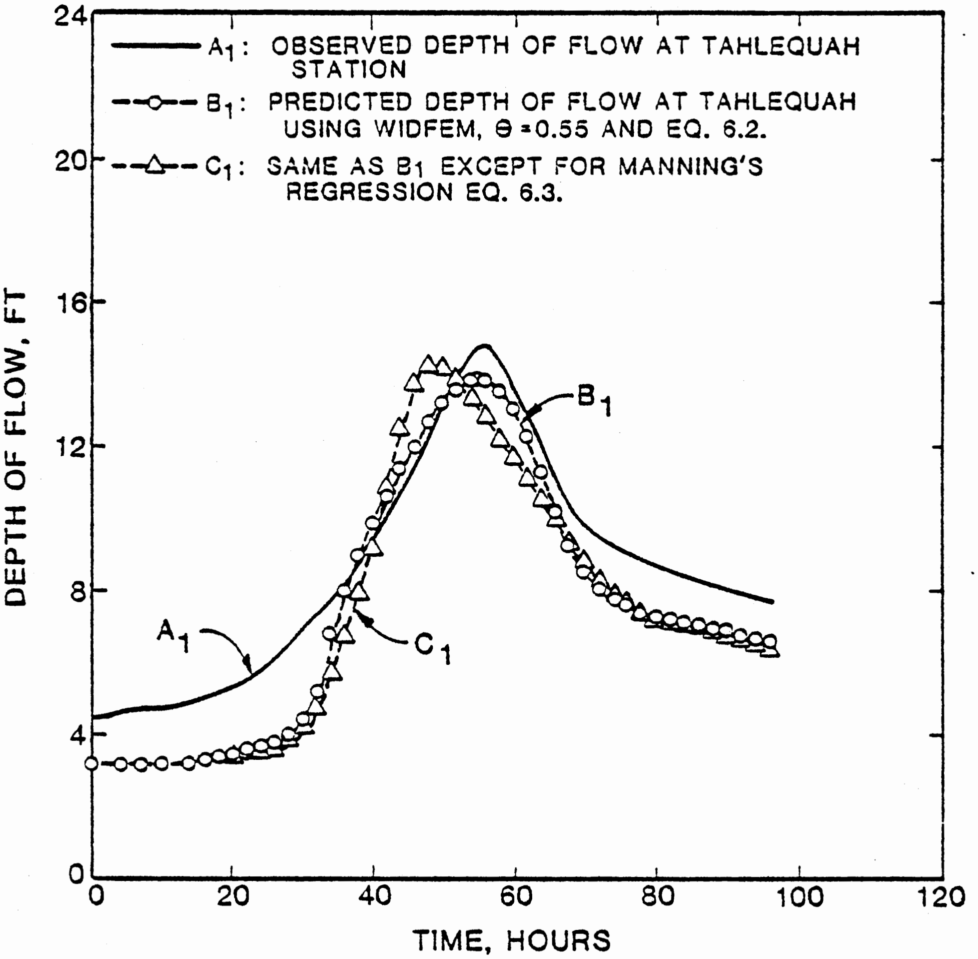


Figure 17. Observed and Predicted Stage Hydrographs at Tahlequah Station from Weighted Implicit Diffusion Finite Element Model for Δt of 1800 Seconds.

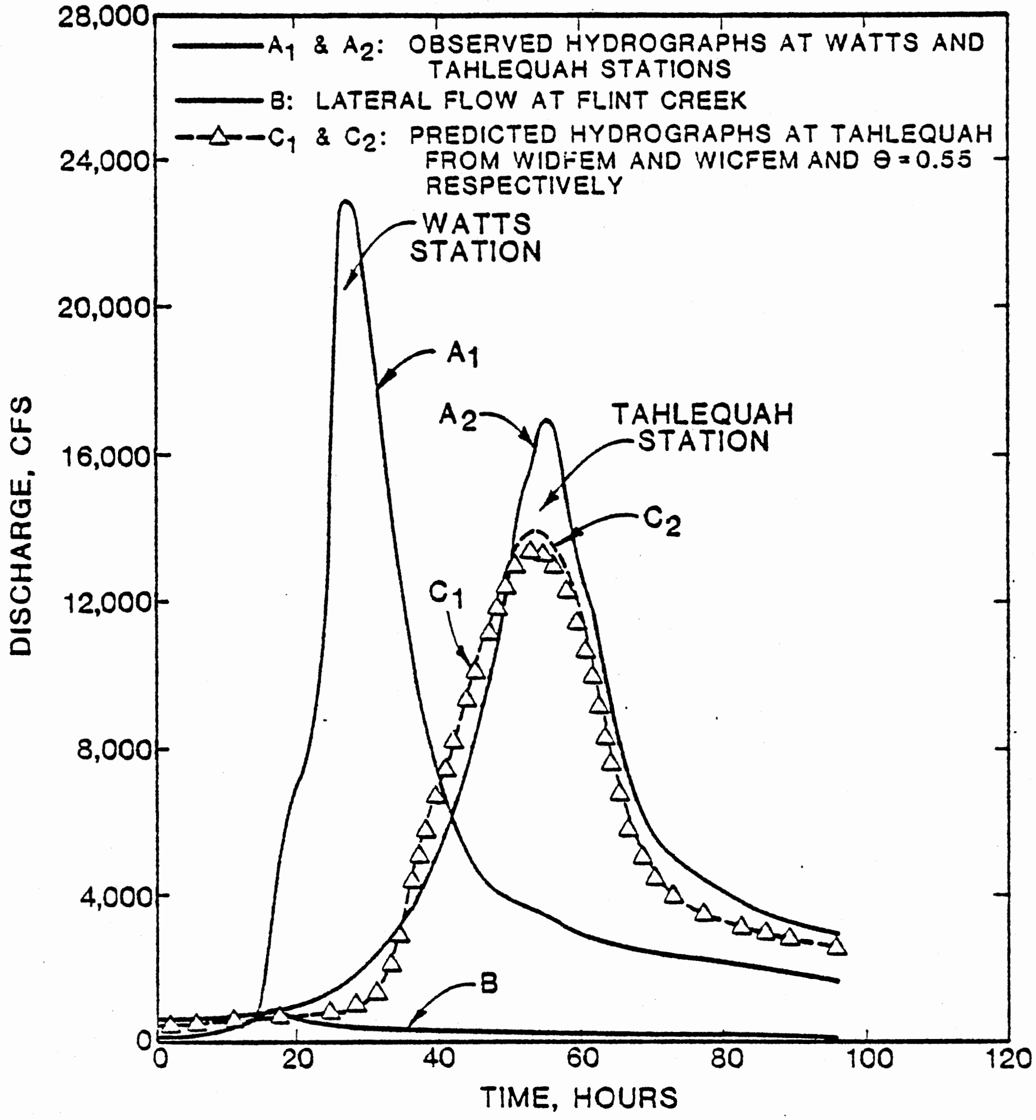


Figure 18. Observed and Simulated Discharge Hydrographs at Tahlequah Station for Δt of 1800 Seconds.

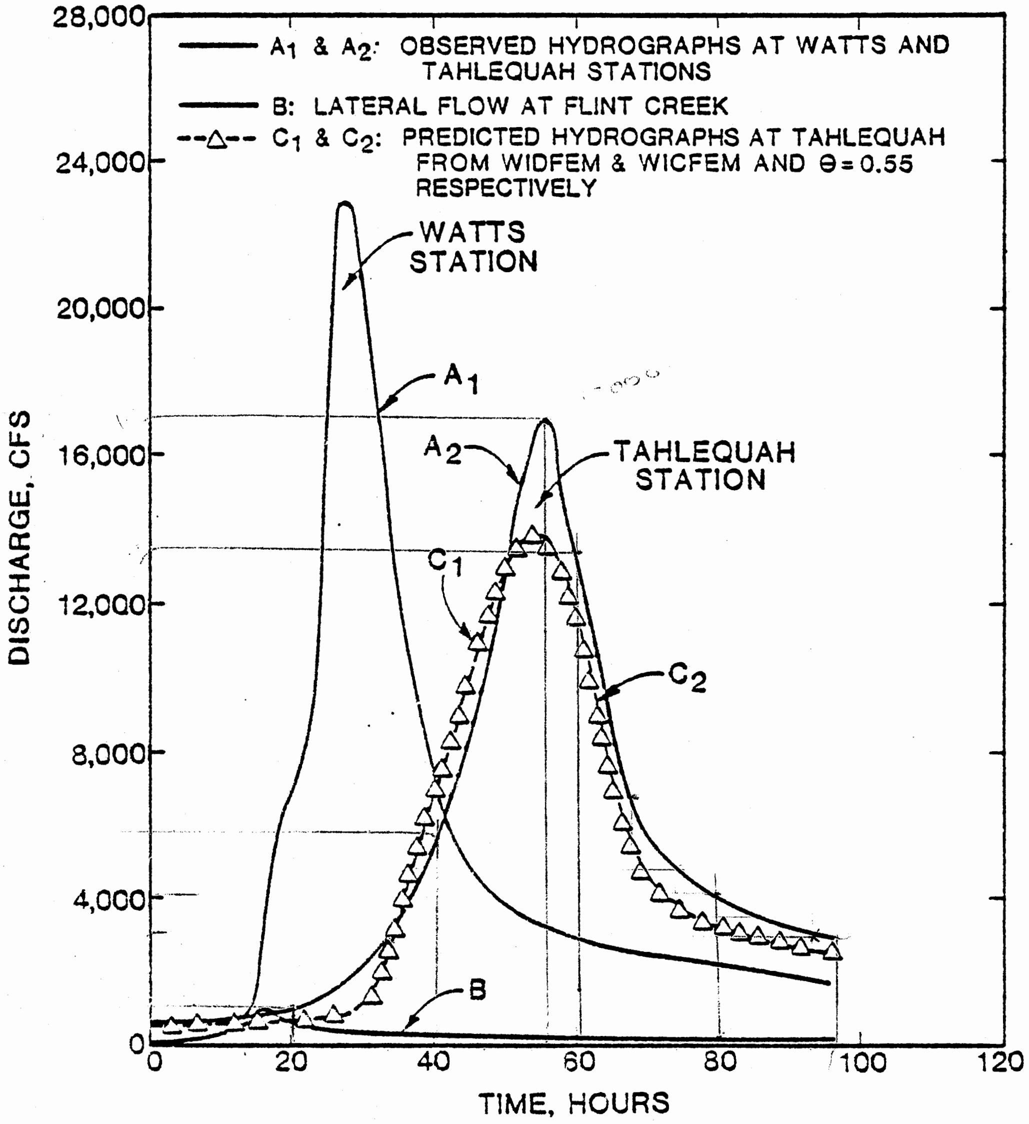


Figure 19. Observed and Simulated Discharge Hydrographs at Tahlequah Station for Δt of 900 Seconds.

about 7% considering prediction with Equation (6.2), while Figure 18 shows an error of 15% for WICFEM. Apparently, the error distribution amongst the predicted depth and velocity as illustrated in the discharge hydrograph are bound to be uneven.

Simulated results of the WIDFEM and WICFEM as shown in Figure 19 for a time step of 15 minutes and weighting factor 0.55 are exactly the same. However, a comparison of the two models for Δt of 30 minutes and θ of 0.55 indicates a slight difference only at the peaks, Figure 18. Invariably, the WICFEM seems to sustain lesser numerical distortion for larger time steps than the WIDFEM. Still, there is much to be gained in the use of WIDFEM. Hydrograph results from WIDFEM are more comparable to those from WICFEM. In addition, the computer time and cost are slightly less for WIDFEM. Appendix K compares computer CPU time and cost of models.

CHAPTER VII

SUMMARY AND CONCLUSIONS

Numerical Performance of Models

The numerical properties of the flow models--EKFEM, WIKFEM, WIDFEM, and WICFEM--such as rate of convergence, accuracy, and stability, need to be assessed through well established mathematical relations. For instance, the Courant condition is employed in the explicit finite difference technique to evaluate the dynamic stability condition arising from the size of the time steps. Since similar conditions in the finite element techniques are not versatile and few in use are formulated under limited assumptions, we are therefore encouraged to draw comparisons from documentations established for the finite difference schemes at least for the time being.

The convergence criterion is a condition in which the solution of the finite element equation for a finite grid size approaches the true solution of the original partial differential equation. For the weighted implicit finite difference scheme proposed by Fread (1974), the convergence criterion was developed by determining the functional form of the truncation error through the Taylor series expansion about the point at which the difference equation is computed. The truncations error, TR, can be expressed as:

$$TR = (2\theta - 1)0(\Delta t) + 0(\Delta t^2) + 0(\Delta x^2) \qquad (7.1)$$

where 0 indicates "order of", and when $\theta = 1$, the truncation error is:

$$TR = 0(\Delta t) + 0(\Delta t^2) + 0(\Delta x^2) \qquad (7.2)$$

Equation (7.2) shows that the fully implicit difference scheme is only first order accurate due to Δt term. However, when $\theta = 0.5$, the error shows a second order accuracy for Δt and Δx.

The WIKFEM, WIDFEM, and WICFEM converge to the true solution for various values of the weighting factor ranging from 0.55 to 1.00. For θ less than 0.55, the models are completely unstable and invariably do not converge. This leads to the concept of numerical stability, defined as a condition whereby the numerical round-off errors introduced in a computational procedure fail to be amplified into an unlimited error. If errors generated at time level $(t + \Delta t)$ are smaller than the errors at time t and not vice versa, the solution is said to be stable.

Stability of the non-linear difference equations of Saint Venant has been investigated by fourier analysis (Fread, 1973, 1974). This analysis is known as the Von Neumann method. In general, results indicate that an implicit difference formulation of the unsteady flow equation is unconditionally stable for any ratio of $\Delta x/\Delta t$, when the weighting factor, θ, is restricted to the range $0.5 < \theta < 1.0$. The analysis proves also that stability of the implicit difference equation does not depend on the ratio $\Delta x/\Delta t$ like the explicit method and method of characteristics. The weighted implicit finite element flow models--WIKFEM, WIDFEM, and WICFEM--are found to be unconditionally stable for the weighting factor in the range of $0.55 \leq \theta \leq 1.0$. However, rapid convergence for weighting factorof unity isobserved only for WIKFEM.

The EKFEM bears similar restrictions as the explicit finite difference scheme. Numerical stability is conditional as defined by the Courant condition. Also, the WIKFEM, WIDFEM, and WICFEM reflect similar numerical properties as the implicit finite difference routine. The concept of explicit and implicit schemes applied to the finite element, FE, and finite difference, FD, formulations tends to tie the FE and FD in the same numerical subset.

Conclusions

Based on the results of the finite element modeling of the streamflow routing for idealized and natural channels, the following conclusions can be drawn:

1. Explicit and weighted implicit kinematic, weighted implicit diffusion, and complete flow models have been developed to predict the velocity of flow, depth of flow, and discharge in a stream.
2. The explicit kinematic finite element model, EKFEM, solves the flow routing problems, having a maximum time step of two seconds.
3. The weighted implicit kinematic finite element model, WIKFEM yields accurate results, with a maximum time interval of ten minutes and weighting factor in a range of 0.55 to 1.00 for a rectangular channel.
4. Both the weighted implicit diffusion and complete finite element models yield accurate and unconditionally stable solutions.

5. All the models--EKFEM, WIKFEM, WIDFEM, and WICFEM--have been tested against a problem presented by Viessman et al. (1972). The comparisons of the flood hydrographs are in close agreement, and the observed difference resides on the speed and stability. In this regard, the weighted implicit models excel.
6. Use of a simplified model such as WIKFEM and WIDFEM in terms of computer storage and cost will be preferred provided good engineering judgement is exercised in their application. For this reason, these models will be favored over the complete solution of the unsteady flow equations.
7. Only the weighted implicit diffusion and complete finite element models were applied to a natural channel, the Illinois River in Oklahoma, for a flood observed on April 10, 1979. Simulated discharge hydrographs at the Tahlequah station, 50.4 miles downstream from the Watts Station, with time steps of 15 and 30 minutes and a weighting factor of 0.55 are in close agreement with the observed flows. A discrepancy of 8 percent in the maximum stage and 15 percent in the maximum discharge is attributed to the degree of accuracy of the input data, especially the roughness coefficient.
8. Not much observable difference exists between the simulated results of the WIDFEM and WICFEM for the natural channel flood routing test. For these particular test results, there is more to be gained in using the simplified diffusion model as discussed earlier in (6) above.

CHAPTER VIII

SUGGESTIONS FOR FUTURE STUDY

The following suggestions for future study would be helpful in using the flow models for predicting the depth, velocity, and volumetric flow rate in a natural channel.

1. Modify the present flow models to incorporate boundary geometry at bridges showing contracting and expanding flow. In addition, field surveys of the hydraulic roughness values for various channel reaches are vital. Variation should be indicated in terms of longitudinal channel distance as well as the depth of flow or volumetric flow rate. Roughness coefficient values imposed on each cross section are usually helpful in locating where a cross section should be subdivided to determine distributed properties. For instance, values of 0.3 and 0.1 are assumed for the expansion and contraction co-coefficients, respectively.
2. Determine what portion of the cross section conveys flow and what portion stores water, particularly for smaller flood events. This might not be necessary for a very large flood wave. For instance, in the present study, it was assumed that the entire cross section conveyed flow for the flood of April 10, 1979, in the Illinois

River. For smaller events, it is assumed that all conveyance occurs in or near the main channel (Thomas, 1975).

3. Study other possible forms of modeling the flow cross sections and the corresponding top width besides using higher order polynomial curve fitting methods. Clearly, there are many possible approaches such as: (a) higher order spline function (cubic spline), (b) logarithmic or exponential regression equations, and (c) simple averaging and interpolation of the input data for intermediate values.

SELECTED BIBLIOGRAPHY

Akan, A.O. and B.C. Yen. 1981. "Diffusion-Wave Flood Routing in Channel Networks," J. Hydraul. Div. Amer. Soc. Civil Eng., Vol. 107, No. HY6, pp. 719-732.

Amein, M. 1966. "Streamflow Routing on Computer by Characteristics," Water Resour. Research, Vol. 2, No. 1, pp. 123-130.

Amein, M. and C.S. Fang. 1969. "Streamflow Routing with Application to North Carolina Rivers," Report No. 17, Water Res. Inst. of the Univ. of North Carolina, Raleigh, N.C.

Amein, M. and C.S. Fang. 1970. "Implicit Flood Routing in Natural Channels," J. Hydraul. Div. Amer. Soc. Civil Eng., Vol. 96, No. HY12, pp. 2481-2500.

Ames, W.F. 1977. Numerical Methods for Partial Differential Equations 1st. Ed. New York: Acad. Press, 350 pp.

Baltzer, R.A. and C. Lai. 1968. "Computer Simulations of Unsteady Flow in Waterways," J. Hydraul. Div. Amer. Soc. Civil Eng., Vol. 96, No. HY4, pp. 1083-1117.

Barre´de Saint-Venant, 1871. "Theory of Unsteady Water Flow, with Application to Floods and to Propagations of Tides in River Channels," French Academy of Science, Vol. 73, pp. 148-154, 237-240. Translated into English by W.W. Geddings, Jr., Waterways Experiment Sta. No. 49-9, Vicksburg, Mississippi, July, 1949.

Chaudhry, Y.M. and D.N. Contractor. 1973. "Application of Implicit Methods to Surges in Channels," Water Resour. Research, Vol. 9, No. 6, pp. 1605-1612.

Chow, V.T. 1959. Open Channel Hydraulics. 1st Ed. New York: McGraw-Hill, 680 pp.

Chow, V.T. ed. 1964. Handbook of Applied Hydrology. 1st Ed. New York: McGraw-Hill, Section 25-II.

Chung, T.J. 1978. Finite Element Analysis in Fluid Dynamics. 1st Ed. New York: McGraw-Hill, 378 pp.

Circiani, T.A., U. Maione, and J.R. Wallis, ed. 1974. "Mathematical Models for Surface Water Hydrology," Proceedings of the Workshop held at the IBM Scientific Center, Pia, Italy. New York: John Wiley & Sons, 423 pp.

Clough, R.W. 1960. "The Finite Element Method in Plane Stress Analysis," *J. Struct. Div. Amer. Soc. Civil Eng.*, Proc. 2nd Conf. Electronic Computation, pp. 345-378.

Cooley, R.L. and S.A. Moin. 1976. "Finite Element Solution of Saint-Venant Equations," *J. Hydraul. Div. Amer. Soc. Civil Eng.*, Vol. 102, No. HY6, pp. 759-776.

Courant, R. 1943. "Variational Methods for the Solution of Problems of Equilibrium and Vibrations," *Bull. Amer. Math. Soc.*, Vol. 49, pp. 1-23.

Davis, J.C. 1973. *Statistics and Data Analysis in Geology*, 1st ed. New York: John Wiley & Sons, 550 pp.

Desai, C.S., K.T. Oden, and L.D. Johnson. 1974. "Evaluation and Analyses of Some Finite Element and Finite Difference Procedures for Time-Dependent Problems," *U.S. Army Eng. Waterw. Exp. Stn.*, Vicksburg, Mississippi, Misc. pap. S-75-7.

Desai, C.S. and R.L. Lytton. 1975. "Stability Criteria for Two Finite Element Schemes for Parabolic Equation," *Int. J. Numer. Method Eng.*, Vol. 9, pp. 721-726.

Desai, C.S. and J.T. Christian. 1977. "Introduction, Numerical Methods and Special Topics," *Conference papers, published as Numerical Methods in Geotechnical Engineering*, McGraw-Hill, New York, pp. 1-64.

Douglas, J., Jr., D.W. Peaceman, and H.H. Rachford, Jr., 1959. "A Method for Calculating Multidimensional Immiscible Displacement," *Transc. Soc. Petrol. Eng. Amer. Inst. Mining, Metallurgical and Petroleum Eng.*, Vol. 216, pp. 297-308.

Finlayson, B.A. 1972. *The Method of Weighted Residuals and Variational Principles.* New York: Academic Press.

Fread, D.L. 1971. "Discussion of Implicit Flood Routing in Natural Channels," M. Amein and C.S. Fang, *J. Hydraul. Div. Amer. Soc. Civil Eng.*, Vol. 97, No. HY7, pp. 1156-1159.

Fread, D.L. 1973. "Effects of Time Step Size in Implicit Dynamic Routing," *Water Resour. Bulletin, AWRA*, Vol. 9, No. 2, pp. 339-351.

Fread, D.L. 1974. "Numerical Properties of Implicit Four-point Finite Difference Equations of Unsteady Flow," *NOAA Tech. Memo. Nws., HYDRO-18*, National Weather Service, National Oceanic and Atmospheric Administration, U.S. Dept. of Commerce, Silver Spring, Md., 38 pp.

Fread, D.L. 1976. "Theoretical Development of Implicit Dynamic Routing Model," presented at Dynamic Routing Seminar, Lower Mississippi River Forecase Center, Slidell, Louisiana, NWS., NOAA, Silver Springs, Maryland.

Fread, D.L. and G.F. Smith. 1978. "Calibration Technique for 1-D Unsteady Flow Models," J. Hydraul. Div. Amer. Soc. Civil Eng., Vol. 104, No. HY7, pp. 1027-1044.

Fread, D.L. 1978. "The NWS Dam-Break Flood Forecasting Model, Dam-Break Modeling Seminar," held at NWS Kansas City, Mo.

Freeze, R.A. 1972. "Role of Subsurface Flow in Generating Surface Runoff, 1. Base flow Contributions to Channel Flow," Water Resour. Res., Vol. 8, No. 3, pp. 609-623.

Fujii, H. 1972. "Finite Element Schemes Stability and Convergence," Proc. 2d US-Japan Semin. Matrix Methods Struct. Mech., Univ. of Alabama Press, Huntsville, Ala., August.

Galerkin, B.G. 1915. "Vestn. Inzh. Tech.," (USSR) 19, 897, (translation 63-18924, clearinghouse, Fed. Sci. Tech. Info.,) Springfield, Va.

Gallagher, R.H., J.T. Oden, C. Taylor, and O.C. Zienkiewicz, ed. 1974 "Finite Elements in Fluids," International Conference On Finite Element Method in Flow Analysis, University College of Wales, Vol. 1 and 2. New York: John Wiley and Sons, 1975.

Gray, W.G. and G.F. Pinder. 1974. "Galerkin Approximation of the Time Derivation in the Finite Element Analysis of Groundwater Flow," Water Resour. Res., Vol 10, No. 4, pp. 821-828.

Gray, W.G., G.F. Pinder and C.A. Brebbia, ed. 1976. "Finite Element in Water Resources," Proceedings of the First Intern. Conf. on Finite Elements in Water Resour. held at Princeton University, USA in July. Published by Pentech Press Limited, Estover Road, Plymouth, Devon.

Grotkop, G. 1973. "Finite Element Analysis of Long-period Water Waves," Compt. Methods Appl. Mech. Eng., Vol. 2, pp. 147-157.

Gunaratnam, D.J. and F.E. Perkins. 1970. "Numerical Solutions of Un-Steady Flows in Open Channels," Hydrodynamics Lab. Rep. 127, Department of Civil Eng., Mass. Inst. of Tech., Cambridge, Mass., 260 pp.

Henderson, F.M. 1966. Open Channel Flow. 1st Ed. New York: Mac-Millian Publishing Co., 521 pp.

Isaacson, E., J.J. Stoker and A. Troesch. 1954, 1956. "Numerical Solution of Flood Prediction and River Regulation Problems," Rep. 1 mm-234, Inst. for Math. and Mech., New York Univ., N.Y., 47 pp., 70 pp.

Javandel, I. and P.A. Witherspoon. 1968. "Application of the Finite Element Method to Transient Flow in Porous Media," Soc. Petrol. Eng. J., Vol. 8, No. 3, pp. 241-252.

Keuning, D.H. 1976. "Application of Finite Element Method to Open Channel Flow," J. Hydraulic Div. Amer. Soc. Civil Eng., Vol. 102, No. HY4, pp. 459-467.

Kreig, R.D. and S.W. Key. 1973. "Comparison of Finite Element and Finite-Element and Finite-Difference Methods," Proceeding, ONR Symp. Numer. Mtd. and Comput. Mtd. in Struct. Mech., New York: Academic Press, Inc.

Lagrange, I.L. 1783. Memoir on the Theory of Movement of Fluids, Berlin, Germany, Acad. Royal.

Laplace, P.S. 1975-76. "Researches on Some Points of World System," Paris, France, Acad. Sci.

Lax, P.D. and R.D. Richtmyer. 1956. "Survey of the Stability of Finite Difference Equations," Comm. Pure and Appl. Math., Vol. 9 Vol. 9, pp. 267-923.

Liggett, J.A. and D.A. Wollhiser. 1967. "Difference Solution of Shallow-Water Equations," J. Eng. Mech. Div. Amer. Soc. Civil Eng, Vol. 95, No. EM2, pp. 39-71.

Manam, P.R. and A. Prakash. 1977. "Discussion, Finite Element Soltion of Saint Venant Equation," J. Hydraul. Div. Amer. Soc. Civil Eng., Vol. 103, No. HY5, pp. 565-577.

Martin, H.C. and G.F. Carey. 1973. Introduction to Finite Element Analysis. 1st Ed. New York: McGraw-Hill, 386 pp.

McCarthy, G.T. 1938. "The Unit Hydrograph and Flood Routing," unpublished manuscript presented at Conference of the North Atlantic Div., U.S. Army Corps of Eng., June 24.

Myers, G.L. 1971. Analytical Methods in Conduction Heat Transfer. 1st Ed. New York: McGraw-Hill, 508 pp.

Norrie, D.H. and G DeVries. 1973. The Finite Element Method. 1st Ed. Academic Press, 322 pp.

Norrie, D.H. and G. DeVries. 1975. "A Finite Element Bibliography, (I. Author Listing; II Keyword Listing; III Citation Listing)," Mech. Eng. Dept. Rept. 57, 58, 59, Univ. of Calgary, Canada.

Oden, J.T. 1969. "A General Theory of Finite Elements, 2, Applications," Int. J. Numer. Methods Eng., 1, pp. 247-259.

Oden, J.T. 1972. Finite Element of Nonlinear Continua. 1st Ed. New York: McGraw-Hill, 432 pp.

Oden, J.T. and R.B. Frost. 1973. "Convergence, Accuracy and Stability of Finite Element Approximations of a Class of Nonlinear Hyperbolic Equation," Intern. J. Numer. Methods Eng., Vol. 6, No. 3, pp. 357-365.

Oklahoma Water Resources Board. 1980. Oklahoma Comprehensive Water Plan, Pub. 94, Oklahoma City, 248 pp.

Pinder, G.F. and S.P. Sauer. 1971. "Numerical Simulation of Flood Wave Modification Due to Bank Storage Effect," Water Resour. Res., Vol. 7, No. 1, pp. 63-70.

Puls, L.G. 1928. "Flood Regulation of the Tennessee River," House Document No. 185, 70th Congress, 1st Session.

Rayleigh, J.W.S. 1877. Theory of Sound. 1st Ed. revised. New York: Dover Publications, Inc., 1945.

Rippl, W. 1883. "The Capacity of Storage. Reservoir for Water Supply," Proc. Inst. of Civil Engineers, Vol. 71, pp. 370-278.

Ritz, W. 1909. "Uber Eine Neue Methods Zur Losung Gewisser Variations-Prombleme der Mathematischen Physik," J. Reine Angew. Math., Vol. 135, No. 1, P. 1.

Ross, B.B. and V.O. Shanholtz. 1979. "A One-Dimensional Finite Element for Modeling the Hydrology of Small Upland Watershed," Proc. of the Hydro. Transp. Modeling Symposium, Amer. Soc. Agric. Eng., Pub. 4-80, pp. 42-59.

Segerlind, L.J. 1976. Applied Finite Element Analysis. 1st Ed., New York: John Wiley & Sons, 422 pp.

Strelkoff, T. 1970. "Numerical Solution of the Saint-Venant Equations," J. Hydraul. Div., Amer. Soc. Civil Eng., Vol. 96, No. HY1, pp. 223-252.

Thomas, W.A. 1975. "Water Surface Profiles," Hydrologic Eng. Methods for Water Resour. Development, HEC, Corps of Engineers, U.S. Army, Davis, Cal., Vol. 6, July.

Turner, M.J., R.W. Clough, H.C. Martine, and L.P. Topp. 1956. "Stiffness and Deflection Analysis of Complex Structures," J. Aeron. Sci., Vol. 23, No. 9, pp. 805-823, 852.

Tyagi, A.K. 1971. "Hydrodynamics of Transition Zone Between Fresh and Salt Waters in Coastal Aquifers," Ph.D. Dissertation, University of California, Berkeley, 168 pp.

Tyagi, A.K. 1975a. "Finite Element Modeling of Unsaturated Flow in Porous Media," 16th Congress, International Assoc. Hydr. Res., Sao Paulo, Brazil, Vol. 5, pp. 181-190.

Tyagi, A.K. 1975b. "Finite Element Modeling of Sea Water Intrusion in Coastal Aquifers," Second World Congress, International Water Resour. Assoc., New Delhi, India, Vol. 3, pp. 325-338.

Tyagi, A.K. 1975c. "Finite Element Simulation of Groundwater Flow in Aquifer Systems," International Assoc. Hydr. Res., Rapperswill, Switzerland, Vol. 2, pp. 1-6.

Van Genuchten, M.T. 1977. "On the Accuracy and Efficiency of Several Numerical Schemes for Solving the Convective-Dispersion Equation," Finite Element in Water Resour., Proceeding of the 1st Intern. Conf. on Finite Element Methods in Water Resour. held at Princeton Univ., USA, July, Published by Pentech Press Ltd., Devon.

Varga, R.S. 1962. Matrix Iterative Analysis: New York, Prentice-Hall, 322 pp.

Viessman, W., Jr., J.W. Knapp, G.L. Lewis and T.E. Harbaugh 1972. Introduction to Hydrology. 2nd Ed. New York: Harper and Row Publishers, 704 pp.

Von Rosenberg, D.U. 1969. "Methods for the Numerical Solution of Partial Differential Equations," Nomograph, Amer. Elsevier Publishing Company, New York, 128 pp.

Water Waves. 1965. New York, Interscience Publishers.

Weinmann, P.E. and E.C. Laurenson. 1979. "Approximate Flood Routing Methods: A Review," J. Hydraul. Div. Amer. Soc. Civil Eng., Vol. 105, No. HY12, pp. 1521-1536.

Weigant, P. 1982. Personal Interview. National Weather Service River Forecast Center, Tulsa, Oklahoma, February 19.

Wisler, C.O. and E.F. Brater. 1931. "A Direct Method of Flood Routing," Transc. Amer. Soc. Civil Eng., Vol. 197, pp. 1519-1529.

Wylie, E.B. and B.L. Streeter. 1978. Fluid Transient. New York: McGraw-Hill, 384 pp.

Zienkiewicz, O.C. and Y.K. Cheung. 1965. "Finite Elements in the Solution of Field Problems," The Engineer, Vol. 50, pp. 507-510.

Zienkiewicz, O.C., P. Mayer, and Y.K. Cheung. 1966. "Solution of Anisotropic Seepage Problems by Finite Elements," Proc. Amer. Soc. Civil Eng., Vol. 92, No. EM1, pp. 111-120.

Zienkiewicz, O.C. 1977. The Finite Element Methods in Engineering Science. 3rd ed. New York: McGraw-Hill, 521 pp.

Zienkiewicz, O.C. and R.W. Lewis. 1973. "Analysis of Various Time-Stepping schemes for Initial Value Problems," Earthquake Eng. Struct. Dyn., Vol. 1, pp. 407-408.

Zienkiewicz, O.C. 1974. "Why Finite Elements?", Intern. Conf. on the Finite Element Method in Flow Analysis, Univ. College of Wales, Vol. 1, New York: John Wiley and Sons.

APPENDIX A

NEWTON-RAPHSON FUNCTIONAL ITERATIVE METHOD FOR SOLUTION OF NON-LINEAR SYSTEM(S)

A. Single Variable Equation

Consider a single non-linear variable equation expressed in functional form as follows:

$$f\ (x) = 0 \tag{A.1}$$

where

x = a real variable;

$f(x)$ = any reasonably well-behaved function.

The solution of the variable x of Equation (A.1) is obtained in an iterative manner, proceeding from the first solution estimate, x^n, towards the succeeding improved estimate, x^{n+1}, which tends to converge toward the solution variable x. The orderly procedure by which the improved solution estimate x^{n+1} is sought, such that it converges to the true solution x, is known as Newton-Raphson Iteration and is described as follows.

Let the non-linear equation f(x) be expanded using its Taylor series for an initial iterate x^o.

i.e.

$$f(x) = f(x^o) + \frac{(x-x^o)}{1!} f'(x^o) + \frac{(x-x^o)^2}{2!} f''(x^o) + \frac{(x-x^o)^3}{3!} f'''\ (x^o) \tag{A.2}$$

The linear function of x^o that best approximates the non-linear function f(x), evaluated at x^o, is obtained by retaining only the first order terms of Equation (A.2) such as:

$$f(x) = f(x^o) + \Delta x f'(x^o) \tag{A.3}$$

where:

$\Delta x = x-x^o$ (correction value);

$f'(x^o) = \frac{\partial f(x^o)}{\partial x^o}$ (Jacobian term evaluated at x^o).

An iteration procedure is desired which will cause the function $f(x^o)$ to approach zero as Δx approaches zero. Thus, theleft-hand side of Equation (A.3) is made equal to zero with the following resulting generalized iteration algorithm:

$$f'(x^n)(x^{n+1}-x^n) = -f(x^n) \tag{A.4}$$

where:

n and n+1 are previous and current iterates respectively.

The Jacobian $f'(x^n)$ needs to be updated at every iteration cycle. However, the initial Jacobian can be kept and used for all cycles or updated at selected iteration cycles at the expense of slow convergence. The iteration process is stopped when convergence is achieved. This can be checked in two ways--the absolute and relative tests. The former requires the absolute difference between the current and previous iterates to be less or equal to a specified value called error criterion.

$$|x^{n+1} - x^n| \leq \varepsilon_1 \tag{A.5}$$

where:

ε_1 = error criterion

The relative test is expressed as:

$$\frac{|x^{n+1} - x^n|}{\text{MAX } (|x^n|, |x^{n+1}|)} \leq \varepsilon_2 \tag{A.6}$$

The relative error test is usually preferred to the absolute test because while the latter requires the knowledge of the size of x^n, the former takes that already into account.

B. Multi-Variable Equation

For a system of non-linear multi-variable equations, the Newton-Raphson method is equally efficient in providing the roots or solution of such a system (Amein and Fang, 1969; Fread, 1976). Consider the following N-dimensional system of non-linear algebraic equations:

$$\begin{aligned} &f_1(x_1, x_2, x_3, ---, x_N) = 0 \\ &f_2(x_1, x_2, x_3, ---, x_N) = 0 \\ &\vdots \\ &f_N(x_1, x_2, x_3, ---, x_N) = 0 \end{aligned} \qquad (A.7)$$

or in a vector notation:

$$f_i(x) = 0 \qquad (A.8)$$

where:

subscript i denotes a particular equation.

In a manner analogous to the steps discussed for a single variable equation (EQ. A.1 through A.6), the linearized form of equation A.8 is as follows (see EQ. A.4):

$$f_i{}'(x^n)(x^{n+1} - x^n) = -f_i(x^n) \qquad (A.9)$$

Express EQ. (A.9) in a more concise form as:

$$f_i{}'(x^n)\Delta x = -f_i(x^n) \qquad (A.10)$$

where:

$$\Delta x = \begin{pmatrix} x_1{}^{n+1} - x_1{}^n \\ x_2{}^{n+1} - x_2{}^n \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ x_N{}^{n+1} - x_N{}^n \end{pmatrix}, \quad f_i(x^n) \begin{pmatrix} f_1(x^n) \\ f_2(x^n) \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ f_N(x^n) \end{pmatrix}$$

$$f_i{}'(x^n) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \frac{\partial f_N}{\partial x_3} & \cdots & \frac{\partial f_N}{\partial x_N} \end{bmatrix}$$

The solution of the linear system of equation represented in vector form by Equation (A.10) is sought for the unknown linear correction vector Δx by a suitable matrix solution technique. For a system of (N x N) matrix equation, Gaussian elimination may be employed. However, the most efficient triangular decomposition solution technique for a compact bi-tridiagonal matrix (Douglas et al., 1959) is presented in Appendix B.

The convergence of the iteration process, Equation (A.6), depends on a good initial solution vector estimate $x°$. If the initial iterates are sufficiently close to x, convergence is attained at a quadratic rate since the iterative procedure is second order, that is, involves the first derivative.

APPENDIX B

COMPACT BI-TRIDIAGONAL SOLUTION ALGORITHM

Consider the following system of linear algebraic equations generated by weighted implicit diffusion- or complete-flow models of the finite element approximations of Saint-Venant equations and the Newton-Raphson iterative method as a bi-tridiagonal system:

$$\begin{aligned} a_i^{(1)} y_{i-1} + a_i^{(2)} v_{i-1} + b_i^{(1)} y_i + b_i^{(2)} v_i + c_i^{(1)} y_{i+1} + c_i^{(2)} v_{i+1} &= d_i^{(1)} \\ a_i^{(3)} y_{i-1} + a_i^{(4)} v_{i-1} + b_i^{(3)} y_i + b_i^{(4)} v_i + c_i^{(3)} y_{i+1} + c_i^{(4)} v_{i+1} &= d_i^{(2)} \end{aligned} \qquad \text{(B.1)}$$

for $1 \leq i \leq N$

with $a_1^{(m)} = c_N^{(m)} = 0$ for $1 \leq m \leq 4$

Equation (B.1) is an equivalent form of Equation (A.10), Appendix A, and can be conveniently expressed in a compact (2Nx6) matrix form as follows:

$$\begin{bmatrix} 0 & 0 & b_1^{(1)} & b_1^{(2)} & c_1^{(1)} & c_1^{(2)} \\ 0 & 0 & b_1^{(3)} & b_1^{(4)} & c_1^{(3)} & c_1^{(4)} \\ \text{---} & \text{--} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{--} & \text{---} & \text{---} & \text{---} & \text{---} \\ a_i^{(1)} & a_i^{(2)} & b_i^{(1)} & b_i^{(2)} & c_i^{(1)} & c_i^{(2)} \\ a_i^{(3)} & a_i^{(4)} & b_i^{(3)} & b_i^{(4)} & c_i^{(3)} & c_i^{(4)} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ a_N^{(1)} & a_N^{(2)} & b_N^{(1)} & b_N^{(2)} & 0 & 0 \\ a_N^{(3)} & a_N^{(4)} & b_N^{(3)} & b_N^{(4)} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta v_1 \\ \Delta y_2 \\ \Delta v_2 \\ \text{---} \\ \text{---} \\ \Delta y_{i-1} \\ \Delta v_{i-1} \\ \Delta y_i \\ \Delta v_i \\ \Delta y_{i+1} \\ \Delta v_{i+1} \\ \text{---} \\ \text{---} \\ \Delta y_{N-1} \\ \Delta v_{N-1} \\ \Delta y_N \\ \Delta v_N \end{bmatrix} = \begin{bmatrix} d_1^{(1)} \\ d_1^{(2)} \\ \text{---} \\ \text{---} \\ d_i^{(1)} \\ d_1^{(2)} \\ \text{---} \\ \text{---} \\ d_{N-1}^{(1)} \\ d_{N-1}^{(2)} \end{bmatrix} \qquad \text{(B.2)}$$

The compact solution algorithm developed by Douglas et al (1959) and later used by Von Rosenberg (1969) is a direct solution technique for a system of linear equations. The algorithm is an efficient triangular decomposition method that yields a recursion equation, thus substantially reducing computations and computer core storage.

The algorithm is as follows:

First Computer

$$\begin{aligned}
\beta_i^{(1)} &= b_i^{(1)} - a_i^{(1)} \lambda_{i-1}^{(1)} - a_i^{(2)} \lambda_{i-1}^{(3)} \\
\beta_i^{(2)} &= b_i^{(2)} - a_i^{(1)} \lambda_{i-1}^{(2)} - a_i^{(2)} \lambda_{i-1}^{(4)} \\
\beta_i^{(3)} &= b_i^{(3)} - a_i^{(3)} \lambda_{i-1}^{(1)} - a_i^{(4)} \lambda_{i-1}^{(3)} \\
\beta_i^{(4)} &= b_i^{(4)} - a_i^{(3)} \lambda_{i-1}^{(2)} - a_i^{(4)} \lambda_{i-1}^{(4)}
\end{aligned} \tag{B.3}$$

with $\beta_1^{(m)} = b_1^{(m)}$ for $1 \leq m \leq 4$

and

$$\delta_i^{(1)} = d_i^{(1)} - a_i^{(1)} \gamma_{i-1}^{(1)} - a_i^{(2)} \gamma_{i-1}^{(2)}$$

$$\delta_i^{(2)} = d_i^{(2)} - a_i^{(3)} \gamma_{i-1}^{(1)} - a_i^{(4)} \gamma_{i-1}^{(2)}$$

with $\delta_i^{(1)}= d_1^{(1)}$ and $\delta_1^{(2)} = d_1^{(2)}$

and $\mu_i = \beta_i^{(1)} \beta_i^{(4)} - \beta_i^{(2)} \beta_i^{(3)}$

The $\beta_i^{(m)}$, $\delta_i^{(m)}$, and μ_i are computed to aid in the computation of the following functions and need not be stored after the computation of

$$\begin{aligned}
\lambda_i^{(1)} &= (\beta^{(4)} c_i^{(1)} - \beta_i^{(2)} c_i^{(3)})/\mu_i \\
\lambda^{(2)} &= (\beta^{(4)} c_i^{(2)} - \beta_i^{(2)} c_i^{(4)})/\mu_i
\end{aligned} \tag{B.4}$$

$$\begin{aligned}
\lambda_i^{(3)} &= (\beta_i^{(1)} c_i^{(3)} - \beta_i^{(3)} c_i^{(1)})/\mu_i \\
\lambda^{(4)} &= (\beta_i^{(1)} c_i^{(4)} - \beta_i^{(3)} c_i^{(2)})/\mu_i
\end{aligned} \tag{B.4}$$

and

$$\gamma_i^{(1)} = (\beta_i^{(4)}\delta_i^{(1)} - \beta_i^{(2)}\delta_i^{(2)})/\mu_i$$

$$\gamma_i^{(2)} = (\beta_i^{(1)}\delta_i^{(2)} - \beta_i^{(3)}\delta_i^{(1)})/\mu_i$$

The values of $\lambda_i^{(m)}$ and $\gamma_i^{(m)}$ must be stored as they are used in the back solution. This is

$$y_N = \gamma_N^{(1)}$$
$$v_N = \gamma_N^{(2)} \qquad (B.5)$$

and

$$y_i = \gamma_i^{(1)} - \lambda_i^{(1)} y_{i+1} - \lambda_i^{(2)} v_{i+1}$$

$$v_i = \gamma_i^{(2)} - \lambda_i^{(3)} y_{i+1} - \lambda_i^{(4)} v_{i+1}$$

$$\text{for } (N-1) \geq i \geq 1$$

APPENDIX C

DESCRIPTION OF SUBROUTINES

The list of subroutines and their corresponding functions is given below. The list of major variables and symbols used in the computer program is provided in the comment page of computer program listing, Appendix E. Any temporary storage variables are not included because their definitions are obvious.

Subroutines

MAIN Coordinates the functions of the other subprograms, and prints converged solutions for prescribed time increments.

READW Reads and echo-checks all the input data.

JACOBI Evaluates and updates the Jacobian matrix of (2Nx6) terms.

VECTR Evaluates and updates the column vector of size (2Nx1).

GEOMTR Evaluates and updates the nodal flow area, wetted perimeter, variation of Manning's roughness coefficient with discharge, and the change of hydraulic radius with respect to depth of flow.

BTRIDG Solves the compact (2Nx6) bi-tridiagonal matrix equations.

APPENDIX D

GUIDE FOR DATA INPUT

The formats for entering the data are given below. The same format statements for READ and WRITE are applicable to both diffusion and complete flow models as provided in the subprogram READW. The data deck for the particular example of the Illinois River flood of April 10, 1979, is presented. However, data for the Flint Creek as lateral inflow for a single reach is entered via the MAIN program as a DATA STATEMENT. The reader should refer to Appendix E for the definition of the variables.

CARD	COLUMNS	FORMAT	VARIABLE
1	1 - 10	F 10·4	TPRINT
	11 - 20	F 10·4	TTA
	21 - 30	F 10·4	TSUM
	31 - 40	F 10·4	T
2	1 - 10	F 10·4	TETHA
	11 - 20	F 10·4	DETA
	21 - 30	F 10·4	DETV
	31 - 40	F 10·4	SO
	41 - 50	I10	IMAX
	51 - 60	I10	N1
3	1 - 72	6F12·5	YO(J)
4	1 - 72	6F12·5	QRE(J)
5	1 - 72	6F12·5	QLAT(J)
6	1 - 72	6F12·5	XL(J)
7	1 - 5	I5	JORD
8	1 - 50	5F10·5	ASF(J)

CARD	COLUMN	FORMAT	VARIABLE
9	1 - 50	5F10·5	PSF(J)

Note: QSTR(J) and TRS(J) are included in the main program in DATA STATEMENTS.

APPENDIX E

COMPUTER PROGRAM LISTING FOR WICFEM

```
$JOB            ,TIME=5
C    ***************************************************************
C    *                                                             *
C    *            1 - DIMENSIONAL STREAMFLOW ROUTING MODEL         *
C    *                                                             *
C    *                                                             *
C    ***************************************************************
C    *                                                             *
C    *       COMPLETE FLOW FINITE ELEMENT STREAMFLOW ROUTING       *
C    *                                                             *
C    *       SOLVED IMPLICITLY BY ITERATIVE NEWTON-RAPHSON MTD.    *
C    *                                                             *
C    ***************************************************************
C    *                                                             *
C    *              ... DEFINITION OF TERMS ...                    *
C    *                                                             *
C    *VARIABLES UNITS ARE AS FOLLOWS: TIME(SEC),LENGTH(FT)         *
C     *DETPH(FT),VELOCITY(FT. PER SEC),DISCHARGE(CFS)              *
C    *ACC IS THE ACCELERATION OF GRAVITY,32.2FT. PER SEC PER SEC   *
C    *ACF,PCF ARE POLYNO. COEFF. FOR AREA & WETTED PERIMETER       *
C    *AN,PN ARE THE AREA & WETTED PERIMETER OF FLOW RESPECTIVELY   *
C    *HYD IS THE RATE OF CHANGE OF HYDR.RADIUS WITH DEPTH          *
C    *QSTR IS THE UPSTREAM INFLOW DISCHARGE HYDRCGRAPH             *
C    *QLAT IS THE LATERAL FLOW TERM,FT. PER SEC                    *
C    *QFL IS THE LATERAL INFLOW HYDROGRAPH AT FLINT CREEK          *
C    *SO IS THE CONSTANT CHANNEL SLOPE                             *
C    *RN IS THE MANNING ROUGHNESS COEFF.                           *
C    *XL IS THE NODAL SPACING.                                     *
C    *YO IS THE INITIAL UNIFORM NORMAL DEPTH.                      *
C    *VO IS THE INITIAL UNIFORM NORMAL VELOCITY                    *
C    *N1 IS THE TOTAL NUMBER OF NODES                              *
C    *T IS THE TIME STEP (SECONDS)                                 *
C    *TSUM IS THE ENTIRE FLOOD DURATION IN SECONDS.                *
C    *TSR IS THE TIME FOR UPSTREAM INFLOW HYDROGRAPH               *
C    *TFL IS THE TIME FOR LATERAL INFLOW HYDROGRAPH AT FLINT       *
C    *TPRINT IS THE TIME FOR INITIAL PRINTING (SECONDS)            *
C    *TTA IS THE INCREMENTAL PRINTING TIME (SECONDS)               *
C    *---IF TIME STEP IS GREATER THAN TTA PRINTING WILL BE ---     *
C    *       PERFORMED AT THE INCREMENT OF THE TIME STEP,T---      *
C    *IMAX IS THE MAX. ITERATION LIMIT                             *
C    *NT IS THE NUMBER OF POINTS FOR UPSTREAM INFLOW HYDROGRAPH    *
C    *NTP IS THE UPSTREAM INFLOW HYDROG POINT FOR THE PEAK FLOW    *
C    *JG.JGP ARE SAME AS NT,NTP FOR LATERA INFLOW HYDROG FOR FLINT*
C    *JORD IS THE CRDER OF POLYNOMIAL EQ. FOR AREA & W. PERIMETER *
C    *DETA IS THR CONVERGENCE CRITERIA FOR DEPTH                   *
C    *TETHA IS THE TIME WEIGHTING FACTOR.                          *
C    *YN IS THE CALCULATED DETH OF FLOW                            *
C    *VN IS THE CCRRESPONDING VELOCITY OF FLOW                     *
C    *BSR IS THE JACOBIAN MATRIX OF DIMENSION ( 2N1 X 6 )
C    *CXV1 & CXV2 ARE THE (N1X1) COLUMN VECTORS EVALUATED AT       *
C    *(1-TETHA ) & TETHA RESPECTIVELY.                             *
C    *LDIM,LDIN ARE THE VARIABLE DIMENSIONING PARAMETERS           *
C    *READW IS THE SUBPROGRAM TO READ & ECHOE CHECK INPUT DATA     *
C    *GEOMTR IS THE SUBPROGRAM TO UPDATE FLOW AREA,WETTED,         *
C    *  MANNING'S ROUGHNESS COEFF. & RATE OF CHANGE HYDR. RACIUS   *
C    *                                                             *
C    ***************************************************************
C
           DIMENSION ACF(5),AN(26),BSR(52,6),CSV1(52),CSV2(52),FCF(5),
         1 DTPH(26),HYD(26),QLAT(25),QRE(26),PN(26),VELY(26),
     2 VO(26),VN(26),XL(25),YO(26),YN(26)
       DIMENSION QFL(7),QSTR(11),TFL(7),TSR(11),RN(26)
       DATA QFL,JG,JGP/81.0,408.,629.,931.,566.,318.,164.,4,7/
       DATA TFL/0.0,43200.,50400.,57600.,86400.,136800.,345600./
       DATA ACC,LDIM,LDIN,NR,LP,LK/32.2,52,26,5,2*6/
       DATA NT,NTP/11,6/
C
C      ---READ & ECHOE-CHECK INPUT DATA FROM SUBPROGRAM.
       LDN = LDIN - 1
       CALL READW(ACF,DETA,DETV,QLAT,QRE,PCF,QSTR,TSR,NT,
     1 TPRINT,TTA,TSUM,T,TETHA,SO,IMAX,XL,YO,N1,NR,LP,LDN,LDIN)
C
       JSIZE = TSUM/T
       MS = JG + 1
       MC = NTP + 1
C
       TWRIT = TPRINT
       JSTP = N1 - 2
C
C      ---CALCULATE INITIAL GEOMETRIC PARAMETERS FROM SUBPROGRAM.
       CALL GEOMTR(ACF,AN,HYD,QRE,PCF,PN,YO,RN,N1,LDIN)
       PR = 2./3.
       DO 100 J = 1,N1
   100 VO(J) = QRE(J)/AN(J)
       QO = QRE(1)
       WRITE(LP,140)
   140 FORMAT(////28X,'UPSTREAM',24X,'MIDSTREAM',24X,'DOWNSTREAM')
       WRITE(LP,145)
   145 FORMAT(4X,'TIME(HR.)',7X,'DISCHARGE',4X,'DEPTH',3X,'VELOCITY',6X,
     1 'DISCHARGE',4X,'DEPTH',3X,'VELOCITY',6X,'DISCHARGE',4X,'DEPTH',
     2 3X,'VELOCITY')
C
C      ---USE INITIAL VELOCITY & DEPTH OF FLOW AS GUESS VALUES
C      TO INITIATE SIMULATION.
       DO 150 K = 1,N1
       YN(K) = YO(K)
   150 VN(K) = VO(K)
       FAC1 = TETHA*T
       FAC2 = (1. - TETHA)*T
       DO 155 L = 1,N1
   155 RN(L) = ACC*RN(L)**2/2.2082
C      --- SET LOOP FOR TIME SIMULATION.
       QR = QO
       DO 900 JL =1,JSIZE
       TIME = FLOAT(JL)*T
C
C      ---UPDATE THE LATERAL INFLOW HYDROGRAPH FOR REACH,7
       TCK = TIME - TFL(JG)
       IF(TCK)160,160,200
   160 DO 180 KC = 2,JG
       IF(TIME - TFL(KC))190,190,180
   180 CONTINUE
   190 QFW = QFL(KC-1) + (QFL(KC) - QFL(KC-1))/
     1 (TFL(KC) - TFL(KC-1))*(TIME - TFL(KC-1))
       GO TO 320
   200 IF(TCK - TFL(JGP))220,220,300
   220 DO 230 KS = MS,JGP
       IF(TIME - TFL(KS))240,240,230
   230 CONTINUE
   240 QFW = QFL(KS-1) - (QFL(KS-1) - QFL(KS))/
```

```
     1 (TFL(KS) - TFL(KS-1))*(TIME - TFL(KS-1))
       GO TO 320
  300  QFW = QFL(JGP)
  320  QLAT(7) = QFW/(XL(7)*PN(7))
C
C      --- UPDATE UPSTREAM BOUNDARY CONDITION.
       TD = TIME - TSR(NTP)
       IF(TD)350,350,380
  350  DO 360 LC = 2,NTP
       IF(TIME - TSR(LC))370,370,360
  360  CONTINUE
  370  QR = QSTR(LC-1) + (QSTR(LC) - QSTR(LC-1))/
     1 (TSR(LC) - TSR(LC-1))*(TIME - TSR(LC-1))
       GO TO 520
  380  IF(TD - TSR(NT))400,400,500
  400  DO 420 JC = MC,NT
       IF(TIME - TSR(JC))450,450,420
  420  CONTINUE
  450  QR = QSTR(JC-1) - (QSTR(JC-1) - QSTR(JC))/
     1 (TSR(JC) - TSR(JC-1))*(TIME - TSR(JC-1))
       GO TO 520
  500  QR = QSTR(NT)
  520  CONTINUE
C      ---CALL SUBROUTINE TO GENERATE COLUMN VECTOR (2N X 1)
       JSWTCH = 1
       CALL VECTR(CSV1,FAC2,QLAT,YO,VO,SO,ACC,XL,QR,RN,N1,
     1 AN,PN,QRE,LDN,LDIN,LDIM,JSWTCH)
C
C      ---GENERATE JACOBIAN MATRIX.
       LUP = 0
       CALL JACOBI(BSR,FAC1,YN,VN,XL,QLAT,QR,RN,
     1 AN,PN,QRE,HYD,SO,ACC,LDIM,LDN,LDIN,N1,LK)
C      --- ITERATE TO CONVERGENCE FOR EACH TIME STEP.
       DO 590 LL =1,IMAX
       LAST = 2*N1
       JSWTCH = 2
       LUP = LUP + 1
       CALL VECTR(CSV2,FAC1,QLAT,YN,VN,SO,ACC,XL,QR,RN,N1,
     1 AN,PN,QRE,LDN,LDIN,LDIM,JSWTCH)
       DO 530 K = 1,LAST
       CSV2(K) = CSV2(K) - CSV1(K)
  530  CONTINUE
C      ---OBTAIN SOLUTION VIA TRI-DIAGONAL SUBPROGRAM.
       CALL BTRIDG(CSV2,BSR,DTPH,VELY,LDIM,LDIN,N1,LK)
C      ---UPDATE THE NODAL GEOMETRIC PARAMETERS.
       CALL GEOMTR(ACF,AN,HYD,QRE,PCF,PN,YN,RN,N1,LDIN)
       DO 550 L = 1,N1
  550  RN(L) = ACC*RN(L)**2/2.2082
       JS = N1 - 1
       VEC = QR/AN(1)
       VELY(1) = VN(1) - VEC
C
C
C      --- CHECK FOR RELATIVE CONVERGENCE FOR ALL VARIABLES.
       JERR = 0
       DO 560 J =1,N1
       YB1 = ABS(DTPH(J))
       VB1 = ABS(VELY(J))
       YB2 = YN(J) - DTPH(J)
       VB2 = VN(J) - VELY(J)
```

```
      YB3 = MAX1(ABS(YB2),ABS(YN(J)))
      VB3 = MAX1(ABS(VB2),ABS(VN(J)))
      IF(YB3 .LE. 0.0 .OR. VB3 .LE. 0.0)GO TO 570
      YERROR = YB1/YB3
      VERROR = VB1/VB3
      IF(YERROR .LE. DETA .AND. VERROR .LE. DETV)JERR=JERR+1
  560 CONTINUE
C
C     --- SWITCH CURRENT VALUES OF DEPTH OF FLOW TO OLD ONES.
  570 DO 580 L=1,N1
      YN(L) = YN(L) - DTPH(L)
      VN(L) = VN(L) - VELY(L)
  580 CONTINUE
      IF(JERR .EQ. N1) GO TO 600
C     --- CHECK IF SPECIFIED ITERATION LIMIT IS EXCEEDED.
C
      IF(JERR .LT. N1 .AND. LL .GE. IMAX) GO TO 920
C
C     ---UPDATE THE JACOBIAN MATRIX AT EVERY 3 ITERATIOS.
      IF(LUP - 3)590,585,585
  585 CALL JACOBI(BSR,FAC1,YN,VN,XL,QLAT,QR,RN,
     1 AN,PN,QRE,HYD,SO,ACC,LDIM,LDN,LDIN,N1,LK)
      LUP = 0
  590 CONTINUE
C     --- UPDATE DEPTHS & VELOCITIES OF PREVIOUS TIME STEP.
  600 DO 680 J=1,N1
      VO(J) = VN(J)
      YO(J) = YN(J)
  680 CONTINUE
C     --- PRINT OUT RESULTS.
      IF(TIME - TPRINT + .003)750,700,700
  700 TPRINT = TPRINT + TTA
      TM = TIME/TWRIT
      DO 710 J = 1,N1
      QRE(J) = AN(J)*VN(J)
  710 CONTINUE
      WRITE(LP,720)TM,QRE(1),YN(1),VN(1),QRE(13),YN(13),VN(13),
     1  QRE(N1),YN(N1),VN(N1)
  720 FORMAT(2X,F10.2,5X,3F10.3,5X,3F10.3,5X,3F10.3)
  750 IF(TIME - TSUM)900,950,950
C     --- ADVANCE THE TIME STEP.
  900 CONTINUE
  920 WRITE(LP,930)
  930 FORMAT(///10X,'MAX. ITERATION LIMIT EXCEEDED.')
  950 STOP
      END
C   ***************************************************************
C   *                                                             *
C   *  SUBPROGRAM TO READ AND ECHOE INPUT DATA                    *
C   *                                                             *

      SUBROUTINE READW(ASF,DETA,DETV,QLAT,QRE,PSF,QSTR,TSR,
     1  NT,TPRINT,TTA,TSUM,T,TETHA,SO,IMAX,XL,YO,N1,NR,LP,LDN,LDIN)
C   *                                                             *
C   ***************************************************************
      DIMENSION ASF(5),PSF(5),QLAT(LDN),QRE(LDIN),XL(LDN),YO(LDIN),
     1 QSTR(NT),TSR(NT)
C
C     ---READ TIME PARAMETERS.
      READ(NR,50)TPRINT,TTA,TSUM..
```

```
   50 FORMAT(4F10.2)
      READ(NR,80)TETHA,DETA,DETV,SO,IMAX,N1
   80 FORMAT(4F10.4,2I10)
C     ---READ UPSTREAM BOUNDARY DISCHARGE HYDROGRAPH.
      READ(NR,90)(QSTR(L),L = 1,NT)
      READ(NR,90)(TSR(L),L = 1,NT)
C     ---READ INITIAL DEPTHS OF FLOW,DISCHARGE,LATERAL FLOW& SPACING.
      LDN = N1 - 1
      READ(NR,90)(YO(J),J = 1,N1)
      READ(NR,90)(QRE(J),J = 1,N1)
      READ(NR,90)(QLAT(J),J = 1,LDN)
      READ(NR,90)(XL(J),J = 1,LDN)
   90 FORMAT(6F12.5)
C     ---READ ORDER OF POLYNOMIAL EQ.
      READ(NR,110)JORD
  110 FORMAT(I5)
      LR = JORD + 1
C     ---READ AREA & WETTED PERIMETER POLYNO. COEFF. MATRICE
C      ONE ROW AT A TIME.
      READ(NR,140)(ASF(J),J = 1,LR)
      READ(NR,140)(PSF(J),J = 1,LR)
  140 FORMAT(5F10.5)
C
C     ---PRINT OUT INPUT DATA.
      WRITE(LP,150)
  150 FORMAT(1H1)
      WRITE(LP,160)N1,T,SO
  160 FORMAT(///20X,'TOTAL NO. OF NODES =',I5//20X,
     1'TIME STEP =',F10.3,1X,'SEC.'//20X,'CHANNEL BOTTOM SLOPE =',
     2F10.4)
      WRITE(LP,170)TETHA,IMAX,DETA,DETV
  170 FORMAT(//20X,'TIME WEIGHTING FACTOR =',F10.4//20X,
     1'MAX. ITERATION LIMIT =',I5//20X,
     2'CONVERGENCE CRITERIA FOR DEPTH =',F10.4//20X,
     3'CONVERGENCE CRITERIA FOR VELOCITY =',F10.4)
C
      WRITE(LP,180)(J,TSR(J),QSTR(J),J = 1,NT)
  180 FORMAT(//////23X,'UPSTREAM DISCHARGE HYDROGRAPH'//20X,
     1'J',5X,'TIME PERIOD',5X,'MEASURED FLOW'//(20X,I2,4X,
     2F10.1,3X,F10.1))
C
      WRITE(LP,190)(K,YO(K),QRE(K),K=1,N1)
  190 FORMAT(/////20X,'NODE',5X,'INITIAL DEPTH',5X,'INITIAL DISCHARGE'
     1//(18X,I3,8X,F10.3,12X,F10.3))
      WRITE(LP,200)(J,XL(J),QLAT(J),J =1,LDN)
  200 FORMAT(/////20X,'REACH',9X,'LENGTH',9X,'LATERAL FLOW'//(20X,
     1I3,8X,F10.2,5X,F10.6))
      WRITE(LP,220)
  220 FORMAT(/////20X,'FOURTH-ORDER REGRESSION COEFF. FOR AREA'//23X,
     1'0-TH',7X,'1ST',7X,'2ND',7X,'3RD',7X,'4TH')
      WRITE(LP,300)(ASF(J),J = 1,LR)
      WRITE(LP,240)
  240 FORMAT(/////20X,'FOURTH-ORDER REGRESSION COEFF. FOR WETTED
     1PERIMETER'//23X,'0-TH',7X,'1ST',7X,'2ND',7X,'3RD',7X,'4TH')
      WRITE(LP,300)(PSF(J),J = 1,LR)
  300 FORMAT(/20X,5F10.5)
      RETURN
      END
C   ******************************************************************
C   *                                                                *
```

```
C    *  SUBPROGRAM TO UPDATE THE NODAL FLOW AREA,WETTED PERIMETER ,      *
C    *                                                                   *
C    *  MANNING'S ROUGHNESS COEFF. & THE CHANGE IN HYDRAULIC             *
C    *                                                                   *
C    *  RADIUS WITH RESPECT TO DEPTH                                     *
C    *                                                                   *

        SUBROUTINE GEOMTR(ASF,AX,HYD,QX,PSF,PX,YX,RC,N1,LDIN)
C    *                                                                   *
C    *                                                                   *
C    *********************************************************************
        DIMENSION AX(LDIN),HYD(LDIN),PX(LDIN),YX(LDIN),ASF(5),PSF(5),
     1 QX(LDIN),RC(LDIN)
C
C       ---COMPUTE THE NODAL FLOW AREA,WETTED PERIMETER AND
C       MANNING'S ROUGHNESS COEFF.
        DO 200 J = 1,N1
        AX(J) = ASF(1) + ASF(2)*YX(J) + ASF(3)*YX(J)**2 +
     1 ASF(4)*YX(J)**3 + ASF(5)*YX(J)**4
C
        PX(J) = PSF(1) + PSF(2)*YX(J) + PSF(3)*YX(J)**2 +
     1 PSF(4)*YX(J)**3 + PSF(5)*YX(J)**4
C
        RC(J) = 0.03713 + 0.14097E-05*QX(J) + 0.41739E-10*
     1 QX(J)**2 - 0.230004E-14*QX(J)**3
  200   CONTINUE
C       ---COMPUTE THE NODAL RATE OF CHANGE OF HYDR. RADIUS
        DO 300 K = 1,N1
        STOR1 = ASF(2) + 2.*ASF(3)*YX(K) + 3.*ASF(4)*YX(K)**2 +
     1 4.*ASF(5)*YX(K)**3
C
        STOR2 = PSF(2) + 2.*PSF(3)*YX(K) + 3.*PSF(4)*YX(K)**2 +
     1 4.*PSF(5)*YX(K)**3
C
        HYD(K) = (STOR1*PX(K) - STOR2*AX(K))/PX(K)**2
        HYD(K) = ABS(HYD(K))
  300   CONTINUE
        RETURN
        END
C    ***************************************************************
C    *                                                             *
C    *   SUBPROGRAM TO GENERATE COLUNMN VECTOR (2NX1)              *
C    *                                                             *

        SUBROUTINE VECTR(CXV,FAC,QLAT,YX,VX,SO,G,XL,QT,
     1 RX,N1,AX,PX,QRE,LDN,LDIN,LDIM,JSWTCH)
C    *                                                             *
C    ***************************************************************
C
        DIMENSION CXV(LDIM),QLAT(LDN),YX(LDIN),VX(LDIN),XL(LDN)
        DIMENSION AX(LDIN),QRE(LDIN),PX(LDIN),RX(LDIN)
        K = 1
        PS = 4./3.
        LSTP = N1 - 2
        SPK = QT/AX(1)
        RY1 = (AX(1)/PX(1))**PS
        RY2 = (AX(2)/PX(2))**PS
C       ---UPSTREAM NODAL COLUMN VECTOR.
        A = (2.*XL(1) + FAC*(VX(2) - 4.*SPK))*YX(1)
        B = (XL(1) + FAC*(2.*VX(2) + SPK))*YX(2)
```

```
      C = 3.*FAC*QLAT(1)*XL(1)
      E = (4.*XL(1)-FAC*(2.*VX(1)+VX(2)))*VX(1)
      F = (2.*XL(1)- FAC*(VX(1)+2.*VX(2)))*VX(2)
      GP = 2.*FAC*RX(1)*XL(1)*(2.*VX(1)**2/RY1 +
     1 VX(2)**2/RY2)
      H = 2.*FAC*(2.*XL(1)*QLAT(1)*VX(1)/YX(1) +
     1 XL(1)*QLAT(1)*VX(2)/YX(2))
      P = 6.*G*FAC*(YX(2) - YX(1))
      S = 6.*G*SO*FAC*XL(1)
      IF(JSWTCH - 1) 50,50,80
   50 A = (2.*XL(1) - FAC*(VX(2) - 4.*SPK))*YX(1)
      B = (XL(1) - FAC*(2.*VX(2) + SPK))*YX(2)
      E = (4.*XL(1) + FAC*(2.*VX(1)+VX(2)))*VX(1)
      F = (2.*XL(1) + FAC*(VX(1)+2.*VX(2)))*VX(2)
      CXV(1) = A + B+ C
      CXV(2) = E + F - GP - H - P + S
      GO TO 90
   80 CXV(1) = A + B - C
      CXV(2) = E + F + GP + H + P - S
C     ---INTERIOR NODAL COLUMN VECTORS.
   90 DO 200 J = 1,LSTP
      K = K + 2
      M = K + 1
      RD1 = (AX(J)/PX(J))**PS
      RD2 = (AX(J+1)/PX(J+1))**PS
      RD3 = (AX(J+2)/PX(J+2))**PS
      AI = (XL(J) - FAC*(VX(J+1) + 2.*VX(J)))*YX(J)
      BI = (2.*(XL(J)+XL(J+1)) + FAC*(VX(J+2)-VX(J)))*YX(J+1)
      CI = (XL(J+1) + FAC*(2.*VX(J+2)+VX(J+1)))*YX(J+2)
      DI = 3.*FAC*(QLAT(J)*XL(J) + QLAT(J+1)*XL(J+1))
      EI = (2.*XL(J) + FAC*(2.*VX(J)+VX(J+1)))*VX(J)
      FI = (4.*(XL(J)+XL(J+1)) + FAC*(VX(J)-VX(J+2)))*VX(J+1)
      GI = (2.*XL(J+1) - FAC*(VX(J+1)+2.*VX(J+2)))*VX(J+2)
      HI = 2.*FAC*RX(J+1)*(XL(J)*VX(J)**2/RD1 + 2.*(XL(J)+XL(J+1))*
     1 VX(J+1)**2/RD2 + XL(J+1)*VX(J+2)**2/RD3)
      PI = 2.*FAC*(QLAT(J)*XL(J)*VX(J)/YX(J) + 2.*(XL(J)*QLAT(J)+
     1 XL(J+1)*QLAT(J+1))*VX(J+1/YX(J+1)) +QLAT(J+1)*XL(J+1)*
     2 VX(J+2)/YX(J+2))
      QI = 6.*G*FAC*(YX(J+2) - YX(J))
      SI = 6.*G*SO*FAC*(XL(J) + XL(J+1))
      IF(JSWTCH - 1)100,100,150
  100 AI = (XL(J) + FAC*(VX(J+1) +2.*VX(J)))*YX(J)
      BI = (2.*(XL(J)+XL(J+1)) - FAC*(VX(J+2)-VX(J)))*YX(J+1)
      CI = (XL(J+1) - FAC*(2.*VX(J+2)+VX(J+1)))*YX(J+2)
      EI = (2.*XL(J) - FAC*(2.*VX(J) + VX(J+1)))*VX(J)
      FI = (4.*(XL(J)+XL(J+1)) - FAC*(VX(J)-VX(J+2)))*VX(J+1)
      GI = (2.*XL(J+1) + FAC*(VX(J+1)+2.*VX(J+2)))*VX(J+2)
      CXV(K) = AI + BI + CI + DI
      CXV(M) = EI + FI + GI - HI - PI - QI + SI
      GO TO 200
  150 CXV(K) = AI + BI + CI - DI
      CXV(M) = EI + FI + GI + HI + PI + QI - SI
  200 CONTINUE
C
C     ---DOWNSTREAM NODAL CALCULATION.
      HT1 = AX(N1-1)/PX(N1-1)
      HT2 = AX(N1)/PX(N1)
      RY3 = HT1**PS
      RY4 = HT2**PS
      AN = (XL(N1-1) - FAC*(VX(N1)+ 2.*VX(N1-1)))*YX(N1-1)
```

```
            BN = (2.*XL(N1-1) + FAC*(4.*VX(N1) -VX(N1-1)))*YX(N1)
            CN = 3.*FAC*QLAT(N1-1)*XL(N1-1)
            GN = FAC*RX(N1)*XL(N1-1)*(VX(N1-1)**2/RY3 + 2.*VX(N1)**2/RY4)
            PN = 3.*G*FAC*(YX(N1) - YX(N1-1))
            SN = 3.*G*SO*FAC*XL(N1-1)
            IF(JSWTCH - 1)300,300,400
      300   AN = (XL(N1-1) + FAC*(VX(N1) + 2.*VX(N1-1)))*YX(N1-1)
            BN = (2.*XL(N1-1) - FAC*(4.*VX(N1) - VX(N1-1)))*YX(N1)
            CXV(LDIM-1) = AN + BN + CN
            CXV(LDIM) = SN - GN - PN
            GO TO 500
      400 CXV(LDIM-1) = AN + BN - CN
            CXV(LDIM) = GN + PN - SN
      500   RETURN
            END
C     **********************************************************
C     *                                                        *
C     *   SUBPROGRAM TO GENERATE THE JACOBIAN MATRIX.          *
C     *                                                        *

            SUBROUTINE JACOBI(BTR,FAC,YX,VX,XL,QLAT,QT,RX,
          1 AX,PX,QRE,HYD,SO,G,LDIM,LDN,LDIN,N1,LK)
C     *                                                        *
C     **********************************************************
C
            DIMENSION BTR(LDIM,LK),QLAT(LDN),VX(LDIN),YX(LDIN),XL(LDN)
            DIMENSION AX(LDIN),QRE(LDIN),PX(LDIN),HYD(LDIN),RX(LDIN)
            K = 1
            PW = 7./3.
            PS = 4./3.
            SPK = QT/AX(1)
            SSK = SPK/YX(1)
            RD1 = AX(1)/PX(1)
            RD2 = AX(2)/PX(2)
C           ---EVALUATE THE JACOBIAN TERMS FOR UPSTREAM NODE.
            BTR(1,1) = 0.0
            BTR(1,2) = 0.0
            BTR(2,1) = 0.0
            BTR(2,2) = 0.0
            BTR(1,3) = 2.*XL(1) + FAC*(VX(2) - SSK*YX(2))
            BTR(1,4) = 0.0
            BTR(1,5) = XL(1) + FAC*(2.*VX(2) + SPK)
            BTR(1,6) = FAC*(YX(1) + 2.*YX(2))
            STOR1 = 4.*PS*RX(1)*XL(1)*VX(1)**2*HYD(1)/RD1**PW
            BTR(2,3) = -FAC*(STOR1 + 4.*XL(1)*QLAT(1)*VX(1)/YX(1)**2+6.*G)
            BTR(2,4) = 4.*XL(1) - 2.*FAC*(2.*VX(1) + VX(2) -
          1 4.*XL(1)*VX(1)*RX(1)/RD1**PS - 2.*XL(1)*QLAT(1)/YX(1))
            BTR(2,5) = -2.*FAC*(PS*RX(1)*XL(1)*HYD(2)*VX(2)**2/RD2**PW +
          1 XL(1)*QLAT(1)*VX(2)/YX(2)**2 - 3.*G)
            BTR(2,6) = 2.*XL(1) - 2.*FAC*(VX(1) + 2.*VX(2) - 2.*RX(1)*
          1 XL(1)*VX(2)/RD2**PS - XL(1)*QLAT(1)/YX(2))
C
C           ---EVALUATE JACOBIAN TERMS FOR INTERIOR NODES.
            LSTP = N1 - 2
            DO 300 J =1,LSTP
            K = K + 2
            M = K + 1
            RD1 = AX(J)/PX(J)
            RD2 = AX(J+1)/PX(J+1)
            RD3 = AX(J+2)/PX(J+2)
      BTR(K,1) = XL(J) - FAC*(VX(J+1) + 2.*VX(J))
      BTR(K,2) = -FAC*(2.*YX(J) + YX(J+1))
      BTR(K,3) = 2.*(XL(J)+XL(J+1)) + FAC*(VX(J+2) -VX(J))
      BTR(K,4) = FAC*(YX(J+2) - YX(J))
      BTR(K,5) = XL(J+1) + FAC*(2.*VX(J+2) + VX(J+1))
      BTR(K,6) = FAC*(YX(J+1) + 2.*YX(J+2))
      BTR(M,1) = -2.*FAC*(PS*RX(J+1)*XL(J)*VX(J)**2*HYD(J)/RD1**PW +
     1 XL(J)*QLAT(J)*VX(J)/YX(J)**2 + 3.*G)
      BTR(M,2) = 2.*XL(J) + 2.*FAC*(2.*VX(J) + VX(J+1) +
     1 2.*RX(J+1)*XL(J)*VX(J)/RD1**PS + XL(J)*QLAT(J)/YX(J))
      BTR(M,3) = -4.*FAC*(PS*RX(J+1)*(XL(J)+XL(J+1))*HYD(J+1)*
     1 VX(J+1)**2/RD2**PW + (XL(J)*QLAT(J)+XL(J+1)*QLAT(J+1))*VX(J+1)/
     2 YX(J+1)**2)
      BTR(M,4) = 4.*(XL(J) + XL(J+1)) + 2.*FAC*(VX(J) - VX(J+2) +
     1 4.*RX(J+1)*(XL(J) + XL(J+1))*VX(J+1)/RD2**PS + 2.*(XL(J)*QLAT(J)+
     2 XL(J+1)*QLAT(J+1))/YX(J+1))
      BTR(M,5) =-2.*FAC*(PS*RX(J+1)*XL(J+1)*HYD(J+2)*VX(J+2)**2/
     1 RD3**PW + XL(J+1)*QLAT(J+1)*VX(J+1)/YX(J+2)**2 - 3.*G)
      BTR(M,6) = 2.*XL(J+1) - 2.*FAC*(VX(J+1) + 2.*VX(J+2) -2.*
     1 RX(J+1)*XL(J+1)*VX(J+2)/RD3**PS - XL(J+1)*QLAT(J+1)/YX(J+2))
  300 CONTINUE
C     ---EVALUATE JACOBIAN TERMS FOR DOWNSTREAM NODES.
C
      LB = LDIM - 1
      RDN1 = AX(N1-1)/PX(N1-1)
      RDN2 = AX(N1)/PX(N1)
      BTR(LB,1) = XL(N1-1) - FAC*(VX(N1)+2.*VX(N1-1))
      BTR(LB,2) = -FAC*(2.*YX(N1-1) + YX(N1))
      BTR(LB,3) = 2.*XL(N1-1) + FAC*(4.*VX(N1)-VX(N1-1))
      BTR(LB,4) = FAC*(4.*YX(N1) - YX(N1-1))
      BTR(LDIM,1) = -2.*FAC*(PS*RX(N1)*XL(N1-1)*VX(N1-1)**2*HYD(N1-1)/
     1 RDN1**PW + XL(N1-1)*QLAT(N1-1)*VX(N1-1)/YX(N1-1)**2 + 3.*G)
      BTR(LDIM,2) = 2.*XL(N1-1) + 2.*FAC*(2.*VX(N1-1) + VX(N1) +
     1 2.*RX(N1)*XL(N1-1)*VX(N1-1)/RDN1**PS + XL(N1-1)*QLAT(N1-1)/
     2 YX(N1-1))
      BTR(LDIM,3) = -2.*FAC*(2.*PS*RX(N1)*XL(N1-1)*VX(N1)**2*HYD(N1)/
     1 RDN2**PW + 2.*XL(N1-1)*QLAT(N1-1)*VX(N1)/YX(N1)**2 - 3.*G)
      BTR(LDIM,4) = 4.*XL(N1-1) + 2.*FAC*(VX(N1-1) + 2.*VX(N1) +
     1 4.*RX(N1)*XL(N1-1)*VX(N1)/RDN2**PS + 2.*XL(N1-1)*QLAT(N1-1)/
     2 YX(N1))
      BTR(LB,5) = 0.0
      BTR(LB,6) = 0.0
      BTR(LDIM,5) = 0.0
      BTR(LDIM,6) = 0.0
      RETURN
      END
C     ***********************************************************
C     *                                                         *
C     *   SUBPROGRAM FOR SOLUTION OF A BI-TRIDIAGONAL MATRIX.   *
C     *                                                         *

      SUBROUTINE BTRIDG(COL,VEL,YX,VX,LDIM,LDIN,N1,LK)
C     *                                                         *
C     *                                                         *
C     ***********************************************************
      DIMENSION VEL(LDIM,LK),YX(LDIN),VX(LDIN),COL(LDIM),
     1 BETA(4),DETA(2),SAC(26,4),GAMA(26,2)
C
C     ---PERFORM MATRIX REDUCTION OPERATION
      K = 1
```

```
          DO 20) J = 1,N1
          IF(J -: 1)100,100,150
      100 BETA(1 ) = VEL(1,3)
          BETA(2 ) = VEL(1,4)
          BETA(3 ) = VEL(2,3)
          BETA(4 ) = VEL(2,4)
          DETA(1 ) = COL(1)
          DETA(2 ) = COL(2)
          ZU = BETA(1)*BETA(4) - BETA(2)*BETA(3)
          IF(ZU .EQ. 0.0)ZU = 0.001
          SAC(1, 1 ) = (BETA(4)*VEL(1,5) - BETA(2)*VEL(2,5))/ZU
          SAC(1, 2 ) = (BETA(4)*VEL(1,6) - BETA(2)*VEL(2,6))/ZU
          SAC(1, 3 ) = (BETA(1)*VEL(2,5) - BETA(3)*VEL(1,5))/ZU
          SAC(1, 4 ) = (BETA(1)*VEL(2,6) - BETA(3)*VEL(1,6))/ZU
          GO TO 180
      150 K = K + 2
          M = K + 1
          BETA(1 ) = VEL(K,3) - VEL(K,1)*SAC(J-1,1)-VEL(K,2)*SAC(J-1,3)
          BETA(2 ) = VEL(K,4)-VEL(K,1)*SAC(J-1,2)-VEL(K,2)*SAC(J-1,4)
          BETA(3 ) = VEL(M,3)-VEL(M,1)*SAC(J-1,1)-VEL(M,2)*SAC(J-1,3)
          BETA(4 ) = VEL(M,4)-VEL(M,1)*SAC(J-1,2)-VEL(M,2)*SAC(J-1,4)
C
          DETA(1 ) = COL(K)-VEL(K,1)*GAMA(J-1,1)-VEL(K,2)*GAMA(J-1,2)
          DETA(2 ) = COL(M)-VEL(M,1)*GAMA(J-1,1)-VEL(M,2)*GAMA(J-1,2)
C
          ZU = BETA(1)*BETA(4) - BETA(2)*BETA(3)
          IF(ZU .EQ. 0.0)ZU = 0.001
          SAC(J, 1 ) = (BETA(4)*VEL(K,5)-BETA(2)*VEL(M,5))/ZU
          SAC(J, 2 ) = (BETA(4)*VEL(K,6)-BETA(2)*VEL(M,6))/ZU
          SAC(J, 3 ) = (BETA(1)*VEL(M,5) -BETA(3)*VEL(K,5))/ZU
          SAC(J, 4 ) = (BETA(1)*VEL(M,6) -BETA(3)*VEL(K,6))/ZU
C
      180 GAMA(J,1) = (BETA(4 )*DETA(1)-BETA(2)*DETA(2))/ZU
          GAMA(J,2) = (BETA(1 )*DETA(2)-BETA(3)*DETA(1))/ZU
      200 CONTINUE
C         ---COMPUTE SOLUTION VIA RECURSIVE EQ.
          LIMIT = N1 - 1
          JK = LIMIT
          YX(N1)  = GAMA(LDIN,1)
          VX(N1)  = GAMA(LDIN,2)
          DO 30) L = 1,LIMIT
          YX(JK)  = GAMA(JK,1)-SAC(JK,1)*YX(JK+1)-SAC(JK,2)*VX(JK+1)
          VX(JK)  = GAMA(JK,2)-SAC(JK,3)*YX(JK+1)-SAC(JK,4)*VX(JK+1)
          JK = JK - 1
      300 CONTINUE
          RETURN
          END

$ENTRY
```

APPENDIX F

SAMPLE OUTPUT FOR WICFEM

The sample output listed in the pages following is the format with which the input parameters are reprinted for correction and referral. A clear illustration is drawn from a natural channel simulation of Illinois River, using flood of April 10, 1979, and the complete flow model.

```
TOTAL NO. OF NODES =   26

TIME STEP =  1800.000 SEC.

CHANNEL BOTTOM SLOPE =    0.0009


TIME WEIGHTING FACTOR =    0.5500

MAX. ITERATION LIMIT =   60

CONVERGENCE CRITERIA FOR DEPTH =    0.0100

CONVERGENCE CRITERIA FOR VELOCITY =    0.1000




   UPSTREAM DISCHARGE HYDROGRAPH

J       TIME PERIOD     MEASURED FLOW

 1             0.0         482.0
 2         50400.0         757.0
 3         64800.0        5590.0
 4         79200.0        7710.0
 5         86400.0       11000.0
 6        100800.0       22980.0
 7        129600.0       11320.0
 8        158400.0        5100.0
 9        172800.0        4110.0
10        208800.0        3104.0
11        345600.0        1722.0



NODE        INITIAL DEPTH      INITIAL DISCHARGE

1                3.340              482.000
2                3.340              482.000
3                3.340              482.000
4                3.340              482.000
5                3.340              482.000
6                3.340              482.000
7                3.340              482.000
8                3.340              482.000
9                3.340              482.000
10               3.340              482.000
11               3.340              482.000
12               3.340              482.000
13               3.340              482.000
14               3.340              482.000
```

```
15          3.340          482.000
16          3.340          482.000
17          3.340          482.000
18          3.340          482.000
19          3.340          482.000
20          3.340          482.000
21          3.340          482.000
22          3.340          482.000
23          3.340          482.000
24          3.340          482.000
25          3.340          482.000
26          3.340          482.000


 REACH        LENGTH         LATERAL FLOW

   1        10560.00 2     0.000000
   2        10560.00 4     0.000000
   3        10560.00 6     0.000000
   4        10560.00 8     0.000000
   5        10560.00 10 -  0.000000
   6        10560.00 12    0.000000
   7        10560.00       0.000050
   8        10560.00       0.000000
   9        10560.00       0.000000
  10        10560.00       0.000000
  11        10560.00       0.000000
  12        10560.00       0.000000
  13        10560.00       0.000000
  14        10560.00       0.000000
  15        10560.00       0.000000
  16        10560.00       0.000000
  17        10560.00       0.000000
  18        10560.00       0.000000
  19        10560.00       0.000000
  20        10560.00       0.000000
  21        10560.00       0.000000
  22        10560.00       0.000000
  23        10560.00       0.000000
  24        10560.00       0.000000
  25        12672.00       0.000000


 FOURTH-ORDER REGRESSION COEFF. FOR AREA

    0-TH        1ST        2ND        3RD        4TH

  -32.01320  84.60530   5.47340    0.91215   -0.00452

 FOURTH-ORDER REGRESSION COEFF. FOR WETTED          PERIMETER

    0-TH        1ST        2ND        3RD        4TH

  10.08235   57.67010  -4.90130    0.34127   -0.00601
```

	UPSTREAM			MIDSTREAM			DOWNSTREAM		
TIME(HR.)	DISCHARGE	DEPTH	VELOCITY	DISCHARGE	DEPTH	VELOCITY	DISCHARGE	DEPTH	VELOCITY
1.00	501.643	2.854	1.807	565.813	3.340	1.640	546.570	3.317	1.618
2.00	521.285	2.895	1.845	606.491	3.340	1.758	618.323	3.296	1.850
3.00	540.928	3.060	1.767	628.423	3.340	1.821	656.780	3.327	1.921
4.00	560.571	3.148	1.774	644.660	3.349	1.860	669.159	3.343	1.937
5.00	580.214	3.132	1.845	665.587	3.384	1.891	664.443	3.346	1.920
6.00	599.857	3.199	1.859	699.077	3.460	1.923	661.311	3.344	1.913
7.00	619.500	3.292	1.841	740.850	3.560	1.955	659.136	3.343	1.908
8.00	639.142	3.331	1.863	778.815	3.652	1.980	657.959	3.342	1.906
9.00	659.785	3.377	1.883	806.762	3.719	1.996	657.274	3.341	1.904
10.00	679.428	3.449	1.882	826.363	3.765	2.007	656.856	3.341	1.903
11.00	698.071	3.501	1.892	837.804	3.793	2.013	656.619	3.340	1.903
12.00	717.714	3.539	1.914	840.218	3.799	2.013	656.492	3.340	1.902
13.00	737.357	3.589	1.927	838.264	3.794	2.012	656.418	3.340	1.902
14.00	757.000	3.652	1.928	842.535	3.801	2.015	656.431	3.340	1.902
15.00	1965.249	5.978	2.306	864.539	3.847	2.032	656.830	3.341	1.903
16.00	3173.499	7.102	2.760	908.835	3.939	2.061	658.747	3.346	1.904
17.00	4381.742	8.519	2.691	983.721	4.092	2.106	664.227	3.359	1.909
18.00	5589.992	9.264	2.907	1095.981	4.311	2.166	676.106	3.387	1.920
19.00	6119.992	9.512	3.037	1235.558	4.570	2.230	695.399	3.436	1.934
20.00	6649.992	9.790	3.167	1365.832	4.807	2.277	722.138	3.502	1.952
21.00	7179.992	10.503	2.913	1466.053	4.983	2.307	752.192	3.575	1.972
22.00	7709.992	10.741	3.008	1591.610	5.199	2.329	779.893	3.644	1.986
23.00	9354.992	11.755	2.948	2041.995	5.766	2.529	802.875	3.703	1.998
24.00	10999.990	12.304	3.264	2905.542	6.715	2.766	821.331	3.748	2.008
25.00	13994.990	13.543	3.307	3825.629	7.613	2.906	834.800	3.781	2.014
26.00	16989.990	14.924	3.267	4662.191	8.349	2.980	850.800	3.818	2.023
27.00	19984.990	16.389	3.006	5413.211	8.958	3.019	873.799	3.876	2.037
28.00	22979.990	17.102	3.235	6137.531	9.494	3.056	925.853	3.974	2.072
29.00	21522.490	16.609	3.165	6763.383	10.000	3.035	1002.750	4.117	2.116
30.00	20064.990	16.373	3.041	7510.277	10.505	3.048	1125.035	4.336	2.189
31.00	18607.490	16.201	2.917	8483.727	11.031	3.113	1323.278	4.697	2.288
32.00	17150.000	15.598	2.981	9468.043	11.582	3.137	1680.693	5.246	2.430
33.00	15692.500	14.766	3.063	10539.300	12.153	3.152	2272.846	5.982	2.642
34.00	14235.000	14.530	2.861	11656.490	12.732	3.154	2980.063	6.748	2.809
35.00	12777.500	13.486	2.998	12632.130	13.303	3.107	3649.134	7.374	2.902
36.00	11320.000	12.740	3.059	13835.280	13.835	3.119	4363.063	8.068	2.973
37.00	10542.490	12.447	3.033	14774.170	14.301	3.091	4932.383	8.595	3.023
38.00	9764.996	12.134	2.887	15524.680	14.671	3.069	5637.734	9.049	3.084
39.00	8987.496	11.436	3.032	15938.890	14.932	3.032	6221.613	9.452	3.104
40.00	8209.996	11.010	3.070	16219.390	15.060	3.019	6783.055	9.960	3.106
41.00	7432.496	10.601	2.935	16113.430	15.077	2.995	7471.414	10.340	3.135
42.00	6655.000	9.830	3.080	15850.500	15.002	2.979	8095.895	10.741	3.139
43.00	5877.500	9.427	3.017	15419.780	14.850	2.963	8579.770	11.098	3.143
44.00	5100.000	8.925	2.848	14851.550	14.635	2.952	9299.203	11.428	3.171
45.00	4852.496	8.695	2.840	14168.860	14.360	2.939	9919.813	11.798	3.162
46.00	4604.996	8.461	2.829	13430.570	14.035	2.933	10523.770	12.111	3.172
47.00	4357.496	8.045	2.935	12682.270	13.658	2.944	11093.890	12.383	3.180
48.00	4109.996	8.015	2.830	11765.590	13.226	2.932	11604.490	12.668	3.169
49.00	4009.399	7.827	2.881	10807.480	12.737	2.923	12074.970	12.882	3.179
50.00	3908.799	7.836	2.805	9824.559	12.194	2.919	12464.820	13.103	3.165

51.00	3808.199	7.673	2.848	8829.344	11.611	2.910	12821.030	13.280	3.163
52.00	3707.600	7.614	2.801	7882.184	11.006	2.906	13075.450	13.444	3.144
53.00	3607.000	7.509	2.807	7011.211	10.412	2.896	13275.830	13.567	3.130
54.00	3506.400	7.399	2.793	6257.238	9.857	2.887	13356.730	13.661	3.104
55.00	3405.800	7.324	2.771	5638.133	9.373	2.875	13365.330	13.710	3.083
56.00	3305.200	7.195	2.772	5154.316	8.976	2.862	13246.800	13.722	3.053
57.00	3204.599	7.123	2.740	4790.555	8.666	2.848	13044.680	13.684	3.026
58.00	3104.000	6.995	2.740	4520.684	8.431	2.833	12714.070	13.604	2.991
59.00	3067.631	7.019	2.698	4317.703	8.253	2.819	12294.370	13.472	2.957
60.00	3031.262	6.940	2.720	4158.090	8.112	2.805	11749.540	13.293	2.914
61.00	2994.894	6.933	2.693	4036.201	7.995	2.799	11076.960	13.000	2.875
62.00	2958.526	6.897	2.680	3919.003	7.887	2.787	10216.430	12.585	2.841
63.00	2922.157	6.872	2.664	3806.999	7.784	2.775	9293.941	12.085	2.818
64.00	2885.788	6.835	2.654	3699.744	7.683	2.762	8346.797	11.521	2.799
65.00	2849.420	6.802	2.643	3599.432	7.588	2.750	7437.984	10.922	2.787
66.00	2813.052	6.764	2.635	3509.756	7.501	2.739	6611.473	10.326	2.778
67.00	2776.683	6.726	2.626	3431.770	7.425	2.729	5900.816	9.765	2.774
68.00	2740.315	6.686	2.619	3365.245	7.359	2.720	5319.344	9.271	2.772
69.00	2703.947	6.617	2.644	3302.410	7.301	2.707	4865.309	8.883	2.764
70.00	2667.579	6.581	2.626	3251.927	7.252	2.699	4543.902	8.578	2.764
71.00	2631.210	6.537	2.626	3207.462	7.207	2.692	4283.523	8.329	2.756
72.00	2594.841	6.490	2.619	3166.590	7.166	2.686	4083.387	8.130	2.750
73.00	2558.473	6.451	2.612	3127.312	7.126	2.680	3915.717	7.963	2.741
74.00	2522.105	6.402	2.607	3088.362	7.086	2.673	3778.913	7.823	2.734
75.00	2485.737	6.368	2.587	3058.964	7.044	2.661	3713.560	7.697	2.721
76.00	2449.368	6.286	2.585	3023.510	7.002	2.659	3616.291	7.608	2.720
77.00	2413.000	6.251	2.602	2985.725	6.958	2.655	3517.497	7.519	2.708
78.00	2376.631	6.264	2.554	2946.345	6.914	2.649	3434.561	7.437	2.700
79.00	2340.262	6.131	2.607	2889.579	6.870	2.643	3343.439	7.358	2.699
80.00	2303.894	6.186	2.549	2876.619	6.826	2.646	3284.313	7.304	2.683
81.00	2267.526	6.063	2.565	2810.322	6.781	2.631	3233.575	7.237	2.691
82.00	2231.157	6.037	2.547	2797.600	6.736	2.634	3176.162	7.191	2.673
83.00	2194.789	6.013	2.520	2746.666	6.691	2.617	3157.211	7.144	2.675
84.00	2158.420	5.924	2.552	2688.710	6.644	2.609	3092.535	7.097	2.668
85.00	2122.052	5.940	2.498	2674.288	6.599	2.611	3060.258	7.066	2.657
86.00	2085.684	5.826	2.534	2611.124	6.556	2.594	3020.498	7.014	2.660
87.00	2049.315	5.837	2.484	2599.255	6.515	2.596	2971.886	6.975	2.643
88.00	2012.947	5.742	2.507	2539.855	6.473	2.580	2939.010	6.926	2.647
89.00	1976.578	5.732	2.460	2526.038	6.430	2.581	2889.070	6.885	2.629
90.00	1940.210	5.640	2.495	2463.275	6.383	2.564	2856.548	6.838	2.632
91.00	1903.842	5.653	2.431	2448.794	6.336	2.565	2808.084	6.796	2.614
92.00	1867.473	5.560	2.440	2389.665	6.289	2.536	2787.404	6.745	2.612
93.00	1831.105	5.488	2.435	2357.700	6.243	2.534	2744.664	6.709	2.607
94.00	1794.736	5.456	2.428	2320.382	6.197	2.526	2711.566	6.673	2.599
95.00	1758.368	5.418	2.395	2283.573	6.151	2.518	2674.237	6.629	2.592
96.00	1722.000	5.324	2.438	2231.257	6.103	2.509	2622.735	6.583	2.586

APPENDIX G

COMPUTER PROGRAM LISTING FOR WIDFEM

```
$JOB            ,TIME=5
C    **************************************************************
C    *                                                            *
C    *          1 - DIMENSIONAL STREAMFLOW ROUTING MODEL          *
C    *                                                            *
C    *                                                            *
C    **************************************************************
C    *                                                            *
C             IMPLICIT DIFFUSION FINITE ELEMENT METHOD SOLVED     *
C    *                                                            *
C    *           BY ITERATIVE NEWTON-RAPHSON TECHNIQUE            *
C    *                                                            *
C    **************************************************************
C    *                                                            *
C    *              ... DEFINITION OF TERMS ...                   *
C    *                                                            *
C    *VARIABLES UNITS ARE AS FOLLOWS: TIME(SEC),LENGTH(FT)        *
C     *DETPH(FT),VELOCITY(FT. PER SEC),DISCHARGE(CFS)             *
C    *ACC IS THE ACCELERATION OF GRAVITY,32.2FT. PER SEC PER SEC  *
C    *ACF,PCF ARE POLYNO. COEFF. FOR AREA & WETTED PERIMETER      *
C    *AN,PN ARE THE AREA & WETTED PERIMETER OF FLOW RESPECTIVELY  *
C    *HYD IS THE RATE OF CHANGE OF HYDR.RADIUS WITH DEPTH         *
C    *QSTR IS THE UPSTREAM INFLOW DISCHARGE HYDROGRAPH            *
C    *QLAT IS THE LATERAL FLOW TERM,FT. PER SEC                   *
C    *QFL IS THE LATERAL INFLOW HYDROGRAPH AT FLINT CREEK         *
C    *SO IS THE CONSTANT CHANNEL SLOPE                            *
C    *RN IS THE MANNING ROUGHNESS COEFF.                          *
C    *XL IS THE NODAL SPACING.                                    *
C    *YO IS THE INITIAL UNIFORM NORMAL DEPTH.                     *
C    *VO IS THE INITIAL UNIFORM NORMAL VELOCITY                   *
C    *N1 IS THE TOTAL NUMBER OF NODES                             *
C    *T IS THE TIME STEP (SECONDS)                                *
C    *TSUM IS THE ENTIRE FLOOD DURATION IN SECONDS.               *
C    *TSR IS THE TIME FOR UPSTREAM INFLOW HYDROGRAPH              *
C    *TFL IS THE TIME FOR LATERAL INFLOW HYDROGRAPH AT FLINT      *
C    *TPRINT IS THE TIME FOR INITIAL PRINTING (SECONDS)           *
C    *TTA IS THE INCREMENTAL PRINTING TIME (SECONDS)              *
C    *---IF TIME STEP IS GREATER THAN TTA PRINTING WILL BE ---    *
C    *       PERFORMED AT THE INCREMENT OF THE TIME STEP,T---     *
C    *IMAX IS THE MAX. ITERATION LIMIT                            *
C    *NT IS THE NUMBER OF POINTS FOR UPSTREAM INFLOW HYDROGRAPH   *
C    *NTP IS THE UPSTREAM INFLOW HYDROG POINT FOR THE PEAK FLOW   *
C    *JG,JGP ARE SAME AS NT,NTP FOR LATERA INFLOW HYDROG FOR FLINT*
C    *JORD IS THE ORDER OF POLYNOMIAL EQ. FOR AREA & W. PERIMETER *
C    *DETA IS THR CONVERGENCE CRITERIA FOR DEPTH                  *
C    *TETHA IS THE TIME WEIGHTING FACTOR.                         *
C    *YN IS THE CALCULATED DETH OF FLOW                           *
C    *VN IS THE CORRESPONDING VELOCITY OF FLOW                    *
C    *BSR IS THE JACOBIAN MATRIX OF DIMENSION ( 2N1 X 6 )
C    *CXV1 & CXV2 ARE THE (N1X1) COLUMN VECTORS EVALUATED AT      *
C    *(1-TETHA) & TETHA RESPECTIVELY.                             *
C    *LDIM,LDIN ARE THE VARIABLE DIMENSIONING PARAMETERS          *
C    *READW IS THE SUBPROGRAM TO READ & ECHOE CHECK INPUT DATA    *
C    *GEOMTR IS THE SUBPROGRAM TO UPDATE FLOW AREA,WETTED,        *
C    *  MANNING'S ROUGHNESS COEFF. & RATE OF CHANGE HYDR. RADIUS  *
C    *                                                            *
C    **************************************************************
C
        DIMENSION ACF(5),AN(26),BSR(52,6),CSV1(52),CSV2(52),PCF(5),
      1 DTPH(26),HYD(26),QLAT(25),QRE(26),PN(26),VELY(26),
     2 VO(26),VN(26),XL(25),YO(26),YN(26)
       DIMENSION QFL(7),QSTR(11),TFL(7),TSR(11),RN(26)
       DATA QFL,JG,JGP/81.0,408.,629.,931.,566.,318.,164.,4,7/
       DATA TFL/0.0,43200.,50400.,57600.,86400.,136800.,345600./
       DATA ACC,LDIM,LDIN,NR,LP,LK/32.2,52,26,5,2*6/
       DATA NT,NTP/11,6/
C
C      ---READ & ECHOE-CHECK INPUT DATA FROM SUBPROGRAM.
       LDN = LDIN - 1
       CALL READW(ACF,DETA,DETV,QLAT,QRE,PCF,QSTR,TSR,NT,
     1 TPRINT,TTA,TSUM,T,TETHA,SO,IMAX,XL,YO,N1,NR,LP,LDN,LDIN)
C
       JSIZE = TSUM/T
       MS = JG + 1
       MC = NTP + 1
C
       TWRIT = TPRINT
       JSTP = N1 - 2
C
C      ---CALCULATE INITIAL GEOMETRIC PARAMETERS FROM SUBPROGRAM.
       CALL GEOMTR(ACF,AN,HYD,QRE,PCF,PN,YO,RN,N1,LDIN)
       PR = 2./3.
       DO 100 J = 1,N1
   100 VO(J) = QRE(J)/AN(J)
       QO = QRE(1)
       WRITE(LP,140)
   140 FORMAT(////28X,'UPSTREAM',24X,'MIDSTREAM',24X,'DOWNSTREAM')
       WRITE(LP,145)
   145 FORMAT(4X,'TIME(HR.)',7X,'DISCHARGE',4X,'DEPTH',3X,'VELOCITY',6X,
     1 'DISCHARGE',4X,'DEPTH',3X,'VELOCITY',6X,'DISCHARGE',4X,'DEPTH',
     2 3X,'VELOCITY')
C
C      ---USE INITIAL VELOCITY & DEPTH OF FLOW AS GUESS VALUES
C      TO INITIATE SIMULATION.
       DO 150 K = 1,N1
       YN(K) = YO(K)
   150 VN(K) = VO(K)
       FAC1 = TETHA*T
       FAC2 = (1. - TETHA)*T
       DO 155 L = 1,N1
   155 RN(L) = RN(L)**2/2.2082
C      --- SET LOOP FOR TIME SIMULATION.
       QR = QO
       DO 900 JL =1,JSIZE
       TIME = FLOAT(JL)*T
C
C      ---UPDATE THE LATERAL INFLOW HYDROGRAPH FOR REACH,7
       TCK = TIME - TFL(JG)
       IF(TCK)160,160,200
   160 DO 180 KC = 2,JG
       IF(TIME - TFL(KC))190,190,180
   180 CONTINUE
   190 QFW = QFL(KC-1) + (QFL(KC) - QFL(KC-1))/
     1 (TFL(KC) - TFL(KC-1))*(TIME - TFL(KC-1))
       GO TO 320
   200 IF(TCK - TFL(JGP))220,220,300
   220 DO 230 KS = MS,JGP
       IF(TIME - TFL(KS))240,240,230
   230 CONTINUE
   240 QFW = QFL(KS-1) - (QFL(KS-1) - QFL(KS))/
```

```
     1 (TFL(KS) - TFL(KS-1))*(TIME - TFL(KS-1))
       GO TO 320
  300  QFW = QFL(JGP)
  320  QLAT(7) = QFW/(XL(7)*PN(7))
C
C      --- UPDATE UPSTREAM BOUNDARY CONDITION.
       TD = TIME - TSR(NTP)
       IF(TD)350,350,380
  350  DO 360 LC = 2,NTP
       IF(TIME - TSR(LC))370,370,360
  360  CONTINUE
  370  QR = QSTR(LC-1) + (QSTR(LC) - QSTR(LC-1))/
     1 (TSR(LC) - TSR(LC-1))*(TIME - TSR(LC-1))
       GO TO 520
  380  IF(TD - TSR(NT))400,400,500
  400  DO 420 JC = MC,NT
       IF(TIME - TSR(JC))450,450,420
  420  CONTINUE
  450  QR = QSTR(JC-1) - (QSTR(JC-1) - QSTR(JC))/
     1 (TSR(JC) - TSR(JC-1))*(TIME - TSR(JC-1))
       GO TO 520
  500  QR = QSTR(NT)
  520  CONTINUE
C      ---CALL SUBROUTINE TO GENERATE COLUMN VECTOR (2N X 1)
       JSWTCH = 1
       CALL VECTR(CSV1,FAC2,QLAT,YO,VO,SO,ACC,XL,QR,RN,N1,
     1 AN,PN,QRE,LDN,LDIN,LDIM,JSWTCH)
C
C      ---GENERATE JACOBIAN MATRIX.
       LUP = 0
       CALL JACOBI(BSR,FAC1,YN,VN,XL,QLAT,QR,RN,
     1 AN,PN,QRE,HYD,SO,ACC,LDIM,LDN,LDIN,N1,LK)
C      --- ITERATE TO CONVERGENCE FOR EACH TIME STEP.
       DO 590 LL =1,IMAX
       LAST = 2*N1
       JSWTCH = 2
       LUP = LUP + 1
       CALL VECTR(CSV2,FAC1,QLAT,YN,VN,SO,ACC,XL,QR,RN,N1,
     1 AN,PN,QRE,LDN,LDIN,LDIM,JSWTCH)
       DO 530 K = 1,LAST
       CSV2(K) = CSV2(K) - CSV1(K)
  530  CONTINUE
C      ---OBTAIN SOLUTION VIA TRI-DIAGONAL SUBPROGRAM.
       CALL BTRIDG(CSV2,BSR,DTPH,VELY,LDIM,LDIN,N1,LK)
C      ---UPDATE THE NODAL GEOMETRIC PARAMETERS.
       CALL GEOMTR(ACF,AN,HYD,QRE,PCF,PN,YN,RN,N1,LDIN)
       DO 550 L = 1,N1
  550  RN(L) = RN(L)**2/2.2082
       JS = N1 - 1
       VEC = QR/AN(1)
       VELY(1) = VN(1) - VEC
C
C
C      --- CHECK FOR RELATIVE CONVERGENCE FOR ALL VARIABLES.
       JERR = 0
       DO 560 J =1,N1
       YB1 = ABS(DTPH(J))
       VB1 = ABS(VELY(J))
       YB2 = YN(J) - DTPH(J)
       VB2 = VN(J) - VELY(J)
```

```
      YB3 = MAX1(ABS(YB2),ABS(YN(J)))
      VB3 = MAX1(ABS(VB2),ABS(VN(J)))
      IF(YB3  .LE. 0.0 .OR. VB3 .LE. 0.0)GO TO 570
      YERROR  = YB1/YB3
      VERROR  = VB1/VB3
      IF(YERROR .LE. DETA .AND. VERROR .LE. DETV)JERR=JERR+1
  560 CONTINUE
C
C     --- SWITCH CURRENT VALUES OF DEPTH OF FLOW TO OLD ONES.
  570 DO 580  L=1,N1
      YN(L) = YN(L) - DTPH(L)
      VN(L) = VN(L) - VELY(L)
  580 CONTINUE
      IF(JERR .EQ. N1) GO TO 600
C     --- CHECK IF SPECIFIED ITERATION LIMIT IS EXCEEDED.
C
      IF(JERR .LT. N1 .AND. LL .GE. IMAX) GO TO 920
C
C     ---UPDATE THE JACOBIAN MATRIX AT EVERY 3 ITERATIOS.
      IF(LUP - 3)590,585,585
  585 CALL JACOBI(BSR,FAC1,YN,VN,XL,QLAT,QR,RN,
     1 AN,PN,QRE,HYD,SO,ACC,LDIM,LDN,LDIN,N1,LK)
      LUP = 0
  590 CONTINUE
C     --- UPDATE DEPTHS & VELOCITIES OF PREVIOUS TIME STEP.
  600 DO 680  J=1,N1
      VO(J) = VN(J)
      YO(J) = YN(J)
  680 CONTINUE
C     --- PRINT OUT RESULTS.
      IF(TIME - TPRINT + .003)750,700,700
  700 TPRINT = TPRINT + TTA
      TM = TIME/TWRIT
      DO 710  J = 1,N1
      QRE(J)  = AN(J)*VN(J)
  710 CONTINUE
      WRITE(LP,720)TM,QRE(1),YN(1),VN(1),QRE(13),YN(13),VN(13),
     1  QRE(N1),YN(N1),VN(N1)
  720 FORMAT(2X,F10.2,5X,3F10.3,5X,3F10.3,5X,3F10.3)
  750 IF(TIME - TSUM)900,950,950
C     --- ADVANCE THE TIME STEP.
  900 CONTINUE
  920 WRITE(LP,930)
  930 FORMAT(///10X,'MAX. ITERATION LIMIT EXCEEDED.')
  950 STOP
      END
C   ****************************************************************
C   *                                                              *
C   *  SUBPROGRAM TO READ AND ECHOE INPUT DATA                     *
C   *                                                              *

      SUBROUTINE READW(ASF,DETA,DETV,QLAT,QRE,PSF,QSTR,TSR,
     1  NT,TPRINT,TTA,TSUM,T,TETHA,SO,IMAX,XL,YO,N1,NR,LP,LDN,LDIN)
C   *                                                              *
C   ****************************************************************
      DIMENSION ASF(5),PSF(5),QLAT(LDN),QRE(LDIN),XL(LDN),YO(LDIN),
     1 QSTR(NT),TSR(NT)
C
C     ---READ TIME PARAMETERS.
      READ(NR,50)TPRINT,TTA,TSUM,T
```

```
   50 FORMAT(4F10.2)
      READ(NR,80)TETHA,DETA,DETV,SO,IMAX,N1
   80 FORMAT(4F10.4,2I10)
C     ---READ UPSTREAM BOUNDARY DISCHARGE HYDROGRAPH.
      READ(NR,90)(QSTR(L),L = 1,NT)
      READ(NR,90)(TSR(L),L = 1,NT)
C     ---READ INITIAL DEPTHS OF FLOW,DISCHARGE,LATERAL FLOW& SPACING.
      LDN = N1 - 1
      READ(NR,90)(YO(J),J = 1,N1)
      READ(NR,90)(QRE(J),J = 1,N1)
      READ(NR,90)(QLAT(J),J = 1,LDN)
      READ(NR,90)(XL(J),J = 1,LDN)
   90 FORMAT(6F12.5)
C     ---READ ORDER OF POLYNOMIAL EQ.
      READ(NR,110)JORD
  110 FORMAT(I5)
      LR = JORD + 1
C     ---READ AREA & WETTED PERIMETER POLYNO. COEFF. MATRICE
C      ONE ROW AT A TIME.
      READ(NR,140)(ASF(J),J = 1,LR)
      READ(NR,140)(PSF(J),J = 1,LR)
  140 FORMAT(5F10.5)
C
C     ---PRINT OUT INPUT DATA.
      WRITE(LP,150)
  150 FORMAT(1H1)
      WRITE(LP,160)N1,T,SO
  160 FORMAT(///20X,'TOTAL NO. OF NODES =',I5//20X,
     1 'TIME STEP =',F10.3,1X,'SEC.'//20X,'CHANNEL BOTTOM SLOPE =',
     2 F10.4)
      WRITE(LP,170)TETHA,IMAX,DETA,DETV
  170 FORMAT(//20X,'TIME WEIGHTING FACTOR =',F10.4//20X,
     1 'MAX. ITERATION LIMIT =',I5//20X,
     2 'CONVERGENCE CRITERIA FOR DEPTH =',F10.4//20X,
     3 'CONVERGENCE CRITERIA FOR VELOCITY =',F10.4)
C
      WRITE(LP,180)(J,TSR(J),QSTR(J),J = 1,NT)
  180 FORMAT(//////23X,'UPSTREAM DISCHARGE HYDROGRAPH'//20X,
     1 'J',5X,'TIME PERIOD',5X,'MEASURED FLOW'//(20X,I2,4X,
     2 F10.1,3X,F10.1))
C
      WRITE(LP,190)(K,YO(K),QRE(K),K=1,N1)
  190 FORMAT(/////20X,'NODE',5X,'INITIAL DEPTH',5X,'INITIAL DISCHARGE'
     1 //(18X,I3,8X,F10.3,12X,F10.3))
      WRITE(LP,200)(J,XL(J),QLAT(J),J =1,LDN)
  200 FORMAT(/////20X,'REACH',9X,'LENGTH',9X,'LATERAL FLOW'//(20X,
     1 I3,8X,F10.2,5X,F10.6))
      WRITE(LP,220)
  220 FORMAT(/////20X,'FOURTH-ORDER REGRESSION COEFF. FOR AREA'//23X,
     1 '0-TH',7X,'1ST',7X,'2ND',7X,'3RD',7X,'4TH')
      WRITE(LP,300)(ASF(J),J = 1,LR)
      WRITE(LP,240)
  240 FORMAT(/////20X,'FOURTH-ORDER REGRESSION COEFF. FOR WETTED
     1 PERIMETER'//23X,'0-TH',7X,'1ST',7X,'2ND',7X,'3RD',7X,'4TH')
      WRITE(LP,300)(PSF(J),J = 1,LR)
  300 FORMAT(/20X,5F10.5)
      RETURN
      END
C   ****************************************************************
C   *                                                              *
```

```
C   *  SUBPROGRAM TO UPDATE THE NODAL FLOW AREA,WETTED PERIMETER ,   *
C   *                                                                *
C   *  MANNING'S ROUGHNESS COEFF. & THE CHANGE IN HYDRAULIC          *
C   *                                                                *
C   *  RADIUS WITH RESPECT TO DEPTH                                  *
C   *                                                                *

       SUBROUTINE GEOMTR(ASF,AX,HYD,QX,PSF,PX,YX,RC,N1,LDIN)
C   *                                                                *
C   *                                                                *
C   ******************************************************************
       DIMENSION AX(LDIN),HYD(LDIN),PX(LDIN),YX(LDIN),ASF(5),PSF(5),
     1 QX(LDIN),RC(LDIN)
C
C      ---COMPUTE THE NODAL FLOW AREA,WETTED PERIMETER AND
C      MANNING'S ROUGHNESS COEFF.
       DO 200 J = 1,N1
       AX(J) = ASF(1) + ASF(2)*YX(J) + ASF(3)*YX(J)**2 +
     1 ASF(4)*YX(J)**3 + ASF(5)*YX(J)**4
C
       PX(J) = PSF(1) + PSF(2)*YX(J) + PSF(3)*YX(J)**2 +
     1 PSF(4)*YX(J)**3 + PSF(5)*YX(J)**4
C
       RC(J) = 0.03713 + 0.14097E-05*QX(J) + 0.41739E-10*
     1 QX(J)**2 - 0.230004E-14*QX(J)**3
  200  CONTINUE
C      ---COMPUTE THE NODAL RATE OF CHANGE OF HYDR. RADIUS
       DO 300 K = 1,N1
       STOR1 = ASF(2) + 2.*ASF(3)*YX(K) + 3.*ASF(4)*YX(K)**2 +
     1 4.*ASF(5)*YX(K)**3
C
       STOR2 = PSF(2) + 2.*PSF(3)*YX(K) + 3.*PSF(4)*YX(K)**2 +
     1 4.*PSF(5)*YX(K)**3
C
       HYD(K) = (STOR1*PX(K) - STOR2*AX(K))/PX(K)**2
       HYD(K) = ABS(HYD(K))
  300  CONTINUE
       RETURN
       END
C   *************************************************************
C   *                                                           *
C   *                                                           *
C   *     SUBPROGRAM TO GENERATE COLUNMN VECTOR (2NX1)          *
C   *                                                           *

       SUBROUTINE VECTR(CXV,FAC,QLAT,YX,VX,SO,G,XL,QT,
     1 RX,N1,AX,PX,QRE,LDN,LDIN,LDIM,JSWTCH)
C   *                                                           *
C   *************************************************************
       DIMENSION CXV(LDIM),QLAT(LDN),YX(LDIN),VX(LDIN),XL(LDN)
       DIMENSION AX(LDIN),QRE(LDIN),PX(LDIN),RX(LDIN)
C
C      --- UPSTREAM NODAL CALCULATION
       K = 1
       PS = 4./3.
       LSTP = N1 - 2
       SPK = QT/AX(1)
       RY1 = (AX(1)/PX(1))**PS
       RY2 = (AX(2)/PX(2))**PS
C      ---UPSTREAM BOUNDARY EQS.
```

```
      A =(2.*XL(1) + FAC*(VX(2) - 4.*SPK))*YX(1)
      B = (XL(1) + FAC*(2.*VX(2) + SPK))*YX(2)
      C = 3.*FAC*QLAT(1)*XL(1)
      G1 = XL(1)*RX(1)*FAC*(2.*VX(1)**2/RY1 + VX(2)**2/RY2)
      P1 = 3.*FAC*(YX(2) - YX(1))
      S1 = 3.*FAC*XL(1)*SO
      IF(JSWTCH - 1) 50,50,80
   50 A = (2.*XL(1)-FAC*( VX(2)-4.* SPK))*YX(1)
      B = (XL(1) - FAC*(2.*VX(2)+ SPK))*YX(2)
      CXV(1) = A + B+ C
      CXV(2) = S1 - G1 - P1
      GO TO 90
   80 CXV(1) = A + B - C
      CXV(2) = G1 + P1 - S1
C     ---INTERIOR NODAL CALCULATION.
   90 DO 200 J = 1,LSTP
      K = K + 2
      M = K + 1
      RD1 = (AX(J)/PX(J))**PS
      RD2 = (AX(J+1)/PX(J+1))**PS
      RD3 = (AX(J+2)/PX(J+2))**PS
      AI = (XL(J) - FAC*(VX(J+1) + 2.*VX(J)))*YX(J)
      BI = (2.*(XL(J)+XL(J+1)) + FAC*(VX(J+2)-VX(J)))*YX(J+1)
      CI = (XL(J+1) + FAC*(2.*VX(J+2)+VX(J+1)))*YX(J+2)
      DI = 3.*FAC*(QLAT(J)*XL(J) + QLAT(J+1)*XL(J+1))
      HI = RX(J+1)*FAC*(XL(J)*VX(J)**2/RD1 + 2.*(XL(J) +
     1 XL(J+1))*VX(J+1)**2/RD2 + XL(J+1)*VX(J+2)**2/RD3)
      PI = 3.*FAC*(YX(J+2) - YX(J))
      SI = 3.*FAC*(XL(J) + XL(J+1))*SO
      IF(JSWTCH - 1)100,100,150
  100 AI = (XL(J) + FAC*(VX(J+1) +2.*VX(J)))*YX(J)
      BI = (2.*(XL(J)+XL(J+1)) - FAC*(VX(J+2)-VX(J)))*YX(J+1)
      CI = (XL(J+1) - FAC*(2.*VX(J+2)+VX(J+1)))*YX(J+2)
      CXV(K) = AI + BI + CI + DI
      CXV(M) = SI - HI - PI
      GO TO 200
  150 CXV(K) = AI + BI + CI - DI
      CXV(M) = HI + PI - SI
  200 CONTINUE
C     ---DOWNSTREAM NODAL CALCULATION.
      HT1 = AX(N1-1)/PX(N1-1)
      HT2 = AX(N1)/PX(N1)
      RY3 = HT1**PS
      RY4 = HT2**PS
      AN = (XL(N1-1) - FAC*(VX(N1)+ 2.*VX(N1-1)))*YX(N1-1)
      BN = (2.*XL(N1-1) + FAC*(4.*VX(N1) -VX(N1-1)))*YX(N1)
      CN = 3.*FAC*QLAT(N1-1)*XL(N1-1)
      GN = XL(N1-1)*RX(N1)*FAC*(VX(N1-1)**2/RY3 + 2.*VX(N1)**2/RY4)
      PN = 3.*FAC*(YX(N1) - YX(N1-1))
      SN = 3.*FAC*XL(N1-1)*SO
      IF(JSWTCH - 1)300,300,400
  300 AN = (XL(N1-1) + FAC*(VX(N1)+2.*VX(N1-1)))*YX(N1-1)
      BN = (2.*XL(N1-1) - FAC*(4.*VX(N1)-VX(N1-1)))*YX(N1)
      CXV(LDIF-1) = AN + BN + CN
      CXV(LDIF) = SN - GN - PN
      GO TO 500
  400 CXV(LDIF-1) = AN + BN - CN
      CXV(LDIF) = GN + PN - SN
  500 RETURN
      END
```

```
C     ************************************************************
C     *                                                          *
C     *                                                          *
C     *     SUBPROGRAM TO GENERATE THE JACOBIAN MATRIX.          *
C     *                                                          *
            SUBROUTINE JACOBI(BTR,FAC,YX,VX,XL,QLAT,QT,RX,
         1 AX,PX,QRE,HYD,SO,G,LDIM,LDN,LDIN,N1,LK)
C     *                                                          *
C     ************************************************************
C
            DIMENSION BTR(LDIM,LK),QLAT(LDN),VX(LDIN),YX(LDIN),XL(LDN),
         1 AX(LDIN),QRE(LDIN),PX(LDIN),HYD(LDIN),RX(LDIN)
            K = 1
            PM = 7./3.
            PS = 4./3.
            SPK = QT/AX(1)
            SPP = QT*YX(2)/(AX(1)*YX(1))
            RD1 = AX(1)/PX(1)
            RD2 = AX(2)/PX(2)
C          ---EVALUATE JACOBIAN TERMS FOR UPSTREAM NODES.
            BTR(1,1) = 0.0
            BTR(1,2) = 0.0
            BTR(2,1) = 0.0
            BTR(2,2) = 0.0
            BTR(1,3) = 2.*XL(1) + FAC*(VX(2) - SPP)
            BTR(1,4) = 0.0
            BTR(1,5) = XL(1) + FAC*(2.*VX(2) + SPK)
            BTR(1,6) = FAC*(YX(1) + 2.*YX(2))
            BTR(2,3) = -FAC*(2.*XL(1)*RX(1)*PS*VX(1)**2*HYD(1)/RD1**PM+3.)
            BTR(2,4) = 4.*XL(1)*RX(1)*FAC*VX(1)/RD1**PS
            BTR(2,5) = -FAC*(XL(1)*RX(1)*PS*VX(2)**2*HYD(2)/RD2**PM-3.)
            BTR(2,6) = 2.*XL(1)*RX(1)*FAC*VX(2)/RD2**PS
C
C          ---EVALUATE JACOBIAN TERMS FOR INTERIOR NODES.
            LSTP = N1 - 2
            DO 300 J =1,LSTP
            K = K + 2
            M = K + 1
            RD1 = AX(J)/PX(J)
            RD2 =  AX(J+1)/PX(J+1)
            RD3 =  AX(J+2)/PX(J+2)
            BTR(K,1) = XL(J) - FAC*(VX(J+1) + 2.*VX(J))
            BTR(K,2) = -FAC*(2.*YX(J) + YX(J+1))
            BTR(K,3) = 2.*(XL(J)+XL(J+1)) + FAC*(VX(J+2) -VX(J))
            BTR(K,4) = FAC*(YX(J+2) - YX(J))
            BTR(K,5) = XL(J+1) + FAC*(2.*VX(J+2) + VX(J+1))
            BTR(K,6) = FAC*(YX(J+1) + 2.*YX(J+2))
            BTR(M,1) = -FAC*(PS*RX(J+1)*XL(J)*VX(J)**2*HYD(J)/RD1**PM + 3.)
            BTR(M,2) = 2.*RX(J+1)*FAC*XL(J)*VX(J)/RD1**PS
            BTR(M,3) = -2.*FAC*PS*RX(J+1)*(XL(J) + XL(J+1))*VX(J+1)**2*
         1 HYD(J+1)/RD2**PM
            BTR(M,4) = 4.*FAC*RX(J+1)*(XL(J) + XL(J+1))*VX(J+1)/RD2**PS
            BTR(M,5) = -FAC*(PS*RX(J+1)*XL(J+1)*VX(J+2)**2*HYD(J+2)/
         1 RD3**PM - 3.)
            BTR(M,6) = 2.*FAC*RX(J+1)*XL(J+1)*VX(J+2)/RD3**PS
       300  CONTINUE
C          ---EVALUATE JACOBIAN TERMS FOR DOWNSTREAM NODES.
            LB = LDIM - 1
            RDN1 = AX(N1-1)/PX(N1-1)
```

```
      RDN2 = RX(N1)/PX(N1)
      BTR(LB,1) = XL(N1-1) - FAC*(VX(N1)+2.*VX(N1-1))
      BTR(LB,2) = -FAC*(2.*YX(N1-1) + YX(N1))
      BTR(LB,3) = 2.*XL(N1-1) + FAC*(4.*VX(N1)-VX(N1-1))
      BTR(LB,4) = FAC*(4.*YX(N1) - YX(N1-1))
      BTR(LDIM,1) = -FAC*(PS*RX(N1)*XL(N1-1)*VX(N1-1)**2*HYD(N1-1)/
     1 RDN1**PW + 3.)
      BTR(LDIM,2) = 2.*FAC*RX(N1)*XL(N1-1)*VX(N1-1)/RDN1**FS
      BTR(LDIM,3) = -FAC*(2.*XL(N1-1)*PS*RX(N1)*VX(N1)**2*HYD(N1)/
     1 RDN2**PW - 3.)
      BTR(LDIM,4) = 4.*FAC*RX(N1)*XL(N1-1)*VX(N1)/RDN2**PS
      BTR(LB,5) = 0.0
      BTR(LB,6) = 0.0
      BTR(LDIM,5) = 0.0
      BTR(LDIM,6) = 0.0
      RETURN
      ENC
C    ***********************************************************
C    *                                                         *
C    *                                                         *
C    *    SUBPROGRAM TO SOLVE THE BI-TRIDIAGONAL MATRIX        *
C    *                                                         *
C    *                                                         *

      SUBROUTINE BTRICG(COL,VEL,YX,VX,LDIM,LDIN,N1,LK)
C    *                                                         *
C    ***********************************************************
      DIMENSICN VEL(LDIM,LK),YX(LDIN),VX(LDIN),COL(LDIM),
     1 BETA(4),DETA(2),SAC(26,4),GAMA(26,2)
C
C     ---PERFCRM MATRIX REDUCTION OPERATION.
      K = 1
      DC 200 J = 1,N1
      IF(J - 1)100,100,150
  100 BETA(1) = VEL(1,3)
      BETA(2) = VEL(1,4)
      BETA(3) = VEL(2,3)
      BETA(4) = VEL(2,4)
      DETA(1) = COL(1)
      DETA(2) = COL(2)
      ZU = BETA(1)*BETA(4) - BETA(2)*BETA(3)
      IF(ZU .EG. 0.0)ZU = 0.001
      SAC(1,1) = (BETA(4)*VEL(1,5) - BETA(2)*VEL(2,5))/ZU
      SAC(1,2) = (BETA(4)*VEL(1,6) - BETA(2)*VEL(2,6))/ZU
      SAC(1,3) = (BETA(1)*VEL(2,5) - BETA(3)*VEL(1,5))/ZU
      SAC(1,4) = (BETA(1)*VEL(2,6) - BETA(3)*VEL(1,6))/ZU
      GO TO 160
  150 K = K + 2
      M = K + 1
      BETA(1) = VEL(K,3) - VEL(K,1)*SAC(J-1,1)-VEL(K,2)*SAC(J-1,3)
      BETA(2) = VEL(K,4)-VEL(K,1)*SAC(J-1,2)-VEL(K,2)*SAC(J-1,4)
      BETA(3) = VEL(M,3)-VEL(M,1)*SAC(J-1,1)-VEL(M,2)*SAC(J-1,3)
      BETA(4) = VEL(M,4)-VEL(M,1)*SAC(J-1,2)-VEL(M,2)*SAC(J-1,4)
C
      DETA(1) = COL(K)-VEL(K,1)*GAMA(J-1,1)-VEL(K,2)*GAMA(J-1,2)
      DETA(2) = COL(M)-VEL(M,1)*GAMA(J-1,1)-VEL(M,2)*GAMA(J-1,2)
C
      ZU = BETA(1)*BETA(4) - BETA(2)*BETA(3)
      IF(ZU .EQ. 0.0)ZU = 0.001
      SAC(J,1) = (BETA(4)*VEL(K,5)-BETA(2)*VEL(M,5))/ZU
      SAC(J,2) = (BETA(4)*VEL(K,6)-BETA(2)*VEL(M,6))/ZU
      SAC(J,3) = (BETA(1)*VEL(M,5) -BETA(3)*VEL(K,5))/ZU
      SAC(J,4) = (BETA(1)*VEL(M,6) -EETA(3)*VEL(K,6))/ZU
C
  180 GAMA(J,1) = (BETA(4)*DETA(1)-BETA(2)*DETA(2))/ZU
      GAMA(J,2) = (BETA(1)*DETA(2)-BETA(3)*DETA(1))/ZU
  200 CONTINUE
C     ---COMPUTE SOLUTION VIA RECURSIVE EQ.
      LIMIT = N1 - 1
      JK = LIMIT
      YX(N1) = GAMA(LDIN,1)
      VX(N1) = GAMA(LCIN,2)
      DO 300 L = 1,LIMIT
      YX(JK) = GAMA(JK,1)-SAC(JK,1)*YX(JK+1)-SAC(JK,2)*VX(JK+1)
      VX(JK) = GAMA(JK,2)-SAC(JK,3)*YX(JK+1)-SAC(JK,4)*VX(JK+1)
      JK = JK - 1
  300 CONTINUE
      RETURN
      END

$ENTRY
```

APPENDIX H

SAMPLE OUTPUT FOR WIDFEM

```
TOTAL NO. OF NODES =   26

TIME STEP =  1800.000 SEC.

CHANNEL BOTTOM SLOPE =    0.0009

TIME WEIGHTING FACTOR =    0.5500

MAX. ITERATION LIMIT =   60

CONVERGENCE CRITERIA FOR DEPTH =    0.0100

CONVERGENCE CRITERIA FOR VELOCITY =    0.1000



   UPSTREAM DISCHARGE HYDROGRAPH

J      TIME PERIOD      MEASURED FLOW

 1            0.0         482.0
 2        50400.0         757.0
 3        64800.0        5590.0
 4        79200.0        7710.0
 5        86400.0       11000.0
 6       100800.0       22980.0
 7       129600.0       11320.0
 8       158400.0        5100.0
 9       172800.0        4110.0
10       208800.0        3104.0
11       345600.0        1722.0



NODE      INITIAL DEPTH      INITIAL DISCHARGE

 1             3.340              482.000
 2             3.340              482.000
 3             3.340              482.000
 4             3.340              482.000
 5             3.340              482.000
 6             3.340              482.000
 7             3.340              482.000
 8             3.340              482.000
 9             3.340              482.000
10             3.340              482.000
11             3.340              482.000
12             3.340              482.000
13             3.340              482.000
14             3.340              482.000
```

```
15            3.340            482.000
16            3.340            482.000
17            3.340            482.000
18            3.340            482.000
19            3.340            482.000
20            3.340            482.000
21            3.340            482.000
22            3.340            482.000
23            3.340            482.000
24            3.340            482.000
25            3.340            482.000
26            3.340            482.000


 REACH        LENGTH         LATERAL FLOW

   1         10560.00        0.000000
   2         10560.00        0.000000
   3         10560.00        0.000000
   4         10560.00        0.000000
   5         10560.00        0.000000
   6         10560.00        0.000000
   7         10560.00        0.000050
   8         10560.00        0.000000
   9         10560.00        0.000000
  10         10560.00        0.000000
  11         10560.00        0.000000
  12         10560.00        0.000000
  13         10560.00        0.000000
  14         10560.00        0.000000
  15         10560.00        0.000000
  16         10560.00        0.000000
  17         10560.00        0.000000
  18         10560.00        0.000000
  19         10560.00        0.000000
  20         10560.00        0.000000
  21         10560.00        0.000000
  22         10560.00        0.000000
  23         10560.00        0.000000
  24         10560.00        0.000000
  25         12672.00        0.000000


 FOURTH-ORDER REGRESSION COEFF. FOR AREA

    0-TH        1ST        2ND        3RD        4TH

  -32.01320  84.60530   5.47340   0.91215  -0.00452

 FOURTH-ORDER REGRESSION COEFF. FOR WETTED        PERIMETER
    0-TH        1ST        2ND        3RD        4TH

   10.08235  57.67010  -4.90130   0.34127  -0.00601
```

```
                           UPSTREAM                                MIDSTREAM                               DOWNSTREAM
TIME(HR.)        DISCHARGE     DEPTH    VELOCITY        DISCHARGE     DEPTH    VELOCITY         DISCHARGE     DEPTH    VELOCITY
    1.00           501.643     2.862      1.800           547.535     3.340      1.587            547.536     3.340      1.587
    2.00           521.285     2.922      1.850           586.984     3.340      1.707            586.982     3.340      1.707
    3.00           540.928     3.139      1.731           615.879     3.340      1.785            615.711     3.340      1.784
    4.00           560.571     3.131      1.781           633.674     3.347      1.829            630.179     3.340      1.826
    5.00           580.214     3.125      1.855           656.857     3.383      1.867            639.148     3.340      1.852
    6.00           599.856     3.234      1.832           693.304     3.460      1.907            644.937     3.340      1.869
    7.00           619.499     3.304      1.828           737.928     3.563      1.945            648.740     3.340      1.880
    8.00           639.142     3.332      1.861           777.436     3.656      1.973            651.258     3.340      1.887
    9.00           658.785     3.397      1.869           805.849     3.722      1.991            652.936     3.340      1.892
   10.00           678.428     3.466      1.867           825.888     3.768      2.004            654.054     3.340      1.896
   11.00           698.071     3.509      1.885           837.967     3.795      2.011            654.801     3.340      1.898
   12.00           717.714     3.551      1.905           840.720     3.802      2.012            655.256     3.340      1.899
   13.00           737.357     3.604      1.915           835.380     3.797      2.011            655.624     3.340      1.900
   14.00           757.000     3.666      1.918           844.768     3.807      2.016            655.876     3.340      1.901
   15.00          1965.249     5.981      2.303           867.685     3.855      2.033            656.433     3.341      1.902
   16.00          3173.499     7.131      2.740           912.249     3.947      2.063            658.450     3.345      1.904
   17.00          4381.742     8.541      2.677           987.039     4.099      2.108            664.086     3.359      1.909
   18.00          5589.992     9.473      2.844          1095.315     4.317      2.168            676.037     3.388      1.919
   19.00          6119.992     9.495      3.044          1235.016     4.576      2.232            696.008     3.437      1.934
   20.00          6649.992    10.162      2.896          1368.973     4.811      2.278            723.015     3.503      1.953
   21.00          7179.992    10.361      3.058          1455.516     4.588      2.286            749.258     3.578      1.962
   22.00          7709.992    10.810      2.948          1613.265     5.215      2.358            779.055     3.648      1.982
   23.00          9354.992    11.762      2.948          2067.883     5.783      2.548            802.541     3.706      1.996
   24.00         10999.990    12.337      3.221          2902.875     6.704      2.772            821.097     3.749      2.006
   25.00         13994.990    13.587      3.300          3814.856     7.594      2.911            836.045     3.785      2.014
   26.00         16989.990    14.959      3.238          4650.125     8.331      2.984            853.498     3.823      2.025
   27.00         15984.980    16.120      3.197          5401.914     8.539      3.027            881.427     3.883      2.043
   28.00         22979.980    17.395      3.015          6116.488     9.473      3.058            929.037     3.982      2.075
   29.00         21522.490    17.075      3.003          6723.230     9.572      3.034            997.431     4.136      2.098
   30.00         20064.990    16.091      3.153          7436.621    10.467      3.042           1103.959     4.363      2.137
   31.00         18607.500    15.862      3.041          8227.418    10.582      3.046           1277.103     4.702      2.196
   32.00         17150.000    15.497      2.935          9127.801    11.528      3.053           1586.659     5.238      2.290
   33.00         15692.490    14.626      3.063         10145.020    12.096      3.064           2107.491     5.986      2.440
   34.00         14235.000    14.210      2.965         11185.910    12.669      3.059           2732.856     6.772      2.561
   35.00         12777.500    13.559      3.023         12417.830    13.224      3.093           3509.136     7.463      2.764
   36.00         11320.000    13.095      2.857         13466.900    13.732      3.087           4202.035     8.053      2.875
   37.00         10542.490    12.313      3.067         14395.320    14.172      3.077           4880.863     8.571      2.967
   38.00          9764.996    11.795      3.076         15136.990    14.538      3.059           5546.050     9.042      3.040
   39.00          8987.496    11.359      3.052         15641.020    14.798      3.036           6174.414     9.477      3.086
   40.00          8209.996    10.926      3.035         15865.120    14.944      3.012           6790.938     9.886      3.119
   41.00          7432.496    10.390      3.047         15840.180    14.979      2.991           7404.016    10.277      3.144
   42.00          6655.000     9.859      3.023         15556.810    14.916      2.973           8018.965    10.655      3.162
   43.00          5877.500     9.317      2.991         15175.810    14.770      2.959           8637.613    11.022      3.176
   44.00          5100.000     9.038      2.795         14605.430    14.542      2.946           9258.391    11.378      3.185
   45.00          4852.496     8.762      2.795         13905.190    14.253      2.933           9875.133    11.722      3.190
   46.00          4604.996     8.299      2.924         13217.840    13.912      2.946          10389.570    12.053      3.164
   47.00          4357.456     8.292      2.812         12330.150    13.914      2.930          11029.920    12.369      3.178
   48.00          4109.996     7.917      2.886         11413.670    13.059      2.924          11587.010    12.667      3.171
   49.00          4009.400     7.997      2.760         10441.100    12.948      2.917          12106.830    12.942      3.163
   50.00          3908.799     7.827      2.328          9453.613    11.990      2.911          12557.680    13.187      3.149
```

51.00	3808.199	7.737	2.786	8483.125	11.400	2.906	12929.510	13.398	3.131
52.00	3707.600	7.686	2.769	7571.227	10.602	2.900	13206.930	13.567	3.111
53.00	3607.000	7.520	2.788	6754.367	10.226	2.894	13279.530	13.651	3.090
54.00	3506.399	7.495	2.735	6058.941	9.702	2.885	13439.520	13.764	3.067
55.00	3405.800	7.334	2.762	5456.086	9.253	2.875	13382.750	13.786	3.044
56.00	3305.200	7.279	2.715	5055.020	8.889	2.863	13207.280	13.753	3.020
57.00	3204.600	7.148	2.724	4728.902	8.606	2.849	12912.700	13.665	2.995
58.00	3104.000	7.061	2.693	4480.469	8.389	2.835	12499.500	13.519	2.969
59.00	3067.631	7.056	2.674	4285.047	8.222	2.821	11969.550	13.314	2.941
60.00	3031.262	6.995	2.681	4134.473	8.086	2.806	11327.800	13.046	2.911
61.00	2994.894	6.980	2.660	4002.195	7.970	2.791	10584.830	12.711	2.878
62.00	2958.526	6.926	2.664	3882.707	7.864	2.777	9760.715	12.309	2.845
63.00	2922.157	6.902	2.647	3770.389	7.763	2.762	8887.164	11.842	2.813
64.00	2885.789	6.856	2.646	3662.174	7.665	2.747	8006.555	11.319	2.786
65.00	2849.421	6.826	2.633	3562.933	7.571	2.734	7165.879	10.759	2.769
66.00	2813.052	6.784	2.628	3472.680	7.483	2.722	6407.211	10.155	2.762
67.00	2776.684	6.748	2.619	3393.886	7.404	2.712	5758.938	9.663	2.764
68.00	2740.315	6.707	2.613	3325.938	7.335	2.704	5231.172	9.193	2.770
69.00	2703.947	6.666	2.605	3266.877	7.274	2.697	4816.715	8.802	2.777
70.00	2667.578	6.623	2.600	3214.726	7.219	2.690	4496.770	8.491	2.780
71.00	2631.210	6.578	2.595	3167.942	7.169	2.685	4248.520	8.246	2.778
72.00	2594.842	6.532	2.591	3125.397	7.123	2.679	4051.337	8.052	2.771
73.00	2558.474	6.469	2.597	3100.637	7.082	2.686	3953.182	7.898	2.805
74.00	2522.109	6.447	2.565	3083.990	7.043	2.685	3894.767	7.772	2.805
75.00	2485.737	6.354	2.573	3040.694	7.006	2.673	3735.136	7.664	2.767
76.00	2449.368	6.330	2.589	3005.066	6.969	2.667	3639.834	7.566	2.762
77.00	2413.000	6.344	2.535	2966.977	6.930	2.659	3532.841	7.481	2.743
78.00	2376.631	6.197	2.601	2913.952	6.890	2.652	3422.475	7.405	2.735
79.00	2340.263	6.281	2.531	2901.610	6.849	2.655	3405.922	7.339	2.743
80.00	2303.894	6.131	2.551	2838.401	6.806	2.640	3294.654	7.283	2.714
81.00	2267.526	6.112	2.542	2825.662	6.762	2.643	3289.079	7.230	2.725
82.00	2231.157	6.118	2.494	2775.443	6.716	2.627	3212.459	7.182	2.699
83.00	2194.789	5.981	2.550	2715.998	6.670	2.618	3155.470	7.136	2.696
84.00	2158.420	6.037	2.490	2703.354	6.625	2.621	3136.615	7.092	2.696
85.00	2122.052	5.916	2.501	2640.008	6.583	2.604	3067.410	7.050	2.679
86.00	2085.684	5.877	2.487	2625.351	6.542	2.608	3054.743	7.007	2.683
87.00	2049.315	5.858	2.464	2584.518	6.502	2.591	3001.535	6.965	2.665
88.00	2012.947	5.777	2.480	2532.663	6.459	2.582	2944.252	6.922	2.657
89.00	1976.578	5.762	2.439	2516.731	6.415	2.584	2930.188	6.879	2.659
90.00	1940.210	5.676	2.467	2455.928	6.369	2.566	2862.263	6.835	2.642
91.00	1903.842	5.679	2.408	2443.469	6.323	2.569	2850.243	6.790	2.646
92.00	1867.473	5.595	2.419	2385.747	6.278	2.540	2786.157	6.746	2.617
93.00	1831.105	5.525	2.406	2352.425	6.232	2.536	2751.941	6.701	2.615
94.00	1794.737	5.475	2.408	2313.443	6.184	2.527	2711.081	6.657	2.607
95.00	1758.368	5.450	2.380	2274.423	6.135	2.518	2673.275	6.614	2.600
96.00	1722.000	5.365	2.402	2219.929	6.085	2.509	2620.364	6.571	2.593

APPENDIX I

COMPUTER PROGRAM LISTING FOR WIKFEM

The computer program for the weighted implicit kinematic model is for an idealized channel . It has a built-in option to route flood in a trapezoidal, triangular, or rectangular channel. For the first two geometries, the right- and left-side slopes, captioned as ZRS and ZLS should have assigned values other than zeros, except for rectangular channel. The triangular geometry will have zero width for input value.

The definition of the variables and symbols used in the computer program is provided in the comment page of the program listing. Any temporary storage variables are not included because their definitions are obvious. Instruction for the input data is provided in the MAIN program for each READ STATEMENT and is self-explanatory.

```
$JOB            ,TIME=(0,40)
C
C
C    **************************************************************
C    *                                                            *
C    *               1 - DIMENSIONAL STREAMFLOW ROUTING MODEL     *
C    *                                                            *
C    *                                                            *
C    **************************************************************
C    *                                                            *
C              IMPLICIT KINEMATIC FINITE ELEMENT METHOD SOLVED    *
C    *                                                            *
C    *              BY ITERATIVE NEWTON-RAPHSON TECHNIQUE         *
C    *                                                            *
C    **************************************************************
C    *                                                            *
C    *               ... DEFINITION OF TERMS ...                  *
C    *                                                            *
C    *VARIABLES UNITS ARE AS FOLLOWS: TIME(SEC),LENGTH(FT)        *
C    *DEPTH(FT),VELOCITY(FT. PER SEC),DISCHARGE(CFS)              *
C    *ACC IS THE ACCELERATION OF GRAVITY,32.2FT. PER SEC PER SEC  *
C    *QLAT IS THE LATERAL FLOW TERM,FT. PER SEC                   *
C    *TLL IS THE TOTAL LENGTH OF CHANNEL REACH BEING INVESTIGATED *
C    *SO IS THE CONSTANT CHANNEL SLOPE                            *
C    *RN IS THE MANNING ROUGHNESS COEFF.                          *
C    *XL IS THE NODAL SPACING.                                    *
C    *YO IS THE INITIAL UNIFORM NORMAL DEPTH.                     *
C    *VO IS THE INITIAL UNIFORM NORMAL VELOCITY                   *
C    *N1 IS THE TOTAL NUMBER OF NODES                             *
C    *QMAX IS THE PEAK FLOOD DISCHARGE OF INFLOW HYDROGRAPH       *
C    *T IS THE TIME STEP (SECONDS)                                *
C    *TMX IS THE TIME PERIOD BETWEEN QO & QMAX                    *
C    *TAP IS THE TIME PERIOD AFTER QMAX UNTIL QO                  *
C    *TPRINT IS THE TIME FOR INITIAL PRINTING (SECONDS)           *
C    *TTA IS THE INCREMENTAL PRINTING TIME (SECONDS)              *
C    *---IF TIME STEP IS GREATER THAN TTA PRINTING WILL BE ---    *
C    *       PERFORMED AT THE INCREMENT OF THE TIME STEP,T---     *
C    *TETHA IS THE TIME WEIGHTING FACTOR.                         *
C    *YN IS THE CALCULATED DETH OF FLOW                           *
C    *VN IS THE CORRESPONDING VELOCITY OF FLOW                    *
C    *ZSR IS THE TRAPEZOIDAL CHANNEL RIGHT SIDE SLOPE             *
C    *ZSL IS THE TRAPEZOIDAL CHANNEL LEFT SIDE SLOPE              *
C    *IMAX IS THE MAX. ITERATION LIMIT                            *
C    *DETA IS THR CONVERGENCE CRITERIA FOR DEPTH                  *
C    *BSR IS THE JACOBIAN MATRIX OF DIMENSION (N1X3)              *
C    *CXV1 & CXV2 ARE THE (N1X1) COLUMN VECTORS EVALUATED AT      *
C    *(1-TETHA) & TETHA RESPECTIVELY.                             *
C    *LDIM,LDIN ARE THE VARIABLE DIMENSIONING PARAMETERS          *
C    *                                                            *
C    **************************************************************
      DIMENSION BSR(21,3),CSV1(21),CSV2(21),DTPH(21),QLAT(20),
     1 VELY(21),YO(21),VO(21),YN(21),VN(21),XL(20),QRE(21)
      DATA XL,QLAT,ACC/20*528.,20*0.0,32.2/
      DATA LDIN,LK/21,3/
      DATA NR,LP/5,6/
C
C     ---READ DATA AND ECHOE CHECK
      READ(NR,50)TPRINT,TTA,TSUM,T,YB,ZSR,ZSL
   50 FORMAT(7F10.2)
      READ(NR,60)QMAX,TMX,TAP,TLL,B1
   60 FORMAT( 5F10.2)
      READ(NR,80)TETHA,DETA,IMAX,SO,RN,N1,NN
   80 FORMAT( 2F10.4,I10,2F10.4,2I10)
      JSIZE = TSUM/T
      LDN = LDIN - 1
C
      WRITE(LP,90)
   90 FORMAT( 1H1)
C
      WRITE(LP,100)N1,T,SO,RN
  100 FORMAT( ///20X,'TOTAL NO. OF NODES =',I5//20X,
     1 'TIME STEP =',F10.3,1X,'SEC.'//20X,'CHANNEL BOTTOM SLOPE =',
     2 F10.4,//20X,'MANNING ROUGHNESS COEFF. =',F10.4)
      WRITE(LP,120)TETHA,IMAX,DETA,ZSR,ZSL
  120 FORMAT( //20X,'TIME WEIGHTING FACTOR =',F10.4//20X,
     1 'MAX. ITERATION LIMIT =',I5//20X,
     2 'CONVERGENCE CRITERIA FOR DEPTH =',F10.5//20X,
     3 'TRAPEZOIDAL CHANNEL SIDE SLOPES: RIGHT =',F10.2,
     4 5X,'LEFT =',F10.2)
      WRITE(LP,125)XL(1),QLAT(1)
  125 FORMAT( //20X,'NODAL SPACING =',F10.3,1X,'FT.',5X,
     1 'LATERAL FLOW =',F10.3,1X,'FT.PER SEC.')
      TWRIT = TPRINT
      JSTP = N1 - 2
C     --- CALCULATE THE INITIAL NORMAL DISCHARGE & VELOCITY.
      PR = 2./3.
      CM = 1.486/RN*SQRT( SO)
      ZPP = SQRT(1. + ZSR**2) + SQRT(1. + ZSL**2)
      AE = B1*YB + .5*YB**2*(ZSR + ZSL)
      PE = B1 + YB*ZPP
      VB = CM*(AE/PE)**PR
      QB = AE*VB
      DO 230 J =1,N1
      YO(J) = YB
      VO(J) = VB
      QRE(J) = QB
  230 CONTINUE
      QO = QRE(1)
      WRITE(LP,250)YO(1),VO(1),QRE(1)
  250 FORMAT( //20X,'INITIAL DEPTH =',F10.3,3X,
     1 'INITIAL VEL. =',F10.3,3X,'INITIAL DISCH. =',F10.3)
C
      WRITE(LP,140)
  140 FORMAT( ////28X,'UPSTREAM',24X,'MIDSTREAM',24X,'DOWNSTREAM')
      WRITE(LP,145)
  145 FORMAT( 4X,'TIME(MIN)',7X,'DISCHARGE',4X,'DEPTH',3X,'VELOCITY',6X,
     1 'DISCHARGE',4X,'DEPTH',3X,'VELOCITY',6X,'DISCHARGE',4X,'DEPTH',
     2 3X,'VELOCITY')
C
C     ---USE INITIAL VELOCITY & DEPTH OF FLOW AS GUESS VALUES
C     TO INITIATE SIMULATION.
      DO 280 K = 1,N1
      YN(K) = YO(K)
      VN(K) = VO(K)
  280 CONTINUE
      FAC1 = TETHA*T
      FAC2 = (1. - TETHA)*T
C     --- SET LOOP FOR TIME SIMULATION.
      QR = QO
      DO 900 JL =1,JSIZE
      TIME = FLOAT(JL)*T
C     --- UPDATE UPSTREAM BOUNDARY CONDITION.
      TD = TIME - TMX
      IF(TD)450,450,480
  450 QR = QO + (QMAX - QO)/TMX*TIME
      GO TO 520
  480 IF(TD - TAP)490,490,500
  490 QR = QMAX - (QMAX - QO)/TAP*TD
      GO TO 520
  500 QR = QO
  520 CONTINUE
C     ---CALL SUBROUTINE TO GENERATE COLUMN VECTOR (N1 X 1)
      JSWTCH = 1
      CALL VECTR(B1,CSV1,FAC2,QLAT,YO,VO,XL,QR,RN,N1,ZSR,ZSL,
     1 LDN,LDIN,LK,JSWTCH)
C
C     ---GENERATE JACOBIAN MATRIX.
      LUP = 0
      CALL JACOBI(BSR,B1,FAC1,YN,VN,XL,QLAT,QR,RN,ZSR,ZSL,
     1 LDN,LDIN,N1,LK)
C     --- ITERATE TO CONVERGENCE FOR EACH TIME STEP.
      DO 590 LL =1,IMAX
      JSWTCH = 2
      LUP = LUP + 1
      CALL VECTR(B1,CSV2,FAC1,QLAT,YN,VN,XL,QR,RN,N1,ZSR,ZSL,
     1 LDN,LDIN,LK,JSWTCH)
      DO 530 K =1,N1
      CSV2(K) = CSV2(K) - CSV1(K)
  530 CONTINUE
C     ---OBTAIN SOLUTION VIA TRI-DIAGONAL SUBPROGRAM.
      CALL TRIMTD(BSR,CSV2,DTPH,N1,LDIN,LK)
      JS = N1 - 1
      DO 535 L = 1,N1
      DP1 = YN(L) - DTPH(L)
      AW = B1*DP1 + .5*DP1**2*(ZSR + ZSL)
      PW = B1 + DP1*ZPP
      VN(L) = CM*(AW/PW)**PR
  535 CONTINUE
C     --- CHECK FOR RELATIVE CONVERGENCE FOR ALL VARIABLES.
      JERR = 0
      DO 560 J =1,N1
      YB1 = ABS(DTPH(J))
      YB2 = YN(J) - DTPH(J)
      YB3 = MAX1(ABS(YB2),ABS(YN(J)))
      IF(YB3 .LE. 0.0)GO TO 570
      YERROR = YB1/YB3
      IF(YERROR .LE. DETA)JERR = JERR + 1
  560 CONTINUE
C
C     --- SWITCH CURRENT VALUES OF DEPTH OF FLOW TO OLD ONES.
  570 DO 580 L=1,N1
      YN(L) = YN(L) - DTPH(L)
  580 CONTINUE
      IF(JERR .EQ. N1) GO TO 600
C     --- CHECK IF SPECIFIED ITERATION LIMIT IS EXCEEDED.
C
      IF(JERR .LT. N1 .AND. LL .GE. IMAX) GO TO 920
C
C     ---UPDATE THE JACOBIAN MATRIX AT EVERY 3 ITERATIOS.
      IF(LUP - 3)590,585,585
  585  CALL JACOBI(BSR,B1,FAC1,YN,VN,XL,QLAT,QR,RN,ZSR,ZSL,
     1 LDN,LDIN,N1,LK)
       LUP = 0
  590  CONTINUE
C      --- UPDATE DEPTHS & VELOCITIES OF PREVIOUS TIME STEP.
  600  DO 680 J=1,N1
       VO(J) = VN(J)
       YO(J) = YN(J)
  680  CONTINUE
C      --- PRINT OUT RESULTS.
       IF(TIME - TPRINT + .003)750,700,700
  700  TPRINT = TPRINT + TTA
       TM = TIME/TWRIT
       DO 710 J = 1,N1
       AREA = B1*YN(J) + .5*YN(J)**2*(ZSR + ZSL)
       QRE(J) = AREA*VN(J)
  710  CONTINUE
       WRITE(LP,720)TM,QRE(1),YN(1),VN(1),QRE(11),YN(11),VN(11),
     1  QRE(N1),YN(N1),VN(N1)
  720  FORMAT(2X,F10.2,5X,3F10.3,5X,3F10.3,5X,3F10.3)
  750  IF(TIME - TSUM)900,950,950
C      --- ADVANCE THE TIME STEP.
  900  CONTINUE
  920  WRITE(LP,930)
  930  FORMAT(///10X,'MAX. ITERATION LIMIT EXCEEDED.')
  950  STOP
       END
C    ******************************************************
C    *                                                    *
C    *    SUBPROGRAM TO GENERATE COLUNMN VECTOR ( NX1)    *
C    *                                                    *
C    *                                                    *

       SUBROUTINE VECTR(B1,CXV,FAC,QLAT,YX,VX,XL,QT,
     1 RN,N1,ZSR,ZSL,LDN,LDIN,LK,JSWTCH)
C    *                                                    *
C    ******************************************************
       DIMENSION CXV(LDIN),QLAT(LDN),YX(LDIN),VX(LDIN),XL(LDN)
C
C      --- UPSTREAM NODAL CALCULATION
       LSTP = N1 - 2
       AREA = B1*YX(1) + .5*YX(1)**2*(ZSR + ZSL)
       SPK = QT/AREA
       A =(2.*XL(1) + FAC*(VX(2) - 4.*SPK))*YX(1)
       B = (XL(1) + FAC*(2.*VX(2) + SPK))*YX(2)
       C = 3.*FAC*QLAT(1)*XL(1)
       IF(JSWTCH - 1) 50,50,80
   50  A = (2.*XL(1)-FAC*(VX(2)-4.* SPK))*YX(1)
       B = (XL(1) - FAC*(2.*VX(2)+ SPK))*YX(2)
       CXV(1) = A + B+ C
       GO TO 90
   80  CXV(1) = A + B - C
C      ---INTERIOR NODAL CALCULATION.
   90  DO 200 J = 1,LSTP
       K = J + 1
       AI = (XL(J) - FAC*(VX(J+1) + 2.*VX(J)))*YX(J)
       BI = (2.*(XL(J)+XL(J+1)) + FAC*(VX(J+2)-VX(J)))*YX(J+1)
       CI = (XL(J+1) + FAC*(2.*VX(J+2)+VX(J+1)))*YX(J+2)
       DI = 3.*FAC*(QLAT(J)*XL(J) + QLAT(J+1)*XL(J+1))
       IF(JSWTCH - 1)100,100,150
  585 CALL JACOBI(BSR   1,FAC1,YN,VN,XL,QLAT,QR,RN,ZSR,ZSL,
     1 LDN,LDIN,N1,LK)
      LUP = 0
  590 CONTINUE
C     --- UPDATE DEPT    & VELOCITIES OF PREVIOUS TIME STEP.
  600 DO 680 J=1,N1
      VO(J) = VN(J)
      YO(J) = YN(J)
  680 CONTINUE
C     --- PRINT OUT R   ULTS.
      IF(TIME - TPRIN   + .003)750,700,700
  700 TPRINT = TPRINT    TTA
      TM = TIME/TWRIT
      DO 710 J = 1,N1
      AREA = B1*YN(J)    .5*YN(J)**2*(ZSR + ZSL)
      QRE(J) = AREA*V   J))
  710 CONTINUE
      WRITE(LP,720)TM   RE(1),YN(1),VN(1),QRE(11),YN(11),VN(11),
     1 QRE(N1),YN(N1)   N(N1)
  720 FORMAT(2X,F10.2   X,3F10.3,5X,3F10.3,5X,3F10.3)
  750 IF(TIME - TSUM)   ),950,950
C     --- ADVANCE THE   IME STEP.
  900 CONTINUE
  920 WRITE(LP,930)
  930 FORMAT(///10X,'   (. ITERATION LIMIT EXCEEDED.')
  950 STOP
      END
C   ******************  ****************************************
C   *                                                          *
C   *    SUBPROGRAM TO   NERATE COLUNMN VECTOR ( NX1)          *
C   *                                                          *
C   *                                                          *

      SUBROUTINE VECTF   1,CXV,FAC,QLAT,YX,VX,XL,QT,
     1 RN,N1,ZSR,ZSL,LI   LDIN,LK,JSWTCH)
C   *                                                          *
C   ******************  ****************************************
      DIMENSION CXV(LI   ),QLAT(LDN),YX(LDIN),VX(LDIN),XL(LDN)
C
C     --- UPSTREAM NOI  . CALCULATION
      LSTP = N1 - 2
      AREA = B1*YX(1)    .5*YX(1)**2*(ZSR + ZSL)
      SPK = QT/AREA
      A =(2.*XL(1) + F   *(VX(2) - 4.*SPK))*YX(1)
      B = (XL(1) + FAC   2.*VX(2) + SPK))*YX(2)
      C = 3.*FAC*QLAT(   *XL(1)
      IF(JSWTCH - 1) 5   50,80
   50 A = (2.*XL(1)-FA   (VX(2)-4.* SPK))*YX(1)
      B = (XL(1) - FAC   2.*VX(2)+ SPK))*YX(2)
      CXV(1) = A + B+
      GO TO 90
   80 CXV(1) = A + B -
C     ---INTERIOR NODA   CALCULATION.
   90 DO 200 J = 1,LST
      K = J + 1
      AI = (XL(J) - FA   (VX(J+1) + 2.*VX(J)))*YX(J)
      BI = (2.*(XL(J)+   (J+1)) + FAC*(VX(J+2)-VX(J)))*YX(J+1)
      CI = (XL(J+1) +    C*(2.*VX(J+2)+VX(J+1)))*YX(J+2)
      DI = 3.*FAC*(QLA   J)*XL(J) + QLAT(J+1)*XL(J+1))
      IF(JSWTCH - 1)10   100,150
  100 AI = (XL(J) + FAC*(VX(J+1) +2.*VX(J)))*YX(J)
      BI = (2.*(XL(J)+XL(J+1)) - FAC*(VX(J+2)-VX(J)))*YX(J+1)
      CI = (XL(J+1) - FAC*(2.*VX(J+2)+VX(J+1)))*YX(J+2)
      CXV(K) = AI + BI + CI + DI
      GO TO 200
  150 CXV(K) = AI + BI + CI - DI
  200 CONTINUE
C
C     ---DOWNSTREAM NODAL CALCULATION.
      AN = (XL(N1-1) - FAC*(VX(N1)+ 2.*VX(N1-1)))*YX(N1-1)
      BN = (2.*XL(N1-1) + FAC*(4.*VX(N1) -VX(N1-1)))*YX(N1)
      CN = 3.*FAC*QLAT(N1-1)*XL(N1-1)
      IF(JSWTCH - 1)300,300,400
  300 AN = (XL(N1-1) + FAC*(VX(N1)+2.*VX(N1-1)))*YX(N1-1)
      BN = (2.*XL(N1-1) - FAC*(4.*VX(N1)-VX(N1-1)))*YX(N1)
      CXV(LDIN) = AN + BN + CN
      GO TO 500
  400 CXV(LDIN) = AN + BN - CN
  500 RETURN
      END
C    ************************************************************
C    *                                                          *
C    *    SUBPROGRAM TO GENERATE THE JACOBIAN MATRIX.           *
C    *                                                          *

      SUBROUTINE JACOBI(BTR,B1,FAC,YX,VX,XL,QLAT,QT,RN,ZSR,ZSL,
     1 LDN,LDIN,N1,LK)
C    *                                                          *
C    ************************************************************
C
      DIMENSION BTR(LDIN,LK),QLAT(LDN),VX(LDIN),YX(LDIN),XL(LDN)
      AREA = B1*YX(1) + .5*YX(1)**2*(ZSR + ZSL)
      SPP = QT*YX(2)/(B1*YX(1)**2)
      SPK = QT/AREA
C     ---EVALUATE JACOBIAN TERMS FOR UPSTREAM NODES.
      BTR(1,1) = 0.0
      BTR(1,2) = 2.*XL(1) + FAC*(VX(2) - SPP)
      BTR(1,3) = XL(1) + FAC*(2.*VX(2) + SPK)
C
C     ---EVALUATE JACOBIAN TERMS FOR INTERIOR NODES.
C
      LSTP = N1 - 2
      DO 300 J =1,LSTP
      K = J + 1
      BTR(K,1) = XL(J) - FAC*(VX(J+1) + 2.*VX(J))
      BTR(K,2) = 2.*(XL(J)+XL(J+1)) + FAC*(VX(J+2) -VX(J))
      BTR(K,3) = XL(J+1) + FAC*(2.*VX(J+2) + VX(J+1))
  300 CONTINUE
C     ---EVALUATE JACOBIAN TERMS FOR DOWNSTREAM NODES.
      LB = LDIN
      BTR(LB,1) = XL(N1-1) - FAC*(VX(N1)+2.*VX(N1-1))
      BTR(LB,2) = 2.*XL(N1-1) + FAC*(4.*VX(N1)-VX(N1-1))
      BTR(LB,3) = 0.0
      RETURN
      END
C    ************************************************************
C    *                                                          *
C    *      TRI-DIAGONAL SOLUTION ALGORITHM                     *
C    *                                                          *
```

```
      SUBROUTINE TRIMTD (STIFF,RH,YR,N1,LDIN,LK)
C   *                                                              *
C   ****************************************************************
      DIMENSION STIFF(LDIN,LK),RH(LDIN),YR(LDIN)
      DIMENSION G(50),W(50)
C     ---BEGIN TRIANGULAR REDUCTION OPERATION.
      W(1) = STIFF(1,3)/STIFF(1,2)
      G(1) = RH(1)/STIFF(1,2)
      DO 100 J = 2,N1
      SAV1 = STIFF(J,2) - STIFF(J,1)*W(J-1)
      SAV2 = RH(J) - STIFF(J,1)*G(J-1)
      W(J) = STIFF(J,3)/SAV1
      G(J) = SAV2/SAV1
  100 CONTINUE
C     --- OBTAIN SOLUTION VIA RECURSIVE EQ.
      LIMIT = N1 - 1
      K = LIMIT
      YR(N1) = G(N1)
      DO 200 L=1,LIMIT
      YR(K) = G(K) - W(K)*YR(K+1)
      K = K - 1
  200 CONTINUE
      RETURN
      END
SENTRY
```

APPENDIX J

SAMPLE OUTPUT FOR WIKFEM

Two sample outputs for a rectangular channel flood routing using time steps of 300 and 600 seconds respectively are included. Input data are drawn from the example problem given by Viessman et al. (1972). The computer print-out includes the input data and the simulated flow parameters--depth of flow, velocity of flow, and the volumetric flow rate.

```
                        TOTAL NO. OF NODES =   21

                        TIME STEP =   300.000 SEC.

                        CHANNEL BOTTOM SLOPE =    0.0015

                        MANNING ROUGHNESS COEFF. =    0.0200

                        TIME WEIGHTING FACTOR =    1.0000

                        MAX. ITERATION LIMIT =   50

                        CONVERGENCE CRITERIA FOR DEPTH =   0.01000

                        TRAPEZOIDAL CHANNEL SIDE SLOPES: RIGHT =      0.00     LEFT =      0.00

                        NODAL SPACING =   528.000 FT.     LATERAL FLOW =     0.000 FT.PER SEC.

                        INITIAL DEPTH =     6.000   INITIAL VEL. =     6.946   INITIAL DISCH. =   833.495


                               UPSTREAM                              MIDSTREAM                              DOWNSTREAM
TIME(MIN)               DISCHARGE     DEPTH     VELOCITY        DISCHARGE     DEPTH     VELOCITY        DISCHARGE     DEPTH     VELOCITY
     5.00               1122.043      7.420      7.561           883.244      6.251      7.065           841.941      6.043      6.966
    10.00               1419.995      8.813      8.056          1027.937      6.966      7.378           881.761      6.244      7.061
    15.00               1710.687     10.124      8.449          1258.421      8.065      7.802           981.430      6.739      7.282
    20.00               1996.093     11.378      8.772          1538.243      9.351      8.225          1154.723      7.576      7.621
    25.00               1853.343     10.754      8.617          1741.772     10.262      8.487          1373.234      8.598      7.986
    30.00               1704.763     10.098      8.441          1802.278     10.529      8.558          1570.497      9.497      8.269
    35.00               1558.466      9.442      8.253          1747.810     10.289      8.494          1689.050     10.028      8.422
    40.00               1417.880      8.803      8.053          1634.075      9.782      8.352          1712.527     10.132      8.451
    45.00               1272.331      8.130      7.825          1499.338      9.175      8.171          1659.270      9.895      8.385
    50.00               1126.723      7.442      7.570          1359.225      8.533      7.964          1559.368      9.446      8.254
    55.00                981.157      6.737      7.282          1218.384      7.877      7.734          1436.530      8.889      8.081
    60.00                835.581      6.011      6.951          1078.077      7.209      7.478          1304.531      8.280      7.878
```

```
                TOTAL NO. OF NODES =   21

                TIME STEP =   600.000 SEC.

                CHANNEL BOTTOM SLOPE =    0.0015

                MANNING ROUGHNESS COEFF. =    0.0200


                TIME WEIGHTING FACTOR =    1.0000

                MAX. ITERATION LIMIT =   50

                CONVERGENCE CRITERIA FOR DEPTH =   0.01000

                TRAPEZOIDAL CHANNEL SIDE SLOPES: RIGHT =      0.00     LEFT =      0.00


                NODAL SPACING =   528.000 FT.     LATERAL FLOW =     0.000 FT.PER SEC.


                INITIAL DEPTH =     6.000   INITIAL VEL. =     6.946   INITIAL DISCH. =   833.495



                        UPSTREAM                              MIDSTREAM                             DOWNSTREAM
TIME(MIN)       DISCHARGE    DEPTH    VELOCITY        DISCHARGE    DEPTH    VELOCITY        DISCHARGE    DEPTH    VELOCITY
   10.00       1417.652     8.802     8.053          1080.712     7.221     7.483           936.491     6.517     7.185
   20.00       2004.586    11.415     8.781          1559.096     9.445     8.253          1237.687     7.968     7.767
   30.00       1710.933    10.125     8.449          1727.718    10.200     8.470          1529.685     9.312     8.213
   40.00       1418.977     8.808     8.055          1596.775     9.615     8.304          1611.924     9.683     8.324
   50.00       1127.654     7.446     7.572          1349.300     8.487     7.949          1497.784     9.168     8.169
   60.00        836.238     6.014     6.953          1073.936     7.189     7.470          1280.156     8.166     7.838
```

APPENDIX K

COMPUTER CPU TIME AND COST

The numerical computation associated with each of the flow models; EKFEM, WIKFEM, WIDFEM, and WICFEM respectively is a direct function of computer CPU time and cost. Iterative solution algorithm with a prolonged convergence will translate into enormous computer CPU time and cost.

Comparisons of models and their corresponding CPU time and cost are presented below for a given time weighting factor, time step, channel geometric, and hydraulic data. Simplified models are expected to have less CPU time and cost as compared to the complete flow model.

Models Versus CPU Time and Cost

a) Idealized Channel

Time Weighting Factor, θ = 0.55		
MODELS	Time step, Δt	
	60 seconds	300 seconds

WIKFEM	CPU^{+} = 3.59
	COST$^{\mp}$ = 0.91
		CPU = 1.44
		COST = 0.40

WIDFEM	CPU = 9.15
	COST = 2.23
		CPU = 2.67
		COST = 0.69

WICFEM	CPU = 10.68
	COST = 2.60
		CPU = 4.54
		COST = 1.14

EKFEM	ΔT = 2 seconds
	CPU = 20.87
	COST = 5.01	

$^{+}$ Unit of CPU time is seconds
$^{\mp}$ Unit of cost is dollars

b) Natural Channel

Time Weighting Factor, θ = 0.55		
	Time step, Δt	
MODELS	900 seconds	1800 seconds

WIDFEM	CPU = 72.90
	COST = 17.35
		CPU = 52.58
		COST = 12.53

WICFEM	CPU = 86.50
	COST = 20.57
		CPU = 66.94
		COST = 15.94

VITA

Ifeanyi Lawrence Nwaogazie

Candidate for the Degree of

Doctor of Philosophy

Thesis: FINITE ELEMENT MODELING OF STREAMFLOW ROUTING

Major Field: Civil Engineering

Biographical:

Personal Data: Born in Mbaise, Imo State, Nigeria, July 27, 1954, the son of Mr. and Mrs. Emmanuel Nwaogazie Okwuonu.

Education: Graduated from Ife Secondary School, Mbaise, Nigeria, in May 1974; attended Alvan Ikoku College of Education, Owerri, Nigeria with a major in mathematics/physics from August, 1974, to May, 1976; received the Bachelor of Science in Civil Engineering degree from the University of Kansas, Lawrence, Kansas, in January, 1979; received the Master of Science degree in Water Resources Engineering from the University of Kansas, Lawrence, Kansas, in June, 1980; completed the requirements for the Doctor of Philosophy degree in Civil Engineering at Oklahoma State University, Stillwater, Oklahoma, in July, 1982.

Professional Experience: Graduate research assistant, Kansas Geologic Survey, University of Kansas, April, 1979, to May, 1980; graduate research assistant, Water Resources Institute, University of Kansas, June, 1980, to July, 1980; graduate research/teaching assistant, Civil Engineering School, Oklahoma State University, August, 1980, to present.

Membership in Professional Societies: American Water Works Association, American Geophysical Union, National Water Well Association, Oklahoma Society of Professional Engineers.

Membership in Honorary Societies: Chi Epsilon, Tau Beta Pi.