UNSTEADY FLOW IN OPEN CHANNEL NETWORKS

by J. J. R. WILLIAMS

Thesis submitted to the Faculty of Engineering of the University of Aston in Birmingham for the Degree of Doctor of Philosophy.

The University of Aston in Birmingham September 1974

SUMMARY

Various finite difference systems were developed for the solution of the unsteady flow equations, applicable to fluid flow in open channel networks. Computer programs were written for these systems and comparisons were made between the results obtained from these programs and also with recorded data. Comparisons taken into account were accuracy, computer running time and complexity with regard to programming. Results were collected for increasing time steps of the finite difference grid and the time steps at which the systems either became unstable or did not reach a solution were recorded.

Techniques of schematisation and data presentation were developed, so that a general network might be considered. Each program has identical structures with blocks that deal with these networks from the point of view of cross referencing, input and output facilities etc., the only difference being in the computational blocks.

The programs were run with four unsteady flow models; the first was a purely hypothetical model which was used to eliminate programming errors; the second an unbranched section of a tidal river; the third, a network of part of a river delta and the fourth model was a small laboratory rig from which three tests were analysed. The instrumentation of this latter model provided many problems which required special attention.

Each finite difference scheme was analysed for its convergence and stability properties and expressions found for these criteria. Comparisons were then made between analytical and numerical results. The relevant unsteady formulae applicable to open channel networks were developed and a historical review of their solution is presented. Also given is a literature survey of previously developed finite difference methods.

PREFACE

The author wishes to acknowledge the guidance and constructive criticism given to him by his Supervisor, Dr. T. R. E. Chidley of the Department of Civil Engineering, the University of Aston in Birmingham. He would like to thank Dr. Chidley for his enthusiastic interest, encouragement, and for the time which he has so willingly given up.

The author would also like to express gratitude to the University of Aston who provided financial support throughout the period spent on the research.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The use of mathematical models for the prediction of water levels and discharges in rivers has become a widely used technique. Providing the river has been modelled correctly, i.e., with regard to frictional resistance, survey data, initial and boundary conditions then time varying flow changes may be determined with reasonable accuracy. There are several ways in which the basic equations may be converted into a form suitable for numerical computation on a digital computer. The most attractive method being finite differences in which solutions are determined at specified positions along the river and at predetermined instances in time.

Although a great deal of work has already been done on the solution of the unsteady flow equations by finite differences for single open channels, there appears to have been very little done on the schematisation and development of methods to deal with open channel networks. The basic objectives of the author were then two-fold; the first of which was to develop a suitable method of schematisation of a general network system, which would allow for a large amount of flexibility in referencing it and in presenting the data to the computer. The second objective was to develop and then compare a set of finite difference systems which were capable of solving the equations for the network. Methods of solution were sought that would take advantage of the equation's sparseness, both from the storage point of view and from the speed of solution.

A method of schematisation was aimed at that would allow random reference numbering of the network and the feeding of data into the computer in an arbitrary manner. Such a method would permit the subsequent insertion of extra nodes into positions where greater detail was required. The schematisation adopted was one in which discharge and depths of flow were determined at alternate positions throughout the network. These positions, or nodes, are then connected by channels that can have unequal lengths. Cross-sectional geometry was determined at the discharge nodes only.

There are two basic ways of expressing differential equations in finite difference form and these are commonly known as explicit and implicit methods, both of which are used in the author's programs. The latter method, when applied to the unsteady flow equations leads, in general to a set of nonlinear simultaneous equations and various methods of solution of these were considered. The alternatives available in solving such a system may be broken into two types; the first of which is to linearise the nonlinear equations by keeping constant, co-efficients that do not vary much and then to solve the resulting system using sparse linear algebra techniques. The co-efficients may, if necessary, be updated on an iterative basis. The second alternative is to keep the equations nonlinear and to use techniques that are specifically suited to solving such systems. Various methods based on these alternatives are used by the author and it is believed that this is the first time in which the nonlinear techniques involved have been used in the solution of civil engineering problems. When a single channel is considered "banded" structures are produced upon which fast methods of solution can be used. Numerical procedures were sought that could take advantage of this when a network was not being considered.

Investigations into the behaviour of the different finite difference schemes were done by comparing the results obtained from the running of several unsteady flow models. In order that the programs could be compared with realistic information, then tidal models of a single section of the River Aire in Yorkshire and a network consisting of part of the Ganges Delta were programmed. Comparisons were made by noting the accuracy and running times of the various numerical systems for different time steps. Also, observations were made of the time steps at which the systems either became unstable or did not reach a solution. This provided useful information to be compared with analytical predictions of instability and convergence. As a certain degree of "fitting of results" is done when dealing with river data and also that errors that occur may be attributed to irregularity and schematisation of the cross-sectional geometry, a laboratory network was built and tested. This model consisted of a network of rectangular channels with a constant surface roughness. Unsteady flow situations were propagated in the model for the sole purpose of comparing the accuracy of the finite difference programs.

The question of stability and convergence of finite difference systems resulting from quasi-linear partial differential equations is an important but complicated and involved subject. Techniques which have been suggested by previous workers for the solution of this problem with respect to the unsteady flow equations have not been complete and usually try to deal with only one aspect, i.e., stability or convergence. The author has tried to clarify the position and shown in the text is a procedure for evaluating both the stability and convergence of finite difference equations solving a system of hyperbolic equations. These techniques are applied to the finite difference schemes developed and the results show a high degree of success.

The derivation of the basic partial differential equations describing unsteady flow in open channels are discussed in Chapter 2. These are derived from considerations of conservation of mass, momentum and energy. Slight differences occur between the dynamic equation derived from momentum and energy principles and these are discussed. Also presented in this chapter is a historical review of methods that have been developed for the solution of these equations.

The literature survey in Chapter 3 deals in detail with the finite difference systems that have been suggested and used by previous workers in this field. Most are concerned with single channels, with only a few references to networks. It is useful to review these main methods as they give a clear insight to the problem. Also, they provide a basis upon which extensions could have been made to handle networks. The survey starts by considering two papers, which are themselves reviews of work prior to their publication, written by well known authors who are acknowledged experts on this subject.

Further work is then dealt with.

Chapter 4 gives in detail the finite difference systems developed by the author, together with the various algebraic techniques used to solve these systems. The analytical stability and convergence criteria of these are then derived in Chapter 5.

The laboratory work itself posed many problems regarding instrumentation. Because of the scale of the model unsteady flow situations had to be done quickly and thus required devices that could monitor these changes accurately on a time varying basis. Chapter 6 describes the laboratory apparatus and techniques adapted, together with the unsteady flow tests recorded for numerical computation.

The two physical models of the River Aire and part of the Ganges Delta mentioned previously, together with a hypothetical model which was used in the early stages of the research for program checking, are then described in Chapter 7. Also given is the geometry and schematisation of the layout of each model, together with the boundary conditions for the unsteady flow tests. In both of the physical models some data was missing, either initial conditions or channel friction, or both. The techniques used to determine these are described.

Chapter 8 gives the results of the unsteady flow tests and discusses these with respect to accuracy, computer running time and general behaviour. Chapter 9 draws conclusions from the results of the research as a whole, together with recommendations for further work.

CHAPTER TWO

BASIC THEORY

2.0 Introduction

This section reviews the derivation of the equations of unsteady flow in an open channel. The dynamic equation is derived from both energy and momentum principles and the continuity equation from mass conservation.

Methods of solution of the equations are discussed, one method in particular leads to a classification of the equations.

2.1 Continuity Equation

Consider a section of channel shown in Fig. 2.1. Flow is between points (1) and (2) distance Δx apart. The depth shown in full is at time t and dotted at time $t + \Delta t$. A is the area of section 1 and Q is the flow entering section 1. q is the lateral inflow per unit length, per unit time, which for the purpose of this analysis is assumed constant over the time interval Δt. The lateral inflow may take the form of rainfall runoff or water from a pumping station, reservoir or similar structure. When $q>o$ then water is entering the main stream and when $q<o$ then water is taken from the main stream. Flow from a tributary that is to be treated as independent of the main stream calculations may also be introduced as a q value, in which case the flow is distributed over a particular reach.

We now observe what happens at times t and $t + \Delta t$,

Time t :

Flow entering section 1 = Q, with area A.

flow leaving section 2 is $Q + \frac{\partial Q}{\partial x}\Delta x$ and the area

is $$A + \frac{\partial A}{\partial x}\Delta x$$

Time $t + \Delta t$:

flow entering section 1 is $Q + \frac{\partial Q}{\partial t}\Delta t$

with area $$A + \frac{\partial A}{\partial t}\Delta t$$

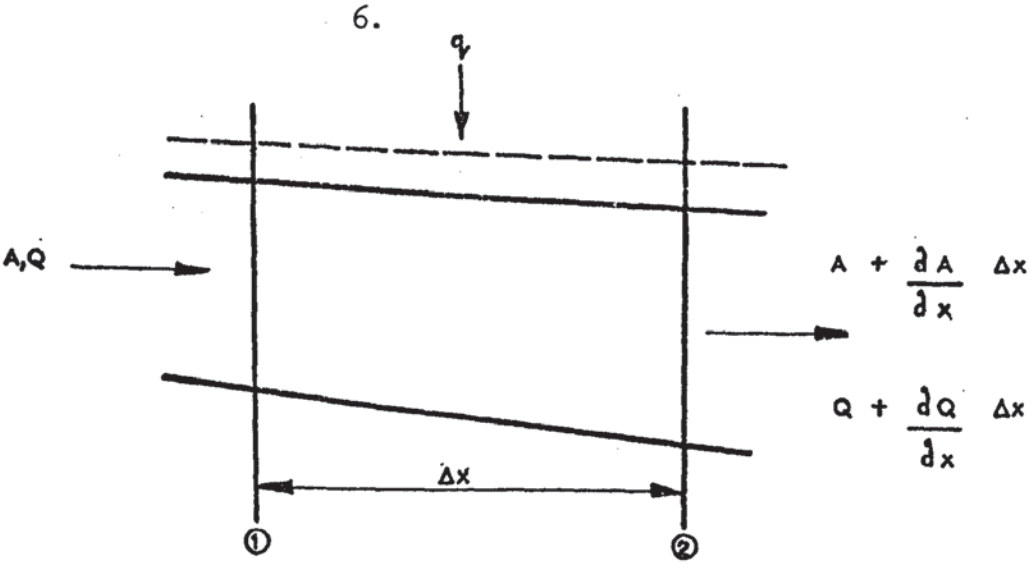

q
A,Q
$A + \frac{\partial A}{\partial x} \Delta x$
$Q + \frac{\partial Q}{\partial x} \Delta x$
Δx
①
②

FIGURE 2.1

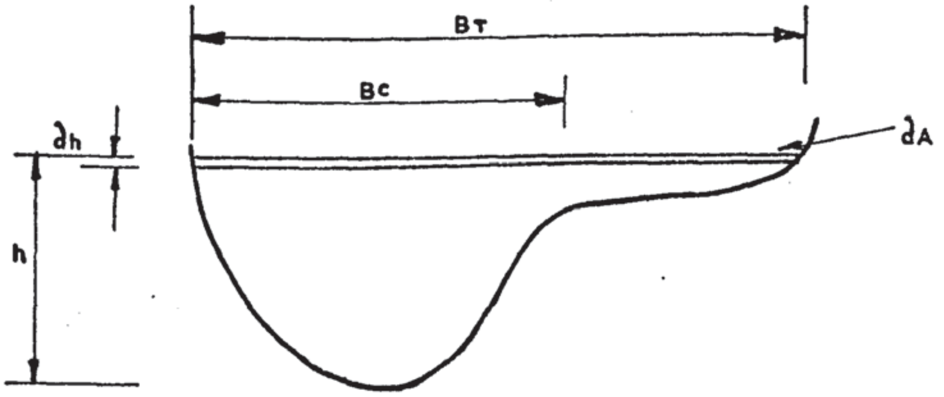


FIGURE 2.2

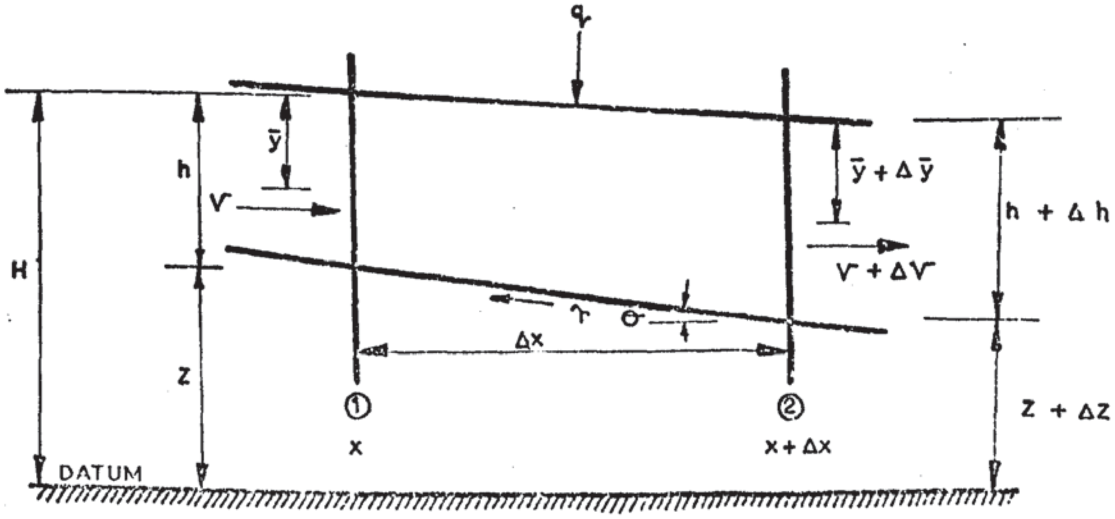


FIGURE 2.3

flow leaving section 2 is $\left(Q + \frac{\partial Q}{\partial x}\Delta x\right) + \frac{\partial}{\partial t}\left(Q + \frac{\partial Q}{\partial x}\Delta x\right)\Delta t$

where the area is now $\left(A + \frac{\partial A}{\partial x}\Delta x\right) + \frac{\partial}{\partial t}\left(A + \frac{\partial A}{\partial x}\Delta x\right)\Delta t$,

It can now be stated that the average net amount of fluid entering the element over the time interval Δt is equivalent to the average amount of increase in storage in that element, thus on oombining the above and neglecting second order terms the following is produced :

$$\left[\left(A + \frac{\partial A}{\partial t}\Delta t\right) + \left(A + \frac{\partial A}{\partial x}\Delta x + \frac{\partial A}{\partial t}\Delta t\right)\right]\frac{\Delta x}{2} - \left[A + \left(A + \frac{\partial A}{\partial x}\Delta x\right)\right]\frac{\Delta x}{2}$$

$$= \left[Q + \left(Q + \frac{\partial Q}{\partial t}\Delta t\right)\right]\frac{\Delta t}{2} - \left[\left(Q + \frac{\partial Q}{\partial x}\Delta x\right) + \left(Q + \frac{\partial Q}{\partial x}\Delta x + \frac{\partial Q}{\partial t}\Delta t\right)\right]\frac{\Delta t}{2}$$

$$+ q\,\Delta x\,\Delta t$$

giving $$\frac{\partial A_T}{\partial t} + \frac{\partial Q}{\partial x} = q \qquad (2.1.1)$$

which is the standard partial differential equation of mass continuity in unsteady flow. It will be noticed that A is replaced by A_T indicating that this term represents the total area, i.e.,including flood plain.

In equation (2.1.1) it is noted that $\partial A = B_T\,\partial h$ where B_T, as shown in Fig. 2.2, is the total width, i.e., including flood plain. h is the depth of water.

Eq. (2.1.1) then becomes $$B_T\frac{\partial h}{\partial t} + \frac{\partial Q}{\partial x} = q \qquad (2.1.2)$$

2.2 Derivation of Dynamic Equation from Momentum Principles

At section 1 the parameters are :

A = convective area of cross section

P = wetted perimeter of convective area

h = depth of water

z = height above datum

$\bar{y}$ = distance to centroid of convective area

V = average velocity

θ = angle of bed slope

x = arbitrary distance

At section 2 the parameters change to :

$A + \Delta A$, $P + \Delta P$, $h + \Delta h$, $z + \Delta z$, $\bar{y} + \Delta\bar{y}$, $V + \Delta V$ and $x + \Delta x$.

The above diagram and parameters together with the following assumptions are applicable to this section and the following section based on energy principles.

Assumptions :

1. The flow is so gradually varied that vertical accelerations are neglected and hence the pressure distribution is hydrostatic.

2. Slopes are sufficiently small such that $\sin\theta = \tan\theta = -\Delta z/\Delta x$ and that hydrostatic forces act in the direction of flow.

3. The velocity distribution in the channel may be represented by the average velocity and that friction losses are proportional to the velocity squared, i.e., as in steady flow.

4. The lateral inflow enters the main stream normally and so has no velocity component in the direction of motion.

5. The transfer of energy and momentum to and from the flood plain is neglected.

The general equation of momentum is now applied to flow between the sections 1 and 2 to obtain

$$\rho QV + \gamma A\bar{y} - \gamma\frac{(2A + \Delta A)}{2}\frac{\Delta z}{\Delta x}\Delta x = \rho(Q + \Delta Q)(V + \Delta V) +$$

$$\gamma(A + \Delta A)(\bar{y} + \Delta\bar{y}) + To\ \Delta x\ \frac{(2P + \Delta P)}{2} \tag{2.2.1}$$

the last term on the left of the equation is the component of the weight of the water acting downstream.

To is the mean longitudinal shear stress acting over the convective wetted perimeter.

γ is the specific weight of the fluid and ρ its density.

If second order terms are ignored in eq. (2.2.1) then it reduces to :

$$0 = \rho Q\Delta V + \rho\Delta QV + \gamma A\Delta\bar{y} + \gamma\Delta A\bar{y} + To\ \Delta x\ P + \gamma A\frac{\Delta z}{\Delta x}\Delta x \tag{2.2.2}$$

now $\Delta Q = q\ \Delta x$ (2.2.3)

and from the following analysis it can be shown that

$$\gamma A \Delta \bar{y} + \gamma \Delta A \bar{y} = \gamma A \Delta h \qquad (2.2.4)$$

When h is increased by Δh the first moment of area about the new surface becomes : (see Fig. 2.4)

$$A\ (\bar{y} + \Delta h) + B_c\ \Delta h\ (\frac{\Delta h}{2})$$

and so $(A + \Delta A)(\bar{y} + \Delta \bar{y}) = A(\bar{y} + \Delta h) + B_c \frac{\Delta h^2}{2}$

When second order terms are ignored this equation becomes :

$$A\ \Delta \bar{y} + \Delta A\ \bar{y} = A\ \Delta h$$

The equations (2.2.3) and (2.2.4) are substituted into equation (2.2.2) the result is divided by A, γ and Δx which when taken to the limit gives :

$$0 = \frac{V}{g}\frac{dV}{dx} + \frac{Vq}{Ag} + \frac{dh}{dx} + \frac{dz}{dx} + \frac{T_o\ P}{\gamma\ A} \qquad (2.2.5)$$

The hydraulic radius R is defined as A/P so the friction term $\frac{T_o\ P}{\gamma\ A}$ is replaced by $\frac{V|V|}{C^2 R}$, where C is Chezy's C.

The modulus term is introduced so that the friction still apposes flow when the velocity is in the reverse direction.

If the total differential terms are now examined in more detail it can be shown from the theory of partial differentiation that :

$$\frac{V}{g}\frac{dV}{dx} = \frac{V}{g}\frac{\partial V}{\partial x} + \frac{1}{g}\frac{\partial V}{\partial t} \qquad (2.2.6)$$

$$\frac{dh}{dx} = \frac{\partial h}{\partial x} + \frac{1}{V}\frac{\partial h}{\partial t} \qquad (2.2.7)$$

$$\frac{dz}{dx} = \frac{\partial z}{\partial x} + \frac{1}{V}\frac{\partial z}{\partial t} \qquad (2.2.8)$$

As vertical accelerations are ignored in the dynamic equation $\frac{dh}{dx}$ becomes $\frac{\partial h}{\partial x}$ and as the bed slope is assumed not to vary over the time concerned $\frac{dz}{dx}$ becomes $\frac{\partial z}{\partial x}$.

The dynamic equation (2.2.5) then reduces to

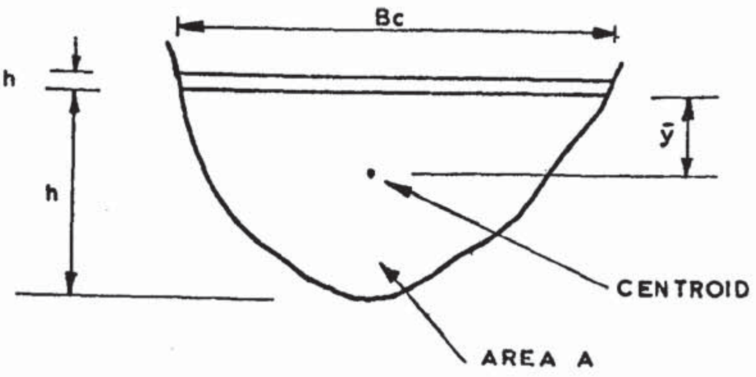


FIGURE 2.4

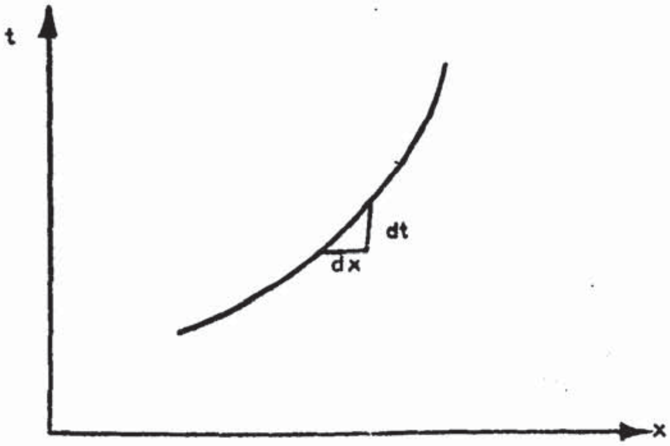


FIGURE 2.5

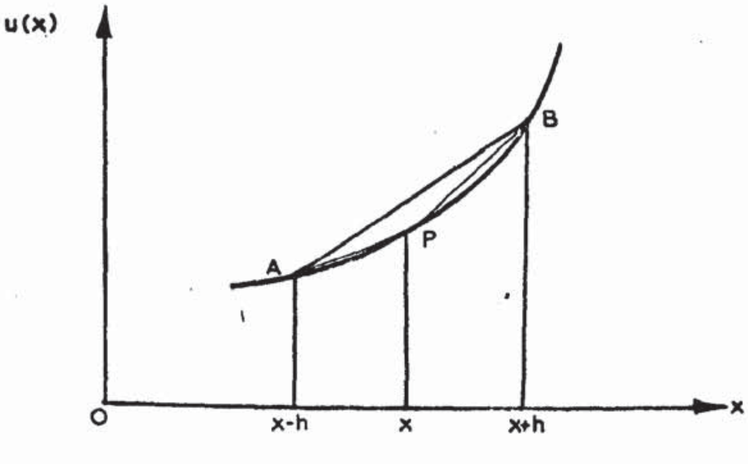


FIGURE 2.6

$$0 \doteq \frac{1}{g}\frac{\partial V}{\partial t} + \frac{V}{g}\frac{\partial V}{\partial x} + \frac{\partial h}{\partial x} + \frac{\partial z}{\partial x} + \frac{V|V|}{C^2R} + \frac{Vq}{Ag} \qquad (2.2.9)$$

i.e., the usual partial differential dynamic equation for unsteady flow.

2.3 Derivation of Dynamic Equation from Energy Principles

Reference is made to Fig. 2.3 and to the assumptions made prior to the derivation of the dynamic equation in the previous section.

The general energy equation is now applied to either side of the element to give the following :

$$\gamma Q\left(\frac{V^2}{2g} + z + h\right) + q\gamma\Delta x(z+h) = \gamma(Q+q\Delta x)\left[\frac{(V+\Delta V)^2}{2g} + (z+\Delta z) + (h+\Delta h)\right]$$
$$+ \mathrm{To}\,\left(\frac{2P+\Delta P}{2}\right)\frac{\Delta x^2}{\Delta t} \qquad (2.3.1)$$

where $\mathrm{To}.\left(\frac{2P+\Delta P}{2}\right)\frac{\Delta x^2}{\Delta t}$ is the rate of doing work in overcoming the friction around the perimeter.

The lateral inflow is assumed to have potential energy before entering the main stream but no kinetic energy, afterwards it has both.

Eq. (2.3.1) after ignoring second order terms becomes :

$$0 = \gamma Q\left(\frac{V}{g}\Delta V + \Delta z + \Delta h\right) + \gamma q\,\Delta x\,\frac{V^2}{2g} + \mathrm{To}\;P\,\frac{\Delta x^2}{\Delta t} \qquad (2.3.2)$$

which after dividing by γ, Q and Δx and taking to the limit gives :

$$0 = \frac{V}{g}\frac{dV}{dx} + \frac{dz}{dx} + \frac{dh}{dx} + \frac{qV}{2Ag} + \frac{\mathrm{To}\;P}{\gamma\;A} \qquad (2.3.3)$$

when the arguments concerning the replacement of the total differentials with partial differentials in the previous section are used again, together with the substitution of $\frac{\mathrm{To}\;P}{\gamma\;A}$ for $\frac{V|V|}{C^2R}$ then eq. (2.3.3) becomes :

$$0 = \frac{1}{g}\frac{\partial V}{\partial t} + \frac{V}{g}\frac{\partial V}{\partial x} + \frac{\partial z}{\partial x} + \frac{\partial h}{\partial x} + \frac{Vq}{2Ag} + \frac{V|V|}{C^2R} \qquad (2.3.4)$$

In eq. (2.3.4) and (2.2.9) $\frac{\partial z}{\partial x} + \frac{\partial h}{\partial x}$ may be replaced by $\frac{\partial H}{\partial x}$ where H, as defined in Fig. 2.3, = z+h.

2.4 Comparison of the Two Dynamic Equations

When equations (2.2.9) and (2.3.4) are compared it can be seen that all the terms correspond apart from the term involving the lateral inflow where there is a difference of a factor of one half. Yen and Wenzel [22], who have done a detailed study of the differences of the two approaches for a steady spatially varied flow, concluded that although both principles were derived from Newtons second law the two methods are inherently different, the momentum approach is a vector relationship and the energy approach a scalar one.

In the case of the lateral inflow, the momentum equation, according to Henderson [9] is to be prefered, as although it does not take into account the potential energy of the lateral inflow before entering the main stream it is considered that energy losses will occur when the arriving flow mixes with the flow already in the channel. However as the effect of the q term in the dynamic equation is small and could be neglected altogether it is sufficient to use the form as in the momentum equation.

If co-efficients had been introduced in the derivation of the equation which took into account the non-uniform velocity distribution then further differences would be evident. As these co-efficients are only slightly greater than unity it is usual in mathematical models to consider them just as unity.

2.5 Variations of the Basic Equations

For the reasons stated above it is decided that the dynamic equation derived from momentum principles eq. (2.2.9) is the appropriate one to use, this together with the continuity eq. (2.1.1) gives two quasi-linear hyperbolic equations in the two unknowns V and H.

The dependent variable V can, however, be expressed in terms of Q and the convective area, Ac, in the following manner :

Q = V Ac, which on differentiating gives ;

$$\frac{\partial Q}{\partial t} = A\,\frac{\partial V}{\partial t} + V\,\frac{\partial Ac}{\partial t} \qquad (2.5.1)$$

and $$\frac{\partial Q}{\partial x} = A\,\frac{\partial V}{\partial x} + V\,\frac{\partial Ac}{\partial\, x} \qquad (2.5.2)$$

After substituting for $\frac{1}{g}\frac{\partial V}{\partial t}$ and $\frac{V}{g}\frac{\partial V}{\partial x}$ in eq. (2.2.9) and replacing V for Q/Ac the following can be obtained :

$$0 = \frac{1}{Acg}\frac{\theta Q}{\partial t} - \frac{Q}{Ac^2g}\frac{\partial A}{\partial t} + \frac{Q}{Ac^2g}\frac{\partial Q}{\partial x} - \frac{Q^2}{Ac^3g}\frac{\partial Ac}{\partial x} + \frac{\partial h}{\partial x} + \frac{\partial z}{\partial x} + \frac{Q^2}{C^2Ac^2R} + \frac{Qq}{Ac^2g} \qquad (2.5.3)$$

It will be noticed that all areas in eq. (2.5.3) are replaced Ac.

In the second term above the partial differential $\frac{\partial A}{\partial t}$ may be replaced by Bc $\frac{\partial h}{\partial t}$ where Bc is the width of the convective channel, the third term involving $\frac{\partial Q}{\partial x}$ may be replaced by $q - \frac{\partial A}{\partial t}$ from eq. (2.1.1). In this case, however $\frac{\partial A}{\partial t}$ is $B_T \frac{\partial h}{\partial t}$ where B_T is the total width, i.e., including the flood plain.

Eq. (2.5.3) then becomes :

$$0 = \frac{1}{Acg}\frac{\partial Q}{\partial t} - \frac{Q}{Ac^2g}(Bc+B_T)\frac{\partial h}{\partial t} - \frac{Q^2}{Ac^3g}\frac{\partial A}{\partial x} + \frac{\partial H}{\partial x} + \frac{Q|Q|}{C^2Ac^2R} + \frac{2Qq}{Ac^2g} \qquad (2.5.4)$$

2.6 Solution of the Unsteady Flow Equations

The partial differential equations as such are far too complex for any analytical solution. Even if the channels were so idealised, i.e., made into equivalent rectangles, and an analytical solution found, then the solution itself would still be a great deal more complex than numerical solutions of the basic equations.

Before the advent of the modern digital computer solutions were obtained by grossly simplifying the unsteady flow equations and then solving the resulting equations graphically. Usually these methods revolved around evaluating some form of the continuity equation (usually written as $\Delta S/\Delta t = I - O$, in which S = storage, I = inflow, O = outflow). One such variation on this is the Muskingum Method. Another variation is the graphical solution of the Method of Characteristics by Schönfeld, the theory of which will be dealt with in more detail later.

Another well used method is the harmonic method, in which the wave motion whether periodic or not, is considered to be composed of harmonic wave-components. The basic equations are simplified by neglecting terms, usually the convective ones, and then linearised. This means that a linear friction law is used and that in some instances the friction term has been disregarded completely.

Three of the later, more popular, methods of solution of the unsteady flow equations in which there is no need to neglect terms are now described.

2.7 Power Series

This method according to Baltzer and Lai [4] was the first method to be introduced, on the arrival of modern computers, for the solution of the equations.

The basis of the method is that a function at a point x_2, say, may be expressed in terms of the function at a near-by point x_1, by a Taylors Series,

$$\text{i.e.,}\quad f(x_2) = f(x_1) + \sum_{n=1}^{\infty} \frac{f^n(x_1)}{n!} (x_2 - x_1)^n \tag{2.7.1}$$

By selecting the reference point $x_1 = 0$ and $x_2 - x_1 = x$ the above becomes a MacLaurins Series

$$\text{i.e.,}\quad f(x) = f(o) + \sum_{n=1}^{\infty} \frac{f^n(o)}{n!} x^n \tag{2.7.2}$$

If H and Q are chosen to be the dependent variables in the unsteady flow equations then from eq. (2.7.2)

$$H_2 = H_1 + x \frac{\partial H_1}{\partial x} + \frac{x^2}{2!} \frac{\partial^2 H_1}{\partial x^2} + \frac{x^3}{3!} \frac{\partial^3 H_1}{\partial x^3} + \text{------} \tag{2.7.3}$$

and $$Q_2 = Q_1 + x \frac{\partial Q_1}{\partial x} + \frac{x^2}{2!} \frac{\partial^2 Q_1}{\partial x^2} + \frac{x^3}{3!} \frac{\partial^3 Q_1}{\partial x^3} + \text{------} \tag{2.7.4}$$

In eq. (2.5.3) $q - \frac{\partial Q}{\partial x}$ is substituted for $\frac{\partial A}{\partial t}$ to form the following :

$$0 = \frac{1}{Ag} \frac{\partial Q}{\partial t} + \frac{2\ Q}{A^2 g} \frac{\partial Q}{\partial x} - \frac{Q^2}{A^3 g} \frac{\partial A}{\partial x} + \frac{\partial H}{\partial x} + \frac{Q|Q|}{C^2 A^2 R} \tag{2.7.5}$$

which after rearranging, and substituting $|Q|/Q$ for λ and C^2A^2R for K gives

$$\frac{\partial H}{\partial x} = -\frac{1}{Ag} \frac{\partial Q}{\partial t} - \frac{2Q}{A^2 g} \frac{\partial Q}{\partial x} + \frac{Q^2}{A^3 g} \frac{\partial A}{\partial x} - \lambda \frac{(Q)^2}{K} \tag{2.7.6}$$

If eq. (2.7.6) is now differentiated with respect to x and the result, together with eq. (2.7.6) itself, are then substituted into eq. (2.7.3) the following is formed after neglecting third and higher order terms;

$$H_2 = H_1 - \lambda \frac{Q^2 x}{K^2} - \frac{x}{Ag}\frac{\partial Q}{\partial t} - \frac{2Qx}{A^2 g}\frac{\partial Q}{\partial x} + \frac{Q^2 x}{A^3 g}\frac{\partial A}{\partial x} - \lambda \frac{Qx^2}{K^2}\frac{\partial Q}{\partial x}$$

$$+ \lambda \frac{Q^2 x^2}{K^3}\frac{\partial K}{\partial x} - \frac{x^2}{2Ag}\frac{\partial}{\partial t}\left(\frac{\partial Q}{\partial x}\right) - \frac{Qx^2}{A^2 g}\frac{\partial}{\partial x}\left(\frac{\partial Q}{\partial x}\right) + \frac{Q^2 x^2}{2gA^3}\frac{\partial}{\partial x}\left(\frac{\partial A}{\partial x}\right)$$

$$- \frac{\lambda x^3}{3\,K^2}\left(\frac{\partial Q}{\partial x}\right)^2 - \frac{\lambda\,Qx^3}{3\,K^2}\frac{\partial}{\partial x}\left(\frac{\partial Q}{\partial x}\right) \qquad (2.7.7)$$

After rearranging this equation and simplifying by using the continuity equation, eq. (2.1.2), the following is obtained :

$$H_2 = H_1 - \lambda \frac{Q^2 x}{K} + \lambda \frac{Qx^2}{K}\left(B\frac{\partial H_1}{\partial t} - q\right) + \frac{B\ \ x^2}{2Ag}\frac{\partial^2 H_1}{\partial t^2} - \frac{x}{Ag}\frac{\partial Q}{\partial t}\left(1 + \lambda \frac{B\ x^2}{3k^2}\frac{\partial Q}{\partial t}\right.$$

$$\left. + \frac{2}{3}\frac{Bx^2\ Q^2\ Ag}{K^4}\right) - \lambda \frac{B^2 x^3}{3K^2}\left(\frac{\partial H_1}{\partial t}\right)^2 \qquad (2.7.8)$$

Again a similar expansion in terms of Q_2 and Q_1 may be found by using the continuity equation in which the first derivative is :

$$\frac{\partial Q}{\partial x} = q - B\frac{\partial H}{\partial t} \qquad (2.7.9)$$

(the subscript in B_T is dropped for simplicity), so $\frac{\partial^2 Q}{\partial x^2} = B\frac{\partial}{\partial x}\left(\frac{\partial H}{\partial t}\right) = \frac{\partial}{\partial t}\left(\frac{\partial H}{\partial x}\right)$,

and by differentiating eq. (2.7.6) with respect to t the following is formed

$$Q_2 = Q_1 - x\,B\frac{\partial H}{\partial t} + x\,q - \lambda Q\frac{B\ x^2}{K^2}\frac{\partial Q}{\partial t} + \frac{\lambda Q^2 B\ x^2}{K^3}\frac{\partial K}{\partial t} - \frac{B\ x^2}{2Ag}\frac{\partial^2 Q}{\partial t^2}$$

$$- \frac{Q\ B\ x^2}{g\ A^2}\frac{\partial}{\partial x}\left(\frac{\partial Q}{\partial t}\right) \qquad (2.7.10)$$

When equation (2.7.7) is written in finite difference from for $\frac{\partial Q}{\partial t}$ in terms of the other expressions, then

$$\Delta Q_{(1,t)} = \left[g\frac{A_t(H_1 - H_2)_t}{x} - \lambda g\frac{A_t\ Q^2_{(1,t)}}{K^2{}_t} + \lambda g\frac{A_t\ Q_{(1,t)}}{K^2{}_t}\,x\left(B_t\frac{\Delta H}{\Delta t}(1,t) - q\right)\right.$$

$$\left. + B_t\frac{x}{2}\left(\frac{H_{(1,t+\Delta t)} - 2\,H_{(1,t)} + H_{(1,t-\Delta t)}}{\Delta t^2}\right) - \lambda g\frac{A_t\ (B_t)^2\ x^2}{3K_t}\left(\frac{\Delta H_{(1,t)}}{\Delta t}^2\right)\right]$$

$$\times \frac{\Delta t}{\left[1 + \frac{2}{3}\left(\frac{B_t\ x^2 Q^2{}_{(1,t)}}{K^4{}_t}g\ A_t\right) + \lambda\frac{B_t\ x^2}{3\ K_t^2}\frac{\Delta Q_{(1,t)}}{\Delta t}\right]} \qquad (2.7.11)$$

where $\frac{\Delta H}{\Delta t}(1,t) = \frac{H_{(1,t+\Delta t)} - H_{(1,t-\Delta t)}}{2\ \Delta t}$ (2.7.12)

Eq. (2.7.11) expresses the increase in flow at point 1 uniquely in terms of $\frac{\partial H}{\partial t}$ and so $Q_{(1,t+\Delta t)}$ is found, i.e., $Q_{(1,t)} + \Delta Q_1$.

By reversing the subscript notations the increase in flow at point 2 may also be found providing the values of $\frac{\Delta H}{\Delta t}$ can be estimated there.

Thus if all the initial values at time t are known and two boundary conditions are also given it is possible to determine all the Q's and H's throughout the reach by successive applications of eq. (2.7.11) and the equivalent form of eq. (2.7.10) on an iterative basis.

By using equation (2.7.11) and the equivalent to eq. (2.7.10) it is possible to use this procedure for a network of channels, although it is much more complicated than a method which simply replaces the derivatives by their equivalent finite differences.

2.8 The Method of Characteristics

The continuity equation (2.1.1) may be expanded in the following manner:

$$\frac{\partial Q}{\partial x} = A\frac{\partial V}{\partial x} + V\frac{\partial A}{\partial x} \tag{2.8.1}$$

now $$\frac{\partial A}{\partial x} = \frac{\partial A}{\partial h}\frac{\partial h}{\partial x} = B\frac{\partial h}{\partial x} = B\frac{\partial (H-z)}{\partial x} \tag{2.8.2}$$

and $-\frac{\partial z}{\partial x} = S_o$, the bed slope, so (2.8.2) becomes :

$$\frac{\partial A}{\partial x} = B\frac{\partial H}{\partial x} + B\ S_o \tag{2.8.3}$$

To simplify matters for the purpose of the following derivations, the channel is, again, considered to have no flood plain, and so

$$\frac{\partial A}{\partial t} = B\frac{\partial H}{\partial t} \tag{2.8.4}$$

the continuity eq. (2.1.1) then becomes :

$$A\frac{\partial V}{\partial x} + V\ B\frac{\partial H}{\partial x} + V\ B\ S_o + B\frac{\partial H}{\partial t} = q \tag{2.8.5}$$

which on dividing by B and rearranging gives :

$$\frac{A}{B}\frac{\partial V}{\partial x} + V\frac{\partial H}{\partial x} + \frac{\partial H}{\partial t} = \frac{q}{B} - V\ S_o \tag{2.8.6}$$

The dynamic eq. (2.2.9) may be rearranged to give :

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} + g\frac{\partial H}{\partial x} = -\left(\frac{gV|V|}{C^2R} + \frac{Vq}{A}\right) \qquad (2.8.7)$$

It can be shown that at every point in the solution domain of a set of partial differential equations, there are two directions along which the integration of the partial differential equations reduces to the integration of equations involving total differentials only, i.e., in these directions the equations are not affected by partial derivatives in other directions. this leads to a natural classification of partial differential equations.

We now look for a curve in the x-t plane along which are given the values of V and H and that their derivatives satisfy equations (2.8.6) and (2.8.7), see Fig. 2.5.

Along this curve the following relationships must hold :

$$dV = \frac{\partial V}{\partial x}dx + \frac{\partial V}{\partial t}dt \qquad (2.8.8)$$

and $$dH = \frac{\partial H}{\partial x}dx + \frac{\partial H}{\partial t}dt \qquad (2.8.9)$$

These two equations together with equations (2.8.6) and (2.8.7) give a set of four simultaneous equations in the four unknowns $\frac{\partial V}{\partial t}$, $\frac{\partial V}{\partial x}$, $\frac{\partial H}{\partial t}$ and $\frac{\partial H}{\partial x}$, which when expressed in matrix form, gives :

$$\begin{bmatrix} dx & dt & 0 & 0 \\ A/B & 0 & V & 1 \\ V & 1 & g & 0 \\ 0 & 0 & dx & dt \end{bmatrix} \begin{bmatrix} \partial V/\partial x \\ \partial V/\partial t \\ \partial H/\partial x \\ \partial H/\partial t \end{bmatrix} = \begin{bmatrix} dV \\ (q/B - V\,S_o) \\ -\left(\frac{gV|V|}{C^2R} + \frac{Vq}{A}\right) \\ dH \end{bmatrix} \qquad (2.8.10)$$

When considering a general set of linear equations A x = B then, by Cramers rule for determinants, any unknown x_r in the vector x may be expressed in the following manner

$x_r = \frac{|A_r|}{|A|}$, where $|A|$ is the determinant of matrix A and $|A_r|$ is the determinant of the matrix in which the rth column has been replaced by the vector B.

i.e.,
$$\frac{\partial V}{\partial x} = \frac{\begin{vmatrix} dV & dt & 0 & 0 \\ (\frac{q}{B} - V\,S_o) & 0 & V & 1 \\ -(\frac{gV|V|}{C^2R} + Vq) & 1 & g & 0 \\ dH & 0 & dx & dt \end{vmatrix}}{\begin{vmatrix} dx & dt & 0 & 0 \\ A/B & 0 & V & 1 \\ V & 1 & g & 0 \\ 0 & 0 & dx & dt \end{vmatrix}} \qquad (2.8.11)$$

and so $\frac{\partial V}{\partial t}$, $\frac{\partial H}{\partial x}$ and $\frac{\partial H}{\partial t}$ may be expressed in a similar form such that :

$$\frac{\partial V/_{\partial x}}{\Delta_1} = \frac{\partial V/_{\partial t}}{\Delta_2} = \frac{\partial H/_{\partial x}}{\Delta_3} = \frac{\partial H/_{\partial t}}{\Delta_4} = \frac{1}{\Delta_5} \qquad (2.8.12)$$

where Δ_1 is the determinant of A in which the first column has been replaced by B, Δ_2 is the determinant of A with the second column replaced by B and so on for Δ_3 and Δ_4. Δ_5 is $|A|$.

If $\Delta_5 \neq 0$ then the values of the unknown partial derivatives may be determined uniquely. If $\Delta_5 = 0$ then usually the values of the partial derivatives are infinite and the known values V and H along the curve will not satisfy the original partial differential equations. If, however $\Delta_5 = 0$ and Δ_1, Δ_2, Δ_3 and Δ_4 are also equal to zero then the partial derivatives can be finite and satisfy the eqns. (2.8.6) and (2.6.7).

It can be shown that if $\Delta_5 = 0$ then, by manipulation of Δ_5, Δ_1, Δ_2, Δ_3 and Δ_4 are also equal to zero.

Δ_5 becomes:-
$$dx^2 + V^2dt^2 - \frac{gA}{B}\,dt^2 - 2Vdxdt = 0 \qquad (2.8.13)$$

which when divided by dt^2 gives

$$\left(\frac{dx}{dt}\right)^2 - 2\,V\,\frac{dx}{dt} + \left(V^2 - \frac{gA}{B}\right) = 0 \qquad (2.8.14)$$

In the ancillary quadratic equation $Ax^2 + Bx + C = 0$ (2.8.15)

$$x_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \qquad (2.8.16)$$

and so $\frac{dx}{dt} = V \pm \left(\frac{gA}{B}\right)^{\frac{1}{2}}$. (2.8.17)

The above eq. (2.8.17) gives two curves in the x-t plane called characteristic curves. When $\frac{dx}{dt} = V + \left(\frac{gA}{B}\right)^{\frac{1}{2}}$ the forward characteristic or C+ curve is given, and when $\frac{dx}{dt} = V - \left(\frac{gA}{B}\right)^{\frac{1}{2}}$ then the backward characteristic or C- curve is given.

The velocities $\frac{dx}{dt}$ are equivalent to the celerity at which an infinitesimal gravity wave would travel in an open channel. The value $\left(\frac{gA}{B}\right)^{\frac{1}{2}}$ is usually denoted by the wave celerity c so that :

$$\frac{dx}{dt} = V \pm c \tag{2.8.18}$$

Expansion of Δ_1 gives :

$$\frac{dH}{dt} + \frac{1}{g}\frac{dV}{dt}\left(\frac{dx}{dt} - V\right) + \frac{1}{g}\left(\frac{gV|V|}{C^2R} + \frac{Vq}{A}\right)\left(\frac{dx}{dt} - V\right) - \left(\frac{q}{B} - V\,S_o\right) = 0 \tag{2.8.19}$$

which on substitution of eq. (2.8.18) for $\frac{dx}{dt}$ gives :

$$\frac{dH}{dt} \pm \frac{c}{g}\frac{dV}{dt} + \left\{VS_o \pm c\,\frac{V|V|}{C^2R} - \frac{q}{B}\left(1 \mp \frac{V}{c}\right)\right\} = 0 \tag{2.8.20}$$

So if the set of partial differential equations (2.8.6) and (2.8.7) are made to run along the characteristic curves C+ and C-, given by eq. (2.8.18) then they can be transformed into two total differential equations given by eq. (2.8.20).

Along the C+ curve :

$$dt - \frac{dx}{V+c} = 0 \tag{2.8.21}$$

$$\text{and } dH + \frac{c}{g}\,dV + F_+\,dt = 0 \tag{2.8.22}$$

and along the C- curve :

$$dt - \frac{dx}{V-c} = 0 \tag{2.8.23}$$

$$\text{and } dH - \frac{c}{g}\,dV + F_-\,dt = 0 \tag{2.8.24}$$

$$\text{in which } F\pm = V\,S_o \pm \frac{cV|V|}{C^2R} - \frac{q}{B}\left(1 \mp \frac{V}{c}\right) \tag{2.8.25}$$

The four equations (2.8.21), (2.8.22), (2.8.23), and (2.8.24) are then transformed into finite differences in which the values at the beginning of the time Δt are related to the two unknowns at the end of the interval. Δt is chosed so that $\Delta t < \frac{\Delta x}{|V \pm a|}$ (2.8.26)

There are two standard methods of solving the four above equations; The first uses a Characteristic grid in which the characteristics are allowed to be projected into the solution domain until they intersect; the solutions obtained are then interpolated to give values at specific points in space and time. The second method uses a Rectangular grid in which at the particular point a solution is sought, i.e., at time $t + \Delta t$, the two characteristics are projected back to the time t line. Information on the line at time $t = t$ is then interpolated for insertion into the four equatioms to give the desired solution at time $t + \Delta t$. Boundary conditions are required at the ends of the channels to give complete solutions over the entire reach.

Once a general partial differential equation or set of equations is reduced to a characteristic equation similar to eq. (2.8.14) then depending on the roots of this equation the original partial differential equation is classified.

Thus referring to eq. (2.8.15):

if $B^2>4AC$ then the roots are real and unequal, and the partial differential equation is then termed hyperbolic (as in unsteady flow), if $B^2=4AC$ then the roots are real and equal and the partial differential equation is then termed parabolic, if $B^2<4AC$ then the roots are complex and the partial differential equation is elliptic.

It is noted that for non-linear equations the classification may be dependent upon the solution.

The method of characteristics is considered by Liggett and Woolhiser [12] to be the most accurate of all the methods, and is the standard by which others should be judged. However, because the solutions are displaced in time and space, thus requiring interpolation procedures, other methods, which give solutions at specified points, become more attractive.

For a more general derivation of characteristic equations see the Government Publication [14].

2.9 Finite Differences

In this section a particular variable u is considered, first as a function of one independent variable, and then later as a function of more than one independent variable.

If u = f(x) then u at a point x+h may be expressed in terms of u at x by means of a Taylor Series expansion;

i.e., $$u(x+h) = u(x) + h\frac{du}{dx} + \frac{h^2}{2!}\frac{d^2u}{dx^2} + \frac{h^3}{3!}\frac{d^3u}{dx^3} + O(h^4) \qquad (2.9.1)$$

where $O(h^4)$ denotes the sum of terms involving fourth and higher powers of h. (see Fig. 2.6).

Similarly, the value of the function at a point x-h may be expressed in the following way :

$$u(x-h) = u(x) - h\frac{du}{dx} + \frac{h^2}{2!}\frac{d^2u}{dx^2} - \frac{h^3}{3!}\frac{d^3u}{dx^3} + O(h^4) \qquad (2.9.2)$$

Addition of equations (2.9.1) and (2.9.2) gives :

$$u(x+h) + u(x-h) = 2u(x) + h^2\frac{d^2u}{dx^2} + O(h^4) \qquad (2.9.3)$$

If the term $O(h^4)$ is assumed negligible in comparison with lower powers of h then after dividing by h^2 (2.9.3) becomes :

$$\left.\frac{d^2u}{dx^2}\right|_{x=x} \simeq \frac{1}{h^2}\left[u(x+h) - 2u(x) + u(x-h)\right] \qquad (2.9.4)$$

with an error of order h^2.

When eq. (2.9.2) is subtracted from eq. (2.9.1) then the result is the central difference formula for the approximation of a first derivative,

i.e., $$\left.\frac{du}{dx}\right|_{x=x} \simeq \frac{1}{2h}\left[u(x+h - u(x-h)\right] \qquad (2.9.5)$$

again with an error of order h^2, if terms of $O(h^3)$ are neglected.

Other approximations to $\frac{du}{dx}$ at x = x are given either by (2.9.1), giving a forward difference formula :

$$\frac{du}{dx}_{x=x} \simeq \frac{1}{h}\left[u(x+h - u(x)\right] \qquad (2.9.6)$$

or by eq. (2.9.2), giving a backward difference formula:

$$\frac{du}{dx}_{x=x} \simeq \frac{1}{h}\left[u(x) - u(x-h)\right] \qquad (2.9.7)$$

Both equations (2.9.6) and (2.9.7) involve neglecting terms of $O(h^2)$ and thus have errors of approximation equal to $O(h)$.

It can be seen that the right hand side of eq. (2.9.5) approximates the tangent to P, in Fig. 2.6, by the chord AB and equations (2.9.6) and (2.9.7) approximate the tangent by the chords PB and AP respectively.

The dependent variable u is now considered to be a function of the independent variables x and t.

As partial derivatives are constructed with only one variable at a time changing, whilst the others are considered constant then it is clear that equations (2.9.4) to (2.9.7) for partial derivatives, say with respect to the variable t become :

$$\frac{\partial^2 u}{\partial t^2}_{t=t} \simeq \frac{1}{k^2}\left[u(x,t+k) - 2u(x,t) + u(x,t-k)\right] \qquad (2.9.8)$$

$$\frac{\partial u}{\partial t}_{t=t} \simeq \frac{1}{2k}\left[u(x,t+k) - u(x,t-k)\right] \qquad (2.9.9)$$

$$\frac{\partial u}{\partial t}_{t=t} \simeq \frac{1}{k}\left[u(x,t+k) - u(x,t)\right] \qquad (2.9.10)$$

$$\frac{\partial u}{\partial t}_{t=t} \simeq \frac{1}{k}\left[u(x,t) - u(x,t-k)\right] \qquad (2.9.11)$$

all with equivalent orders of approximation as their corresponding total differential equations (2.9.4) to (2.9.7). In the same way finite difference forms of partial derivatives with respect to the independent variable x can also be constructed, with t fixed and x at points x, x+h and x-h.

To illustrate the formation of a finite difference equation we will use, as many writers do, the one dimensional heat flow equation in non-dimensional form :

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \tag{2.9.12}$$

The numerical solution of this equation as with other partial differential equations usually requires that the solution domain be divided up into a grid of points or finite difference net.

Then small changes in x and t, i.e., Δx and Δt, are equal to the sides of the rectangles shown in Fig. 2.7, i.e., h and k respectively. The coordinate system is then defined as $x = i\ h$ and $t = jk$, where i and j are integers.

The value of u at any point P is then $u_P = u(ih,\ jk)$ or simply $u_{i,j}$.

The simplest scheme for transforming the heat equation (2.9.12) into a numerical system is shown in Fig. 2.8(a) and is termed an explicit formation. When $\frac{\partial u}{\partial t}$ is approximated by the forward difference eq. (2.9.10) and $\frac{\partial^2 u}{\partial x^2}$ by the equivalent central difference equation to eq. (2.9.4), at the time level j, then eq. (2.9.12) is transformed into :

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \tag{2.9.13}$$

giving $u_{i,j+1} = u_{i,j} + \frac{k}{h^2}(u_{i,j+1} - 2u_{i,j} + u_{i-1,j})$ (2.9.14)

and so the new value of u at point i and time t + Δt is expressed explicitly in terms of values at time t. Further requirements are that the initial conditions, i.e., u at time t = 0 for all i, and at the boundaries, i.e., u at x = 0 and x = 1 for all j, must be specified. The solution may then be "marched out" row by row.

All is not as simple as it first appears, however, as decisions, regarding the choice of different schemes, the number of grid points, their pattern and spacing depends on such things as stability, convergence and accuracy, each of which will be dealt with in greater detail in Chapter 5.

In the above scheme it can be shown that the solution of the difference equations (2.9.14) will converge to the solution of the differential equations as the grid points move closer and closer together providing,

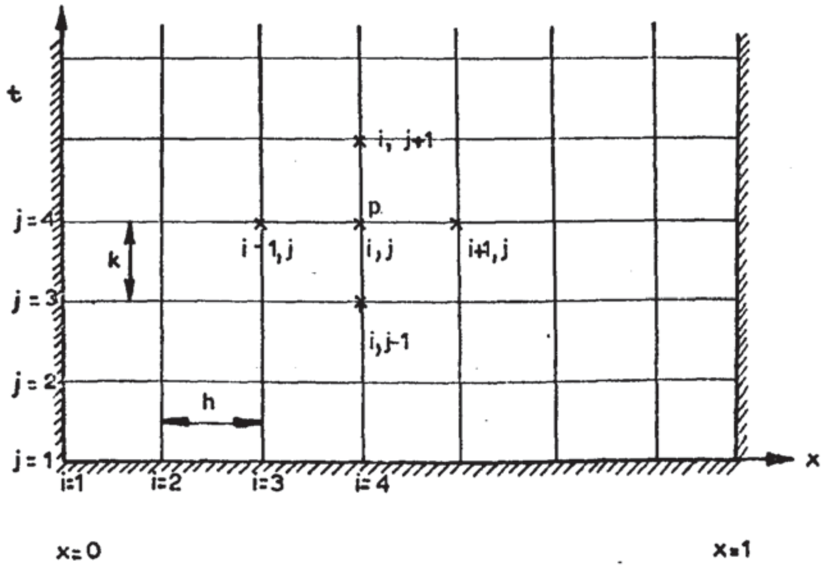
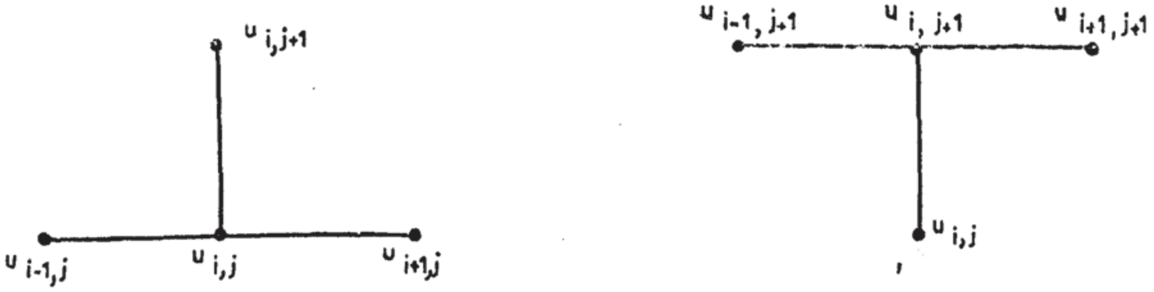


FIGURE 2.7



(A) EXPLICIT GRID POINTS

(B) IMPLICIT GRID POINTS

FIGURE 2.8

$$0 < \frac{\Delta t}{\Delta x^2} < \frac{1}{2} \tag{2.9.15}$$

The condition for stability for the explicit scheme is also that

$$\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{2.9.16}$$

where stability means that errors introduced into the solution decay as the solution proceeds.

Here the conditions for convergence and stability are the same, although this is generally not so, in fact the stability requirement may be lifted if an implicit solution of eq. (2.9.12) is used. In such solutions the unknowns are expressed in terms of other unknowns to give a system of simultaneous equations which may then be solved by a standard procedure.

Thus with reference to Fig. 2.8(b), if $\frac{\partial u}{\partial t}$ is replaced by the backward difference eq. (2.9.11) and $\frac{\partial^2 u}{\partial x^2}$ replaced again by the equivalent to the central difference equation to eq. (2.9.4), but this time at the time level $t + \Delta t$, then eq. (2.9.12) is transformed into :

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} \tag{2.9.17}$$

then $u_{i,j+1}\left(1 + \frac{2k}{h^2}\right) - \frac{k}{h^2}\, u_{i+1,j+1} - \frac{k}{h^2}\, u_{i-1,j+1} = u_{i,j}$ (2.9.18)

if $r =-\frac{k}{h^2}$ then eq. (2.9.18) becomes :

$$r\, u_{i+1,j+1} + (1 - 2r)\, u_{i,j+1} + r\, u_{i-1,j+1} = u_{i,j} \tag{2.9.19}$$

The problem then reduces to the solution of a tridiagonal matrix, where the co-efficients of the matrix are r, (1 - 2r) and r. The solutions of such equations can be acheived very efficiently by the Double Sweep or sometimes the Gauss Seidel methods, providing the boundary conditions are included.

If the resulting finite difference equation, eq. (2.9.19), had involved non-linear terms then solutions may be obtained by applying non-linear techniques, such as Newton-Raphson. In certain circumstances, i.e., unsteady flow, solutions can be obtained by holding one of the non-linear terms constant (if the degree of non-linearality is two) and then treat the system as linear, the constants are then updated at the ends of each iteration.

Although the example chosen above is a simple one the principles involved are those which are used in transforming the unsteady flow equations into finite difference equations.

CHAPTER THREE

LITERATURE SURVEY

3.0 Introduction

This section reviews the finite difference schemes and their methods of solution which have been developed by previous workers. Where such schemes have been suggested for use in a channel network their applicability is discussed.

The review starts with two papers that are themselves reviews of work up until the dates of their publication. Further developments are discussed afterwards.

3.1 Difference Solutions of the Shallow-Water Equation by J. A. Liggett and D. A. Woolhiser

The above paper [12], published in April 1967 provides an excellent discussion of the difference solutions available at the time of publication. Its Closure [13], published in February 1969 suggests a further scheme which changed the writers views regarding explicit schemes. The authors object was to present some of the acceptable numerical methods which could be used in connection with the shallow water equations so that workers who are new to the science may not make the same mistakes that have occurred in the past. The investigation was primarily concerned with the overland flow application although the methods suggested are applicable to unsteady flow in open channels.

The basic equations used are :

$$\frac{\partial h}{\partial t} + u\,\frac{\partial h}{\partial x} + h\,\frac{\partial u}{\partial x} = q \tag{3.1.1}$$

and

$$\frac{\partial u}{\partial t} + u\,\frac{\partial u}{\partial x} + g\,\frac{\partial h}{\partial x} = g\,(So - Sf) - \frac{qu}{h} \tag{3.1.2}$$

where h is the depth of flow, u the mean velocity, q the lateral inflow per unit area per unit time, So the channel slope and $Sf = \frac{u^2}{C^2R}$ where R is equal to h, implying a wide channel or overland flow.

The equations (3.1.1) and (3.1.2) are reduced to dimensionless form by defining :

$u^* = \frac{u}{Vo}$, $h^* = \frac{h}{Ho}$, $x^* = \frac{x}{Lo}$, $t^* = \frac{t\ Vo}{Lo}$, $Fo = \frac{Vo}{\sqrt{gHo}}$, $So = \frac{Vo^2}{C^2 Ho}$

and $k = \frac{So\ Lo}{Fo^2 Ho}$, in which Lo = length of reach (see Fig. 3.1), Ho = normal depth of flow for Qo = q Lo, Vo is the normal velocity of flow for Qo = q Lo = HoVo, assuming the total inflow arises from a constant lateral inflow q.

After substitution of the above into equations (3.1.1) and (3.1.2) the following two dimensionless equations are obtained in which the asterisks have been dropped :

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} + h\frac{\partial u}{\partial x} = 1 \qquad (3.1.3)$$

and

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{Fo^2}\frac{\partial h}{\partial k} = k\left(1 - \frac{u^2}{h}\right) - \frac{u}{h} \qquad (3.1.4)$$

The authors then introduce the concept of the difference operator $L\delta$ where :

$$L_\delta\ Z^{(\delta)} = f^{(\delta)} \qquad (3.1.5)$$

$Z^{(\delta)}$ is the solution of the finite difference scheme and $f^{(\delta)}$ is a function at the particular grid points to which the solution applies.

To illustrate one of the short comings of finite differences the authors chose as their first example an unstable explicit method in which h and u are evaluated at the same grid points.

The subscript j is for referencing a node in the x direction and i in the t direction (Fig. 3.2)

For the unstable scheme the left side of (3.1.5) becomes :

$$\left[L_\delta\ (Z_1)^{(\delta)}\right]_j^i = \left[\frac{h_j^{i+1} - h_j^i}{\Delta t} + u_j^i\frac{(h_{j+1}^i - h_{j-1}^i)}{2\ \Delta x} + h_j^i\ \frac{(u_{j+1}^i - u_{j-1}^i)}{2\ \Delta x}\right] \qquad (3.1.6)$$

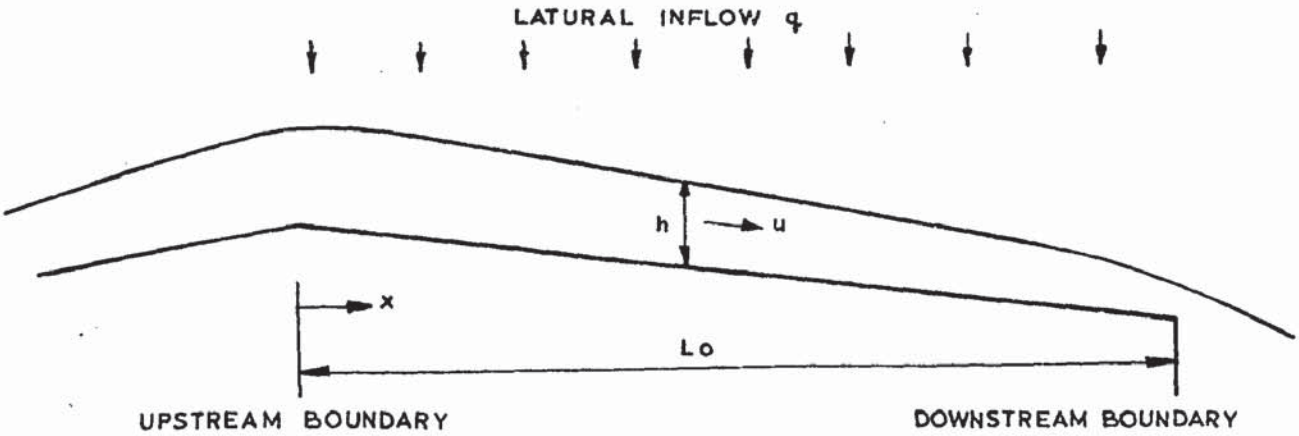


FIGURE 3.1

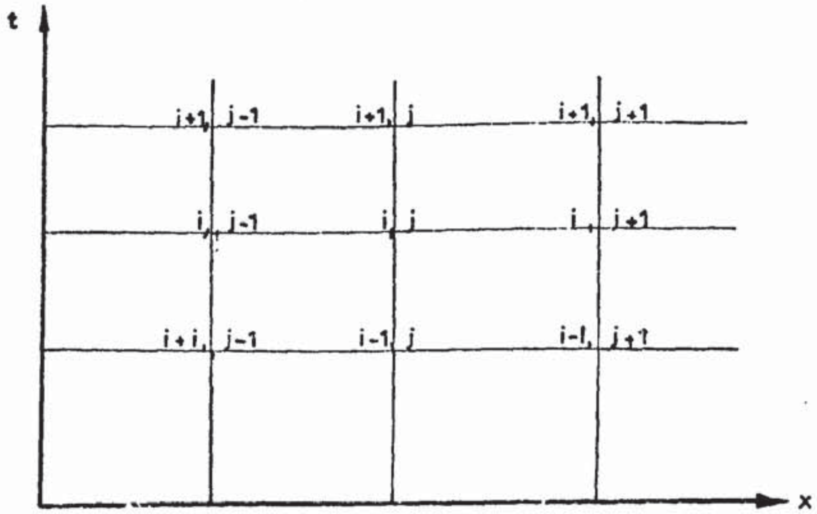


FIGURE 3.2

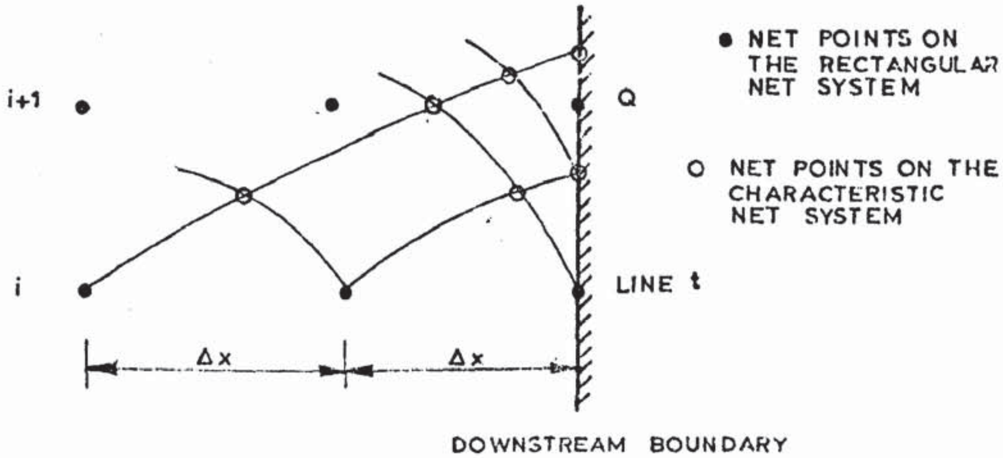

i+1
i
Q
LINE t
Δx
Δx
● NET POINTS ON THE RECTANGULAR NET SYSTEM
O NET POINTS ON THE CHARACTERISTIC NET SYSTEM
DOWNSTREAM BOUNDARY

FIGURE 3.3

and

$$\left[L_\delta \; (Z_2)^{(\delta)} \right]^i_j = \left[\frac{u^{i+1}_j - u^i_j}{\Delta t} + u^i_j \frac{(u^i_{j+1} - u^i_{j-1})}{2\Delta x} + \frac{1}{Fo^2} \frac{(h^i_{j+1} - h^i_{j-1})}{2\Delta x} \right] \qquad (3.1.7)$$

the function on the right side of eq. (3.1.4) is evaluated at the point i,j.

The approximation of the difference equation to the differential equation can be determined by expanding each variable u^{i+1}_j, h^i_{j+1} etc., in equations (3.1.6) and (3.1.7) by a Taylors series, which on substitution into equatinns (3.1.6) and (3.1.7) gives :

$$\left[L\delta \; (Z_1)^{(\delta)} \right]^i_j = \{ \frac{\partial h}{\partial t} + \frac{\Delta t}{2} \frac{\partial^2 h}{\partial t^2} + O(\Delta t^2), + u \left[\frac{\partial h}{\partial x} + O(\Delta x^2) \right] + h \left[\frac{\partial u}{\partial x} + O(\Delta x^2) \right] \qquad (3.1.8)$$

and

$$\left[L\delta \; (Z_2)^{(\delta)} \right]^i_j = \{ \frac{\partial u}{\partial t} + \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} + O(\Delta t^2), + u \left[\frac{\partial u}{\partial x} + O(\Delta x^2) \right] + \frac{1}{Fo^2} \left[\frac{\partial h}{\partial x} + O(\Delta x^2) \right] \qquad (3.1.9)$$

If Δt is now defined as $\gamma \Delta x$ where γ is a constant and Δx is allowed to approach zero then equations (3.1.8) and (3.1.9) become :

$$\left[L_\delta \; (Z_1)^{(\delta)} \right]^i_{j \; \Delta x \to o} = \frac{\partial h}{\partial t} + u \frac{\partial h}{\partial x} + h \frac{\partial u}{\partial x} + O(\Delta x) \qquad (3.1.10)$$

and

$$\left[L_\delta \; (Z_2)^{(\delta)} \right]^i_{j \; \Delta x \to o} = \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{1}{Fo^2} \frac{\partial h}{\partial x} + O(\Delta x) \qquad (3.1.11)$$

and so the difference operator approximates the differential operator to the first order. The approximation can, however be made second order by letting $\Delta t = \gamma \; \Delta x^2$. The disadvantage of the above scheme is that it is not stable and therefore cannot be used. Some workers have found the scheme empirically stable for supercritical flow only or if the right side of (3.1.4) is written for the point i+1, j. However in this case it was found that over a long period of time, inflow may be greater or less than the outflow plus the accumulation of storage.

The next scheme suggested by the authors is the Diffusing Method in which different quotients with respect to time use averaged values of the function before the time step, again h's and u's are evaluated at the same nodes.

$$\left[L_\delta\ (Z_1)^{(\delta)}\right]^i_j = \left[\frac{h^{i+1}_j - \frac{1}{2}\,(h^i_{j+1} - h^i_{j-1})}{\Delta t} + u^i_j\ \frac{(h^i_{j+1} - h^i_{j-1})}{2\ \Delta x} + h^i_j\frac{(u^i_{j+1} - u^i_{j-1})}{2\ \Delta x}\right] \qquad (3.1.12)$$

and

$$\left[L\delta\ (Z_2)^{(\delta)}\right]^i_j = \left[\frac{u^{i+1}_j - \frac{1}{2}\,(u^i_{j+1} + u^i_{j-1})}{\Delta t} + \frac{u^i_j(u^i_{j+1} - u^i_{j-1})}{2\ \Delta x} + \frac{1}{Fo^2}\frac{(h^i_{j+1} - h^i_{j-1})}{2\ \Delta x}\right] \qquad (3.1.13)$$

The right side of (3.1.4) is evaluated at the central point i,j.

There is one variation on this method and that is when averages are used for the right side of (3.1.4) and $\frac{\partial}{\partial x}$ (u.h) is approximated by :

$\dfrac{(u\ h)^i_{j+1} - (u\ h)^i_{j-1}}{2\ \Delta x}$ and is known as the conservation form.

When the terms in (3.1.12) and (3.1.13) are expanded by a Taylors series then :

$$\left[L\delta\ (Z_1)^{(\delta)}\right]^i_j = \frac{\partial h}{\partial t} + \frac{\Delta x^2}{2\Delta t}\frac{\partial^2 h}{\partial x^2} + u\frac{\partial h}{\partial x} + h\frac{\partial u}{\partial x} + O(\Delta x^2,\ \Delta t) \qquad (3.1.14)$$

and

$$\left[L\delta\ (Z_2)^{(\delta)}\right]^i_j = \frac{\partial u}{\partial t} + \frac{\Delta x^2}{2\Delta t}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial u}{\partial x} + \frac{1}{Fo^2}\frac{\partial h}{\partial x} + O(\Delta x^2,\ \Delta t) \qquad (3.1.15)$$

If $\Delta t = \gamma\ \Delta x^2$ then the difference operator approximates a different system of differential equations than (3.1.3) and (3.1.4), or if Δt is taken very small in relation to Δx the second term may assume considerable importance. If the ratio of Δt to Δx satisfies the Courant condition i.e.,

$$\frac{\Delta t}{\Delta x} < \frac{1}{|u| + c} \qquad (3.1.16)$$

then the diffusing method is theoretically stable but the solution to this system of difference equations does not necessarily converge to the solution of the differential equations because of the difference operator does not approximate the differential operator.

The third scheme suggested by the authors is again an explicit one and is a version of the commonly known Leap Frog Method in which time derivatives are approximated by central differences. Again the h's and u's are evaluated at the same nodes.

$$\left[L_\delta \; (Z_1)^{(\delta)}\right]_j^i = \left[\frac{(h_j^{i+1} - h_j^{i-1})}{2\;\Delta t} + u_j^i\frac{(h_{j+1}^i - h_{j-1}^i)}{2\;\Delta x} + h_j^i\frac{(u_{j+1}^i - u_{j-1}^i)}{2\;\Delta x}\right] \qquad (3.1.17)$$

and

$$\left[L_\delta \; (Z_2)^{(\delta)}\right]_j^i = \left[\frac{(u_j^{i+1} - u_j^{i-1})}{2\;\Delta t} + u_j^i\frac{(u_{j+1}^i - u_{j-1}^i)}{2\;\Delta x} + \frac{1}{Fo^2}\;\frac{(h_{j+1}^i - h_{j-1}^i)}{2\;\Delta x}\right] \qquad (3.1.18)$$

the right side of equation (3.1.4) is evaluated at the central point i, j. In this scheme the difference operator approximates the differential operator to the second order, $O(\Delta x^2)$, and is stable if the Courant condition is satisfied. The leap frog method requires initial conditions at row $t = t_o$ and either $t = t_o + \Delta t$ or $t = t_o - \Delta t$.

To conclude the explicit schemes suggested by the writers, they give two Lax-Wendroff methods. The first is a two cycle scheme which first uses the difference scheme of the diffusing method to advance the solutions one row and then uses the leap frog differencing scheme for the second row. This gives a method with second order approximation which should exhibit positive damping of short wave disturbances. The second method is the Single Step Lax-Wendroff method in which the dependent variables h and u.h are expanded as a Taylors series. The finite difference scheme (outlined below) then estimates the terms up to and including second order derivatives.

Equations (3.1.3) and (3.1.4) may be written in the form :

$$\frac{\partial h}{\partial t} + \frac{\partial m}{\partial x} - 1 = 0 \qquad (3.1.19)$$

and $$\frac{\partial m}{\partial t} + \frac{\partial}{\partial x}\left(\frac{m^2}{h}\right) + \frac{1}{2Fo^2}\frac{\partial}{\partial x}(h^2) - kh\left(1 - \frac{m^2}{h^3}\right) = 0 \qquad (3.1.20)$$

in which m = u.h.

In matrix notation (3.1.19) and (3.1.20) may be written as :

$$\frac{\partial W}{\partial t} + \frac{\partial G}{\partial x} + K = 0 \qquad (3.1.21)$$

where $W = \begin{bmatrix} m \\ h \end{bmatrix}$, $G = \begin{bmatrix} \frac{m^2}{h} + \frac{h^2}{2Fo^2} \\ m \end{bmatrix}$ and $K = \begin{bmatrix} kh(1 - \frac{m^2}{h^3}) \\ 1 \end{bmatrix}$

now;

$$W(x,t + \Delta t) = W(x,t) + \Delta t \frac{\partial W}{\partial t} + \frac{\Delta t^2}{2!} \frac{\partial^2 W}{\partial t^2} + O(\Delta t^3) \qquad (3.1.22)$$

If the truncation error $O(\Delta t^3)$ is neglected and substitutions are made for $\frac{\partial W}{\partial t}$ and $\frac{\partial^2 W}{\partial t^2}$ from (3.1.21), then (3.1.22) becomes :

$$W(x,t + \Delta t) = W(x,t) - \Delta t \left(\frac{\partial G}{\partial x} + K\right) + \frac{\Delta t^2}{2}\left\{ \frac{\partial}{\partial x} \left[A\left(\frac{\partial G}{\partial x} + K\right)\right] - \frac{\partial K}{\partial t} \right\} \qquad (3.1.23)$$

where $A = \begin{bmatrix} \frac{2m}{h} & \frac{h}{Fo^2} - \frac{m^2}{h^2} \\ 1 & 0 \end{bmatrix}$;

The above eq. (3.1.23) is then transformed into finite differences in which space differentials are approximated over the interval j+1 and j-1.

The only implicit scheme suggested by the writers is :

$$\left[L_\delta \, (Z_1)^{(\delta)} \right]_j^{i+\frac{1}{2}} = \left[\frac{(h_j^{i+1} - h_j^i)}{\Delta t} + \tilde{u} \left\{ \frac{(h^i_{j+1} - h^i_{j-1}) + (h^{i+1}_{j+1} - h^{i+1}_{j-1})}{4\Delta x} \right\} + \tilde{h} \left\{ \frac{(u^i_{j+1} - u^i_{j-1}) + (u^{i+1}_{j+1} - u^{i+1}_{j-1})}{4\Delta x} \right\} \right] \qquad (3.1.24)$$

and

$$\left[L_\delta \, (Z_2)^{(\delta)} \right]_j^{i+\frac{1}{2}} = \left[\frac{(u_j^{i+1} - u_j^i)}{\Delta t} + \tilde{u} \left\{ \frac{(u^i_{j+1} - u^i_{j-1}) + (u^{i+1}_{j+1} - u^{i+1}_{j-1})}{4 \, \Delta x} \right\} + \frac{1}{Fo^2} \left\{ \frac{(h^i_{j+1} - h^i_{j-1}) + (h^{i+1}_{j+1} - h^{i+1}_{j-1})}{4 \, \Delta x} \right\} \right] \qquad (3.1.25)$$

where the right side of eq. (3.1.4) is centred at $i+\frac{1}{2}$, j and where

$$\tilde{u} = \frac{1}{2} (\hat{u}_j^{i+1} + u_j^i) \text{ and } \tilde{h} = \frac{1}{2} (\hat{h}_j^{i+1} + h_j^i) \tag{3.1.26}$$

In the products $\tilde{u}\, h^i$, $\tilde{u}\, u^i$ or $\tilde{h}\, u^i$, $\hat{u}^{i+1} = u^{i+1}$and $\hat{h}^{i+1} = h^{i+1}$. In the products $\tilde{u}\, h^{i+1}$, $\tilde{u}\, u^{i+1}$ or $\tilde{h}\, h^{i+1}$, $\hat{u}^{i+1} = u^i$ and $\hat{h}^{i+1} = h^i$ for the initial advancement, and thereafter they are set to the recently obtained values, i.e., $\hat{u}^{i+1} = u^{i+1}$ and $\hat{h}^{i+1} = u^{i+1}$. In this way the equations are linearised to facilitate the solution of the nonlinear equations by the Double Sweep method (which will be dealt with in detail later) until the difference between subsequent values of u^{i+1} and h^{i+1} fall within a tolerance.

As the two dependent variables u and h are evaluated at the same nodes then if at a boundary only one of the variables is expressed as a function of time then it is necessary to determine the other. One of the ways the authors suggest of doing this is to assume the velocity at the boundary is zero and thus the flow is then symmetrical about that boundary, i.e., $u(x) = -u(-x)$, $So(x) = - So(-x)$ and $h(x) = h(-x)$. When these symmetry conditions are substituted into the momentum equation, that equation is satisfied to the order Δx without further computation. This gives a system in which the boundary is approximated to a lower degree than the interior. The error produced is actually of the order $k.\Delta x$ so the approximation is considered satisfactory for small k but may be unsatisfactory for large k. The resulting depth is higher than it should be.

The implicit scheme approximates the differential operator to the second order (neglecting boundaries) and the ratio $\Delta t/\Delta x$ is not governed by the Courant condition. However by increasing the time step further than this inaccuracies would result at the boundaries and stability problems could occur.

A more accurate way of determining boundary conditions for finite difference schemes which calculate the two dependent variables at each node is to use, at the boundaries, the method of characteristics. There are two suggested ways to do this the most accurate of which is to build a characteristic net from the last set of calculated points on line t, as shown in Fig. 3.3. The net is continued until the desired point Q is included and the net values are interpolated at that point. Usually two to twelve points have to be computed to obtain the desired point.

The second method which is not so accurate as the first is to extend a characteristic from the desired boundary point Q to the last known line of values on line t, as shown in Fig. 3.4.

The process is an iterative one and the main steps are ;
(i) by using the value of u and h at the penultimate point before the boundary, the approximate slope of PQ is determined,
(ii) PQ is projected down from Q to determine η,
(iii) u and h are found at P by interpolation along line t. Using these values and the appropriate characteristic equation the value at the boundary Q is determined. The slope PQ is then recalculated and the process repeated by returning to step (ii) until there is no change at Q.

This method has the disadvantage that if the time steps are large thenthe method is not very accurate, also that characteristics may bend sharply near boundaries. If one is going to use the characteristic method for boundary conditions then the program becomes complex and it is simpler to use the characteristic method throughout.

In the examination of the stability and approximation of the difference schemes the interior points only were considered i.e., ignoring the boundaries. If the method of characteristics was used to determine the boundary values it was assumed that this would be of a higher order of approximation than the interior. Richtmyer [15] determined the stability criteria of the differential systems in linearised form, as no stability analysis exists that can deal with totally non-linear partial differential systems.

To investigate the effects of instability and flexibility the authors conducted a series of numerical experiments. The characteristic method was used

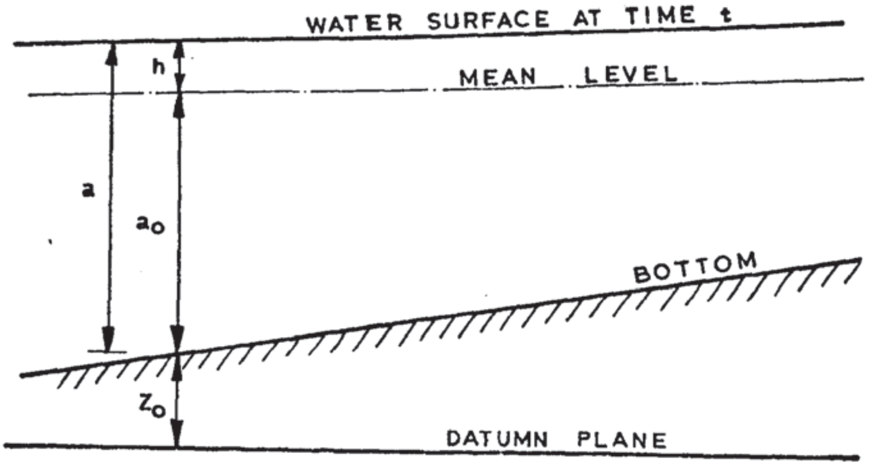


FIGURE 3.5

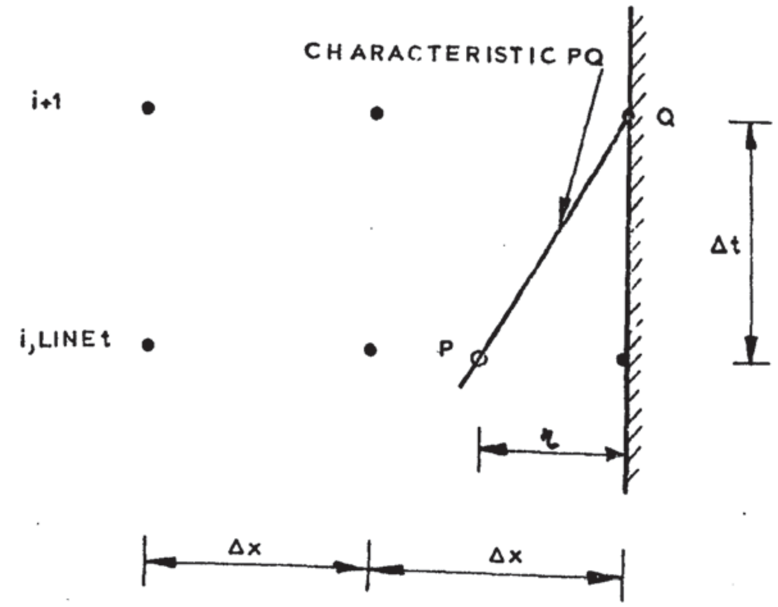


FIGURE 3.4

to provide initial data as this method proved to be consistently most accurate and not subject to convergence problems. Boundary conditions were; u = o at x = o for all t $\geqslant$ o and u = $\sqrt{h}$/Fo at x = 1 (i.e., critical depth) or no downstream boundary condition at all in supercritical flow.

It was observed during testing that instabilities often originated at or near boundaries although it was not established whether the perturbations were due to the boundary conditions or to the large curvature of depth and velocity that occurs near a boundary.

Three steady state profiles were inserted into the different finite difference schemes, each chosen to eliminate problems associated with the difference method at the upstream boundary and to minimise curvature in the depth profile near each boundary. Two perturbations were introduced into the steady state profiles along the length of the reach to investigate their damping and possible interaction with the boundaries. Tests were carried out with Fo varying from 1.0 to 2.0 and So.Lo/Ho varying from 0.1 to 22.5. A further test was that of a rising hydrograph at the downstream boundary, again observations were checked with the characteristic method.

The unstable method was found to be completely unsatisfactory as the steady state profiles broke up into long waves with growing amplitudes. If the time step was halved then this decreased the rate of growth of the oscillations but they soon became unstable. The rising hydrographs followed the solutions obtained by the characteristic method fairly well until the upstream boundary was felt at the downstream end, at which point the method then became unstable.

Results showed that the diffusing method is empirically stable for all cases except where it did not approximate the differential equations, in which case saw-toothed waves were observed superimposed on the solution. This, according to the authors is characteristic of theoretically stable but inflexible methods (i.e., perturbations in the steady state profiles may damp out rapidly or, with some parameter values, may grow in magnitude). The authors state that filtering schemes could be devised that would smooth out such short wave disturbances, thus making an inflexible (but theoretically stable) scheme useful for indefinite calculations.

The first Lax-Wendroff method and the leap frog method both proved to be inflexible in some instances, as saw-toothed disturbances crept into the solution. In some cases positive damping was observed but in others disturbances were so large that they prevented a continuation of the computation. Again a filtering scheme could be used. However, the second single step Lax-Wendroff method did give very good results in that it was able to retain the steady state profiles and follow the rising hydrographs closely.

The implicit method also performed very well as it rapidly converged to the steady state profiles. There was only one case in which it did not do well and this was the rising hydrograph with a high Fo, however, the authors later found that these irregularities were due to algebraic errors and not the methods tried.

The initial conclusions which the authors came to were that efforts should be concentrated on the characteristic and implicit methods due to the inflexibility of the explicit methods. However, in their Closure [13] they found that the single step Lax-Wendroff method was very flexible and accurate, also they stated that inflexible methods can be made useful by properly averaging the resistance term. Finally, that although a linear stability analysis is a very useful tool, the stability criteria developed are approximate for non-linear equations and that a linearly stable difference scheme may fail completely when the non-linear terms become important.

3.2 Tidal Computations for Rivers, Coastal Areas and Seas by J. J. Dronkers

The above paper [8] published in January 1969 gives a review of one and two-dimensional tidal computations by means of the computer and discusses their practical applications. Although the paper is primarily concerned with tidal computations the schemes suggested are applicable to general unsteady flow in open channels. The two dim ensional scheme will not be disucssed here.

Dronkers consideres it reasonable, as many workers do, to use the same flow resistance term in unsteady flow as is used in steady flow, i.e., that Chezy's Law and Manning's Law hold for tidal motion. Although Manning's Law may give a somewhat more accurate dependence on depth he has found no preference of one over the other. In general, C, in Chezy's Law, also depends on the schematization of the region (including the depth) as well as bottom frictional resistance. Friction co-efficients are slightly modified to take into account irregularities in the shapes of rivers including bends. When the boundary conditions are measured then internal data should also be collected in terms of depth and velocity so that the chosen 'C' may be verified.

In the tidal calculations the mean water level is used, as shown in Fig. 3.5. This is determined from observations over a full tidal cycle.

The basic one dimensional equations used are :

$$\rho\left(\frac{\partial u}{\partial t} + u\,\frac{\partial u}{\partial x}\right) = -\rho g\,\frac{\partial h^*}{\partial x} - \frac{\rho g|u|u}{C^2(a_o+h)} + \frac{Wx}{a_o+h} \qquad (3.2.1)$$

and $$\frac{\partial}{\partial x}(A\,u) + b\,\frac{\partial h}{\partial t} + q = 0 \qquad (3.2.2)$$

in which $h^* = z_o + a_o + h$, $z_o(x)$ = height of bed with respect to datum plane, $a_o(x)$ = height of mean water level with respect to the bottom, $h\ (x,\ t)$ = height of the water level with respect to mean water level, u = mean velocity in a cross section, Wx = x component along the river of the wind force on the water surface, C is Chezy's co-efficient, ρ is the density of the water, g the acceleration due to gravity, A is = $b_s\ (a_o+h)$, where $b_s\ (x,a_o+h)$ and $b(x,a_o+h)$ are the stream widths and total widths (storage included) respectively, q is the supplementary discharge per unit length. In the friction term above it will be noticed that (a_o+h) is used as the hydraulic radius, thus implying a wide channel.

If Q is taken as the dependent variable instead of u then the basic equations become :

$$\left(\frac{da_o}{dx} + \frac{\partial h}{\partial x}\right)\left(1 - \frac{\alpha Q^2}{g(a_o+h)A^2}\right) - \frac{(b_s + \alpha b)}{g\,A^2}\, Q\, \frac{\partial h}{\partial t} + \frac{1}{Ag}\,\frac{\partial Q}{\partial t} \qquad (3.2.3)$$

$$+ \frac{Q|Q|}{C^2A^2(a_o+h)} - \frac{W}{g(a_o+h)} + I = 0$$

$$\text{and } \frac{\partial Q}{\partial x} + b\,\frac{\partial h}{\partial t} = 0 \qquad (3.2.4)$$

in which I = bed slope and q is taken as zero. α is defined as $\int \frac{u_A^3\, dA}{u^3\, A}$ and takes into account the effect of the non-uniform velocity distribution on the convective acceleration or Bernoulli term.

The above equations (3.2.1), (3.2.2), (3.2.3) and (3.2.4) are for gradually varying cross sections and if a sudden narrowing or widening occurs then a formula must be applied across the jump that takes into account the head loss. For the case of subcritical flow this is :

$$\eta\left[u^2(x_2) - u^2(x_1)\right] = 2g\left[h(x_1) - h(x_2)\right] \qquad (3.2.5)$$

in which x_2 is the narrowest part of the contraction and x_1 is downstream of the jump. η depends on such factors as friction, irregularity and contraction of the flow etc.

If the flow is from x_1 to x_2 then η is very near 1, however if the flow is from x_2 to x_1 then η could be much smaller. If accelerations and decelerations are small, i.e., changes in velocity of less than 1m/sec. then a modified C may be used as this also depends on u^2. It may be necessary to have different 'C's for ebb and flood.

To analyse the stability of the finite difference schemes Dronkers formed two equations for a geeral linearised finite difference system :

$$(u_m^{n+1} - u_m^{n-r}) + a^*\,(h_{m+1}^{n+s} - h_{m-1}^{n+s}) + b^*\,u_m^{n+1} + c^* = 0 \qquad (3.2.6)$$

$$(h_{m+1}^{n+w} - h_{m+1}^{n}) + d^*\,(u_{m+2}^{n+1} - u_m^{n+1}) + f^* = 0 \qquad (3.2.7)$$

in which a* to f* are co-efficients, some of which depend on u and h evaluated at the previous time steps. r, s and w are positive integers and m and n are x and t grid spacings respectively. Stability is then analysed by a finite Fourier analysis of errors introduced into the finite difference equation (this method will be dealt with in more detail later).

In the general setting up of finite difference schemes Dronkers suggests that difference quotients with respect to distance should be central differences, although the central points for the h's may be different for the Q's or u's. Also if an explicit scheme is to be used then again a central difference is required with respect to time and if an implicit scheme is used then a forward difference quotient is required. Dronkers is of the opinion that by keeping the equations non-linear it is not only time consuming but also that the higher accuracy obtained is not necessary for practical applications. Also, particular attention should be paid to ensure that the values of the cross-sectional area and breadth etc. used are representative of the section to which the equations pertain.

The first scheme suggested for solution of the tidal equations is an explicit one and is an adaption of the leap frog method. In this adaption (see Fig. 3.6), the h and u nodes are displaced both in space and time.

The basic equations used are :

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial h}{\partial x} - \frac{g\,u|u|}{C^2(a_o+h)} \qquad (3.2.8)$$

and $\frac{\partial(Au)}{\partial x} + b\frac{\partial h}{\partial t} = 0$ (3.2.9)

In eq. (3.2.8) da_o/dx and dz_o/dx, which do not depend on time have been omitted. It is possible, however, to redefine h as the height with respect to datum h*, in which case $a_o + h$ becomes $h^* - z_o$.

Equations (3.2.8) and (3.2.9) are transformed into finite difference form to give :

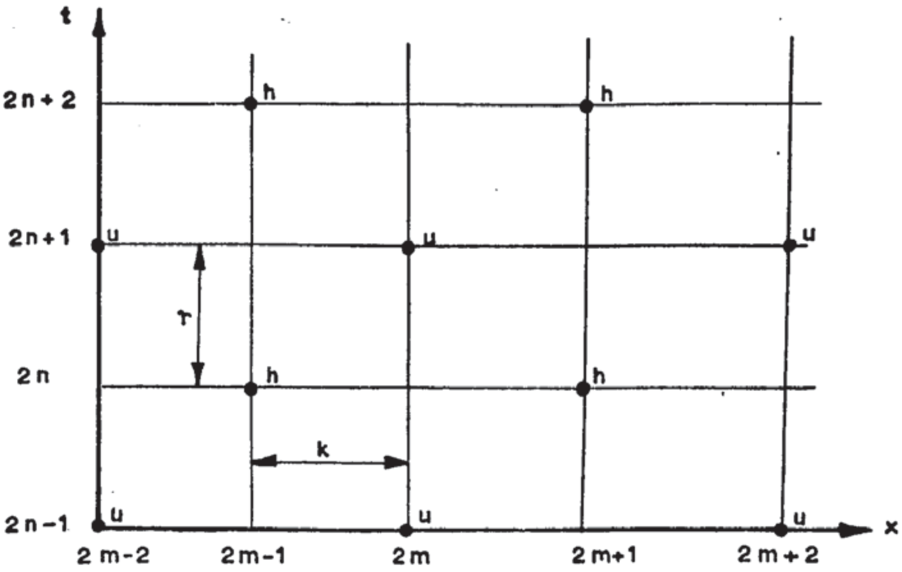


FIGURE 3.6

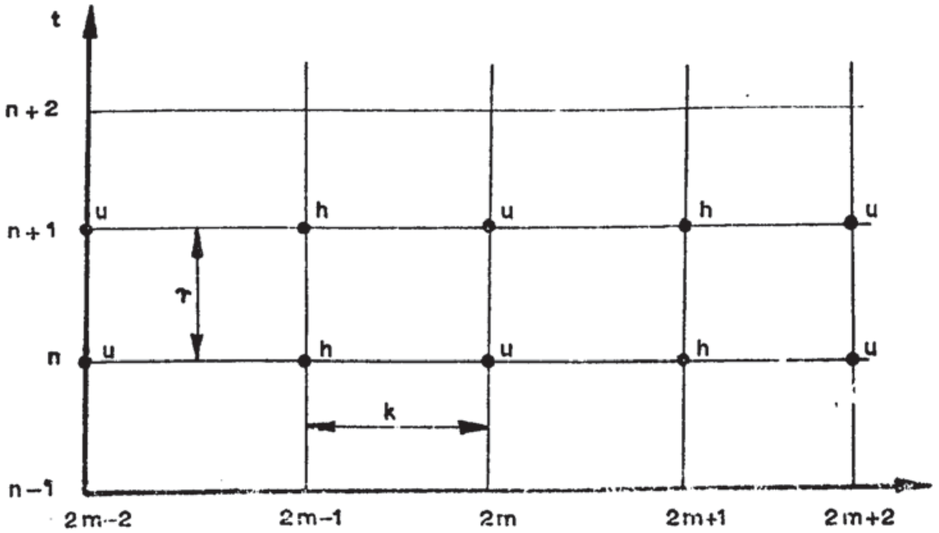


FIGURE 3.7

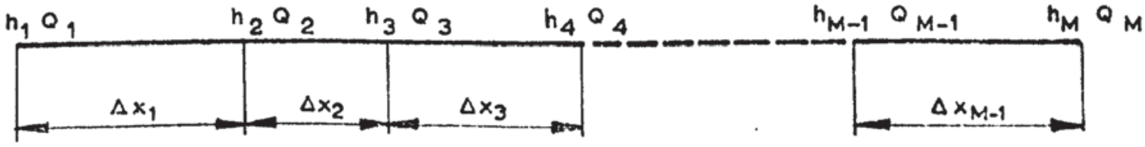


FIGURE 3.8

$$u_{2m}^{2n+1} = u_{2m}^{2n-1} - \left(\frac{T}{2k}\right) u_{2m}^{2n+1} \left(u_{2m+2}^{2n-1} - u_{2m+2}^{2n-1}\right) - \frac{g\,T}{k}\left(h_{2m+1}^{2n} - h_{2m-1}^{2n}\right)$$

$$- \frac{2g\ T\ |u_{2m}^{2n-1}|\,u_{2m}^{2n+1}}{C^2(a_o+h)_{2m}^{2n}} \qquad (3.2.10)$$

and $$h_{2m+1}^{2n+2} = h_{2m+1}^{2n} - \frac{T}{k(b)_{2m+1}^{2n}}\left(u_{2m+2}^{2n+1} A_{2m+2}^{2n} - u_{2m}^{2n+1} A_{2m}^{2n}\right) \qquad (3.2.11)$$

As the above system requires that initial values are given at $t = t_o$ and $t = T$ then it is suggested that the values of h are taken at t = o and u is either determined by $u(x, T) = u(x,o) - \frac{g\,T}{2k}\left[h(x + k,\ o) - h(x-k,\ o)\right]$, or for practical applications simply as the same at t = o. Boundary conditions are usually given as $h = h_o(t)$ at the estuary end and $u = u_o(t)$ at the upstream end of the river, although it is possible to have either at a particular boundary.

After applying the stability analysis discussed previously in which r=1 s = o, w = 2 and c* = f* = o it was found that :

$$\frac{k}{T} > \left(\frac{gA}{b}\right)^{\frac{1}{2}} \left(1 + \frac{b^*}{2}\right)^{-\frac{1}{2}} \qquad (3.2.12)$$

in which

$$b^* = \frac{2g\ T\ \left|u_{2m}^{2n-1}\right|}{C^2(a_o+h)_{2m}^{2n}} + \left(u_{2m+2}^{2n-1} - u_{2m-2}^{2n-1}\right)\frac{T}{2k} \qquad (3.2.13)$$

And so the scheme is stable providing the ratio of dx/dt complies with (3.2.12). It can be seen that the friction influences the stability in a favourable sense and that stability is maintained when it exceeds the Bernoulli term if the latter is negative.

The next scheme suggested is an implicit one in which the dependent variables are staggered in space only. (see Fig. 3.7).

The finite difference equations then become :

$$h_{2m+1}^{n+1} - h_{2m-1}^{n+1} = \frac{-2k}{Tg}\left(u_{2m}^{n+1} - u_{2m}^{n}\right) - \frac{2k\ u_{2m}^{n+1}\left|u_{2m}^{n}\right|}{(a_o+h)_{2m}^{n}\ (C_{2m}^{n})^2} - \frac{1}{2g}\ u_{2m}^{n+1}\left(u_{2m+2}^{n} - u_{2m-2}^{n}\right)$$

$$(3.2.14)$$

and $(Au)^{n+1}_{2m+2} - (Au)^{n+1}_{2m} = \frac{-2k}{T}\left[b^{n}_{2m+1}\,(h^{n+1}_{2m+1} - h^{n}_{2m+1})\right]$ (3.2.15)

In equations (3.2.14) the Bernoulli term is evaluated over 4k, which Dronkers says is a poor estimate unless short lengths are considered and that a better one would be :

$$\frac{u^{n+1}_{2m}\,(u^{n}_{2m+2} - u^{n}_{2m})}{2g} \tag{3.2.16}$$

Also, in eq. (3.2.15) to avoid calculating A at 2m+2 and 2m at which h is not computed, it is permissible to replace the left side with ;

$$A^{n}_{2m+1}\,(u^{n+1}_{2m+2} - u^{n+1}_{2m}) \tag{3.2.17}$$

if in the river there is a relatively small amplitude of vertical tide with respect to a_o. If this is not possible then the equation of continuity must be used in the form :

$$A\,\frac{\partial u}{\partial x} + u\,b_s\,\frac{\partial h}{\partial x} = -b\,\frac{\partial h}{\partial t} \tag{3.2.18}$$

The above equations (3.2.14) and (3.2.15) form a system of simultaneous linear tridiagonal equations which is solved by the Double Sweep Method.

When the stability of this system is considered with r = 0, s = 1 and w = 1 in equations (3.2.6) and (3.2.7), it is found that providing the friction term in b* exceeds the Bernoulli term then the scheme is unconditionally stable. However, when the Bernoulli term is large it is recommended that separate equations be applied such as eq. (3.2.5).

The second implicit scheme suggested by Dronkers is one in which Q is the dependent variable instead of u. This allows a better finite difference representation of the equation of continuity and the Bernoulli term.

The basic equations used are :

$$\frac{\partial h}{\partial x} = \frac{-1}{Ag}\,\frac{\partial Q}{\partial t} - \frac{Q|Q|}{C^2A^2(a_o+h)} + \frac{B}{A^2g}\,Q\,\frac{\partial h}{\partial t} \tag{3.2.19}$$

and $\frac{\partial Q}{\partial x} = -b\,\frac{\partial h}{\partial t}$ (3.2.20)

in which $B = b_s + \alpha b$.

If the same notation is used as in Fig. 3.7 then in (3.2.19)

$$Q\frac{\partial h}{\partial t} = \frac{Q^n_{2m}}{2T}\left[(h^{n+1}_{2m-1} - h^n_{2m-1}) + (h^{n+1}_{2m+1} - h^n_{2m+1})\right] \qquad (3.2.21)$$

and $$\frac{\partial h}{\partial x} = \frac{h^{n+1}_{2m+1} - h^{n+1}_{2m-1}}{2k} \qquad (3.2.22)$$

which on substitution into eq. (3.2.19) and after rearrangement gives :

$$\left[1 - \frac{k}{Tg}\left(\frac{BQ}{A}\right)^n_{2m}\right] h^{n+1}_{2m+1} - \left[1 + \frac{k}{Tg}\left(\frac{BQ}{A}\right)^n_{2m}\right] h^{n+1}_{2m-1} = \frac{-2k}{Tg\,A^n_{2m}}(Q^{n+1}_{2m} - Q^n_{2m}) - \frac{2k|Q^n_{2m}|Q^{n+1}_{2m}}{(C^2A^2a)^n_{2m}} - \frac{k}{Tg}\left(\frac{BQ}{A^2}\right)^n_{2m}(h^n_{2m+1} + h^n_{2m-1}) \qquad (3.2.23)$$

and eq. (3.2.20) becomes :

$$Q^{n+1}_{2m+2} - Q^{n+1}_{2m} = \frac{-2k\,b^n_{2m}}{T}(h^{n+1}_{2m+1} - h^n_{2m+1}) \qquad (3.2.24)$$

A, B and a (hydraulic radius) are average values at the Q nodes using the depths h_{i+1} and h_{i-1} at the time level nT.

The resulting set of simultaneous equations is again solved by the Double Sweep Method which will now be briefly dealt with.

Equations (3.2.23) and (3.2.24) may be expressed as :

$$\alpha_{2m}\,h'_{2m+1} - \beta_{2m}\,h'_{2m-1} + \gamma_{2m}\,Q'_{2m} = \sigma_{2m} \qquad (3.2.25)$$

and $$\epsilon_{2m}\,h'_{2m+1} + Q'_{2m+2} - Q'_{2m} = \lambda_{2m} \qquad (3.2.26)$$

in which $\alpha_{2m} = 1 - \frac{k}{Tg}\left(\frac{BQ}{A}\right)^n_{2m}$, $\beta_{2m} = 1 + \frac{k}{Tg}\left(\frac{BQ}{A}\right)^n_{2m}$, $\gamma_{2m} = \frac{2k}{Tg\,A^n_{2m}} + \frac{2k|Q^n_{2m}|}{(C^2A^2a)^n_{2m}}$,

$\sigma_{2m} = \frac{2k}{Tg\,A^n_{2m}}Q^n_{2m} - \frac{k}{Tg}\left(\frac{BQ}{A^2}\right)^n_{2m}(h^n_{2m+1} + h^n_{2m-1})$, $\epsilon_{2m} = \frac{2k\,b^n_{2m}}{T}$ and

$\lambda_{2m} = \frac{2k\,b^n_{2m}}{T}h^n_{2m+1}$.

In equations (3.2.25) and (3.2.26) the unknown values at n+1 are represented by the primes;

Which for m=1 and m = 2 the equations become :

$$\alpha_2 h'_3 - \beta_2 h'_1 + \gamma_2 Q'_2 = \sigma_2 \qquad (3.2.27)$$

$$\varepsilon_2 h'_3 + Q'_4 - Q'_2 = \lambda_2 \qquad (3.2.28)$$

$$\alpha_4 h'_5 - \beta_4 h'_3 + \gamma_4 Q'_4 = \sigma_4 \qquad (3.2.29)$$

$$\varepsilon_4 h'_5 + Q'_6 - Q'_4 = \lambda_4 \qquad (3.2.30)$$

Now if h'_1 is a boundary then Q'_2 can be expressed in terms of h'_3 ;

i.e., $Q'_2 + q_2 h'_3 = s_2$ (3.2.31)

in which $q_2 = \dfrac{\alpha_2}{\gamma_2}$, and $s_2 = \dfrac{\sigma_2 + \beta_2 h'_1}{\gamma_2}$;

If eq. (3.2.31) is now substituted into eq. (3.2.28) then h'_3 may be expressed in terms of Q'_4 ;

thus $h'_3 + p_3 Q'_4 = r_3$ (3.2.32)

in which $p_3 = \dfrac{1}{q_2+\varepsilon_2}$, and $r_3 = \dfrac{\lambda_2+s_2}{\varepsilon_2+q_2}$.

Eq. (3.2.32) may now be substituted into eq. (3.2.29) to give :

$$Q'_4 + q_4 h'_5 = s_4 \qquad (3.2.33)$$

in which $q_4 = \dfrac{\alpha_4}{\gamma_4+\beta_4 p_3}$, and $s_4 = \dfrac{\sigma_4 + \beta_4 r_3}{\gamma_4 + \beta_4 p_3}$.

The above substitutions give rise to two general equations which effectively reduce the system to an upper triangular matrix. These equations are :

$$h'_{2m+1} + p_{2m+1} Q'_{2m+2} = r_{2m+1} \qquad (3.2.34)$$

and $Q'_{2m+2} + q_{2m+2} h'_{2m+3} = s_{2m+2}$ (3.2.35)

In which $p_{2m+1} = \dfrac{1}{q_{2m} + \varepsilon_{2m}}$, $r_{2m+1} = \dfrac{\lambda_{2m} + s_{2m}}{\varepsilon_{2m} + q_{2m}}$, $q_{2m+2} = \dfrac{\alpha_{2m+2}}{\gamma_{2m+2}+\beta_{2m+2} \; p_{2m+1}}$

and $s_{2m+2} = \dfrac{\sigma_{2m+2} + \beta_{2m+2} r_{2m+1}}{\gamma_{2m+2} + \beta_{2m+2} p_{2m+1}}$ where $p_1 = 0$ and $r_1 = h'_1$.

The process is then continued until the boundary Q'_{2M} is reached ;

i.e., $h'_{2M-1} = -P_{2M-1}\ Q'_{2M} + r_{2M-1}$ (3.2.36)

Knowing Q'_{2M} it is possible to calculate h'_{2M-1}, and using this it is possible to obtain Q'_{2M-2} from (3.2.35) and so on back to h'_1.

As the finite difference equations are similar to the equations for the first implicit system then a stability analysis is not given.

The third implicit system to be suggested by Dronkers is one in which the Q's and h's are determined at the same node (see Fig. 3.8). This scheme then allows the use of different grid lengths and a more flexible schematization.

The basic dynamic equation used is :

$$\frac{\partial h}{\partial x} = \frac{-1}{Ag}\frac{\partial Q}{\partial t} - \frac{Q|Q|}{C^2A^2a} - \frac{B}{A^2g}\frac{Q}{b}\frac{\partial Q}{\partial x} \qquad (3.2.37)$$

and the continuity eq. is as eq. (3.2.20).

The finite difference equations become :

$$h'_{m+1} - h'_m = -\frac{\Delta x_m}{2Tg\ A_m}\left[(Q'_{m+1} - Q_{m+1})+(Q'_m - Q_m)\right] - \frac{\Delta x_m|Q_{m+1} + Q_m|}{4\ C_m^2\ A_m^2\ a_m}(Q'_{m+1} + Q'_m)$$

$$- \frac{B_m}{2Tg\ A_m^2\ b_m}(Q_{m+1} + Q_m)(Q'_{m+1} - Q'_m) \qquad (3.2.38)$$

$$\text{and}\quad Q'_{m+1} - Q'_m = \frac{-\Delta x_m}{2T}\ b_m\left[(h'_{m+1} - h_{m+1})+(h'_m - h_m)\right] \qquad (3.2.39)$$

again the primes are for the unknowns at t = t + T and variables without primes are the knowns at t = t.

As shown previously a general set of equations may be formed for solution by the Double Sweep Method. These are :

$$h'_{m+1} - h'_m + \eta_m\ Q'_{m+1} + \theta_m\ Q'_m = \mu_m \qquad (3.2.40)$$

$$\text{and}\quad \zeta_m\ (h'_{m+1} + h'_m) + Q'_{m+1} - Q'_m = \xi_m \qquad (3.2.41)$$

If at a boundary either h'_1 is specified or Q'_1 then it is possible to form two general equations similar to eq. (3.1.34) and eq. (3.1.35) these are;

$$Q'_{m-1} = -q_{m-1}\, h'_m - t_{m-1}\, Q'_m + s_{m-1} \qquad (3.2.42)$$

$$h'_m = -p_m\, Q'_m + r_m \qquad (3.2.43)$$

in which the recurrent formulae for the co-efficients are :

$$q_{m-1} = \frac{1}{p_{m-1} + \theta_{m-1}}, \quad t_{m-1} = \frac{\eta_{m-1}}{p_{m-1} + \theta_{m-1}}, \quad s_{m-1} = \frac{r_{m-1} + \mu_{m-1}}{p_{m-1} + \theta_{m-1}},$$

$$p_m = \frac{\sigma_{m-1}\, t_{m-1} + 1}{\sigma_{m-1}\, q_{m-1} + \zeta_{m-1}} \text{ and } r_m = \frac{\xi_{m-1} + \sigma_{m-1}\, s_{m-1} - \zeta_{m-1}\, r_{m-1}}{\sigma_{m-1}\, q_{m-1} + \zeta_{m-1}},$$

where $\sigma_{m-1} = p_{m-1}\, \zeta_{m-1} + 1$, $p_1 = 0$ and $r_1 = h'_1$.

This third implicit scheme may be extended to deal with river junctions in the following manner. Instead of elimination h'_1 and letting it equal r_1, it is carried through the computations to give the expressions :

$$Q'_{m-1} = -q_{m-1}\, h'_m - t_{m-1}\, Q'_m + s_{m-1} + b_{m-1}\, h'_1 \qquad (3.2.44)$$

and $h'_m = -p_m\, Q'_m + r_m + a_m\, h'_1$ (3.2.45)

where m is from 2 to k at the junction.

In the above :

$$a_m = \frac{\sigma_{m-1}\, b_{m-1} - \zeta_{m-1}\, a_{m-1}}{\sigma_{m-1}\, q_{m-1} + \zeta_{m-1}} \text{ and } b_{m-1} = \frac{a_{m-1}}{p_{m-1} + \theta_{m-1}} \qquad (3.2.45)$$

Also $a_1 = 1$ and $r_1 = 0$, so that at the junction :

$$p_k\, Q'_k = a_k\, h'_1 - h'_k + r_k \qquad (3.2.46)$$

Now, if the decomposition process is started at the junction and worked towards the boundary, keeping h'_k in the equations, then the following holds for two successive sections n and n-1 ;

$$Q'_n = -q_n{}^*\, h'_{n-1} - t_n{}^*\, Q'_{n-1} + s_n{}^* + b_n{}^*\, h'_k \qquad (3.2.47)$$

and $h'_{n-1} = -p^*_{n-1}\, Q'_{n-1} + r^*_{n-1} + a^*_{n-1}\, h'_k$ (3.2.48)

where $p^*_k = 0$ and $a^*_k = 1$. The co-efficients q^*_n, t^*_n, s^*_n, b^*_n, p^*_n, r^*_n and $a_{n-1}{}^*$ being determined in a similar manner q_m, t_m, s_m, b_m etc.

Thus for $m = 2$, eq. (3.2.48) becomes :

$$p_1^* \, Q_1' = -h_1' + a_1^* \, h_k' + r_1^* \qquad (3.2.49)$$

and so two equations are obtained for a particular reach relating the depths and flows at the ends, i.e., equations (3.2.46) and (3.2.49).

Thus for a particular network the system is broken up into single reaches and sweeps are made up and down each reach to obtain two formulae per reach, similar to equations (3.2.46) and 3.2.49). At a particular junction, if the Bernoulli term is neglected then the depth h_k is assumed constant over the junction and the algebraic sum of flow into the section is zero. At a boundary one value must be given so the system for the end conditions (including junctions) reduces to a set of simultaneous equations in which the number of unknowns equals the number of equations. Once this system is solved for the end conditions the intermediate information may then be filled in, in the normal manner.

The above method requires that the system be broken up into reaches and nodes numbered consecutively within each reach. Also, that separate procedures are required to handle the end conditions, which could become complicated if there are a large number of junctions. The method however, appears to have the advantage that very little extra storage is required over that which would be required if all the reaches were placed end to end.

The stability of the third implicit system is not presented.

To compare the various schemes the tidal propagation of the river Lek, one of the branches of the River Rhine was computed. The total length considered was 70 km. using section lengths of 5.6 km. except for the third implicit scheme where different sized section lengths were considered. The step size T was taken as 600 seconds for all schemes to ensure stability of the explicit scheme and accuracy of the results. This time step keptwithin the limit of 700 seconds derived from eq. (3.2.12) in which $b^* = 0$ at slack water. The boundary conditions were tide at the mouth and an upland discharge

Differences between the results of the explicit method and the observed were very small and differences between all methods and observed appeared to be within the accuracy of the observations which was quoted as 5% of the maximum discharges at a particular location.

3.3 Further Developments

Baltzer and Lai in their paper [4] compute three different methods to simulate unsteady flow in an open channel; these, as discussed in Chapter 2 are- the Power Series Method, the Method of Characteristics and an Implicit method. Each method gave good agreement with recorded data. To illustrate the behaviour of convergence they started each model off with different initial conditions that were considerably in error and were then able to show that each method produced results that rapidly converged to unique discharge curves owing to channel friction. If a channel has no friction whatsoever then each different initial condition would produce a different discharge curve.

Abbott [1] showed that the two-step Lax-Wendroff method suggested by Liggett and Woolhiser could be successfully applied to unsteady flow problems that even contained bores. His adaptation of the method uses the diffusing method to advance the solution to $t + \Delta t/2$ (the two dependent variables h and u being calculated at the same node) and then uses the leap frog method to determine the unknowns at time $t + \Delta t$. The difference between this application and Liggett and Woolhisers appears to be due to the estimate of the friction term in which Abbott takes an average at time $t + \Delta t/2$. Liggett and Woolhiser do not give any indication of how to estimate this term but as mentioned previously they say in their Closure [13] that inflexible schemes can be made useful by the proper averaging of this term. Abbott states that this method is fast, accurate and simple to program. In the chosen examples, which contained bores, the jump was spread over about four times the difference space step, with a slight but well damped oscillation in the solution behind it.

The methods ability to handle jumps without any special treatment is attributed to the $\partial u^2/\partial x^2$ term in the diffusing part of the method which acts as an additional damping force. Boundary conditions were supplied by the method of Characteristics.

Balloffet describes [3] how an adaption of the leap frog method may be used to deal with Bridge or Dam constrictions and Confluences. The difference method used is similar to Dronkers in which the dependent variables are staggered in space and time, using Q instead of u. Bridge or Dam constrictions and Confluences occur at Q nodes where the head loss is taken into account by a factor k, obtained from the U. S. Bureau of Public Roads, this is then added to the friction term. Confluences are solved by applying the dynamic finite difference equation in a forward/backward difference form to each of the branches and then by eliminating the elevation at the branch to produce a set of M - 1 equation, where M = number of branches. The M th equation is that the algebraic sum of the flow at the node is zero and so solutions are found for each branch discharge.

Other versions of the previously mentioned methods include a diffusing scheme, suggested by Strelkoff [19], in which the dependent variables are staggered in the space direction only and an implicit scheme which uses all the dependent variables on the t + Δt line except in evaluating the time derivative. To linearise the friction term he cleverly expands it by a Taylors series to give a truncation error of $O(\Delta t^2)$ and so is then able to express this at t + Δt in terms of its value at t.

Kamphuis [11] uses the third implicit scheme of Dronkers for a tidal study of the St. Lawrence River with a bifurcation. The method of solution of the simultaneous equations is basically the Double Sweep method. However, to handle the off-diagonal elements two extra routines were required to reduce the matrix to upper triangular form. This considerably increased the complexity of the method and was not applicable to a general network.

There appears to have been very little investigation into the use of non-linear methods. One reference is that of Amein and Fang [2], in which the authors set up the momentum and mass continuity equations as non-linear implicit functions, which they then solve by Newtons method.

The velocities and depths of flow were determined at the same grid points and a single channel only was considered. This gave rise to a system that was banded around the main diagonal which is a property, the authors said, could be used to advantage in solving the resulting set of linear equations.

In a practical example tried by the authors, in which channel properties changed significantly from section to section, the implicit method was the only one, of three tried, (i.e., implicit, explicit and the method of characteristics) in which no difficulty was experienced in obtaining a solution. In addition, extremely large time steps could be used, even ones as great as 20 hours, as demonstrated by the authors.

The latter, also state that most data for flow studies is given in 3 hr. to 6 hr. intervals so that smaller time steps than this will have no effect on accuracy. Time steps of this order can be handled by the implicit method tried, together with very economical computational times.

Shubinski, McCarty and Lindorf in their paper [16], discuss briefly the schematization and programming of the Sacramento-San Joaquin Delta of California's Central Valley, U.S.A. The total idealised network consisted of 572 junctions and 625 channels in which depths of flow were determined at the junctions and velocities at the channel half lengths. The finite difference scheme adapted was an explicit one in which solutions were first obtained at time $t + \Delta t/2$ by a modified Runge-Kutta procedure in which only two sub-intervals were used. The values obtained were then treated as the average throughout Δt and the computation was then continued for $t + \Delta t$. The authors do not give any detail of how the $\delta V/\delta x$ term is evaluated at junctions but it is assumed that an average is determined about the junction and this is then divided by an average length.

The results of the authors work showed that good results were obtained near the boundaries where they are under the boundaries strong influence. However, the farther away the stations were from the boundaries then less agreement was obtained. This was attributed to incorrect starting values and inadequate determination of additional inflow and outflow from junctions.

It is worthwhile to note that Brutsaert [7] verified the unsteady flow equations experimentally for a single straight channel.

CHAPTER FOUR

NUMERICAL SYSTEMS

4.0 Introduction

This section deals with the different finite difference methods used by the author to analyse the unsteady flow equations for channel networks.

First discussed is the generalised network layout and schematization of the channel geometry. Following this iseach of the finite difference methods and these are dealt with in detail.

Finally the framework of the basic program is presented and the purpose of each block in the program is discussed.

4.1 Network Layout and Schematization

A generalised channel system may be considered as a series of interconnecting nodes with any number of branches as shown in Fig. 4.1.

The staggered layout in the space direction shown in this figure is used in all methods. Solutions are not, however, staggered in the time direction but are determined at every time increment.

It can be seen from the figures that junctions occur at 'head' nodes, i.e., where the depth of flow is determined and that 'quantity' nodes, i.e., where the rate of flow is determined, occur along the channels. By applying this staggered system the aforementioned problems regarding boundaries and flow around junctions is alleviated. In the chosen system no two nodes of the same type may be next to each other. Also, each reach length Δx_k may be different and channel geometry needs only to be collected at quantity nodes.

The position of quantity nodes is chosen so that the channel geometry at that point is representative of the reach between its two end head nodes. Cross- sectional data may be stored as functions of the depth of flow, the particular function depending on the irregularity of the cross-section. The information required for computation is Ac and Bc the convective areas and breadths, A_T and B_T the total areas and breadths, R the hydraulic radius, which

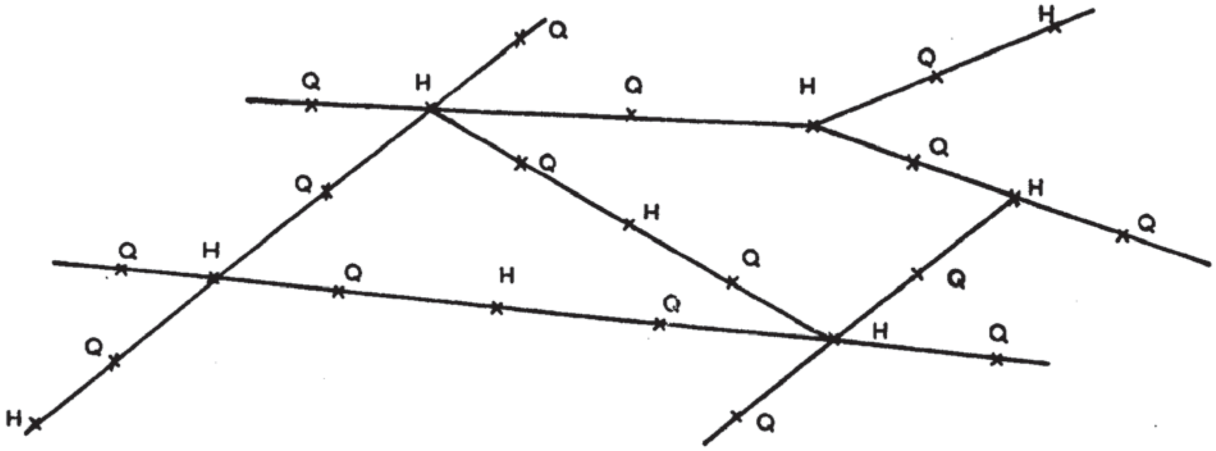


FIGURE 4.1

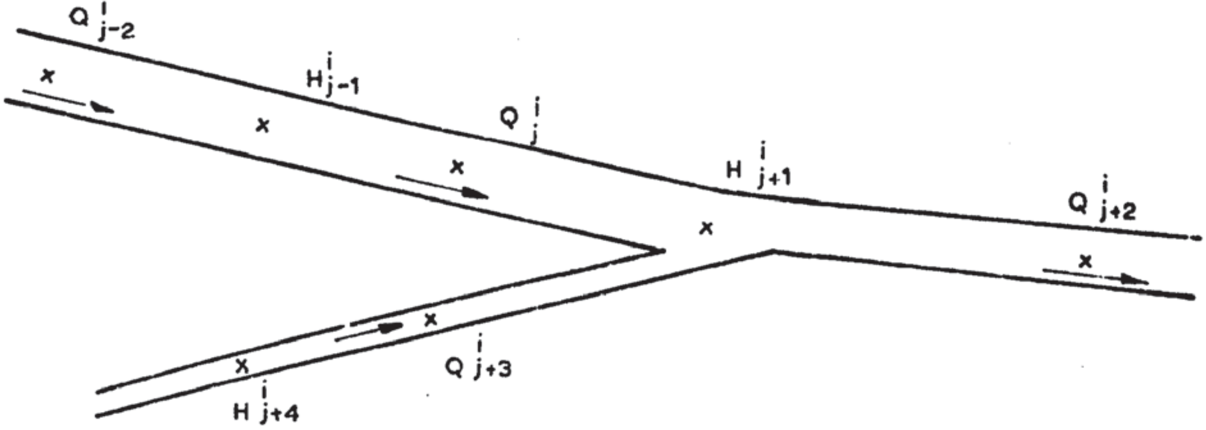


FIGURE 4.3

uses the convective area and perimeter, Chezy 'C' which may also be expressed as a function of the velocity of flow, and finally Δx which is the length along the centre line of the channel between two nodes. The only information that needs to be supplied at a head node is z, the height of its bottom above a common datum.

Data that is likely to alter during the running of a program is the boundary conditions and the sideflow function q. The former are identified by their node number, type, and value at a particular time. The sideflow function q may vary with locality, i.e., each reach and also with time.

4.2. First Explicit Method

This first explicit method utilizes central differences for the space derivative and forward differences for the time derivative. The basic dynamic equations used is as eq. (2.2.9) with $\frac{\partial h}{\partial x} + \frac{\partial z}{\partial x}$ replaced by $\frac{\partial H}{\partial x}$, i.e.,

$$0 = \frac{1}{g}\frac{\partial V}{\partial t} + \frac{V}{g}\frac{\partial V}{\partial x} + \frac{\partial H}{\partial x} + \frac{V|V|}{C^2R} + \frac{Vq}{Acg} \qquad (4.2.1.)$$

It is necessary to extend the equation of continuity to deal with a junction and this may be done in the following manner. It can be stated that the algebraic sum of flow into a node over a time interval Δt is equal to the net volume stored, and so for any particular node j with M branches :

$$\Delta H \sum_{m=1}^{M} B_{Tj+m}\, \Delta x_{k+m} = \sum_{m=1}^{M} \left(q_{k+m}\, \Delta x_{k+m} - Q_{j+m}\right) \Delta t \qquad (4.2.2)$$

The above equation implies the sign convention that channel flow leaving the head node j is positive and entering it is negative, and that sideflow entering is positive and leaving is negative. In the product $B_{Tj+m}\, \Delta x_{k+m}$, B_T is the total breadth at the Q_{j+m} th node connected by the k+m th channel to node j. This may be replaced by SA_{j+m}, a surface area function and may be determined independently of B_T as a function of h.

Eq. (4.2.2) then becomes :

$$\frac{\Delta H}{\Delta t} = \frac{\sum_{m=1}^{M} (q_{k+m}\,\Delta x_{k+m} - Q_{j+m})}{\sum_{m=1}^{M} SA_{j+m}} \tag{4.2.3}$$

Using the grid notation shown in Fig. 4.2 the finite difference form of eq. (4.2.1) for this method, becomes :

$$0 = \frac{(V_j^{i+1} - V_j^{i})}{g\,\Delta t} + \frac{V_j^i}{g}\frac{\partial V_j^i}{\partial x} + \frac{H_{j+1}^i - H_{j-1}^i}{\mathcal{K}_j} + \frac{V_j^i\,\bar{q}_j}{g\,Ac} + \frac{V_j^{i+1}|V_j^i|}{C^2R} \tag{4.2.4}$$

In which $\mathcal{K}_j = \Delta x_k + \Delta x_{k-1}$ and is the total distance between the two end head nodes for node Q_j.

Equation (4.2.3) for the node H_{j+1} in Fig. 3.2 becomes :

$$0 = \frac{H_{j+1}^{i+1} - H_{j+1}^{i}}{\Delta t} + \sum_{m=0,2,5} \frac{(Q_{j+m}^i - q_{k+m}^i\,\Delta x_{k+m})}{\sum_{m=0,2,5} SA_{j+m}^i} \tag{4.2.5}$$

in which the sum of the flows is the algebraic sum so that their direction has to be taken into account.

In eq. (4.2.4) Ac, C and R use the average of H_{j-1}^i and H_{j+1}^i, and $\bar{q}_j^{\,i}$ is the average of q_k^i and q_{k-1}^i. Q_j is $V_j\ Ac_j$. To evaluate the $\frac{\delta V}{\delta x}$ term in this equation for the node Q_{j-2} it is simply :

$$\frac{\delta V_{j-2}}{\delta x} = \frac{V_j^i - V_{j-4}^i}{\Delta x_{k-1} + \mathcal{K}_{j-2} + \Delta x_{k-4}} \tag{4.2.6}$$

To obtain the advance flow for the $\delta V/\delta x$ term say for Q_j^i then averages are taken of flows about node H_{j+1} (excluding Q_j^i) which are flowing in the same direction as Q_j^i, unless they are all apposing in which case an average of all the flows is taken (except Q_j^i). For instance if Q_j^i is flowing from

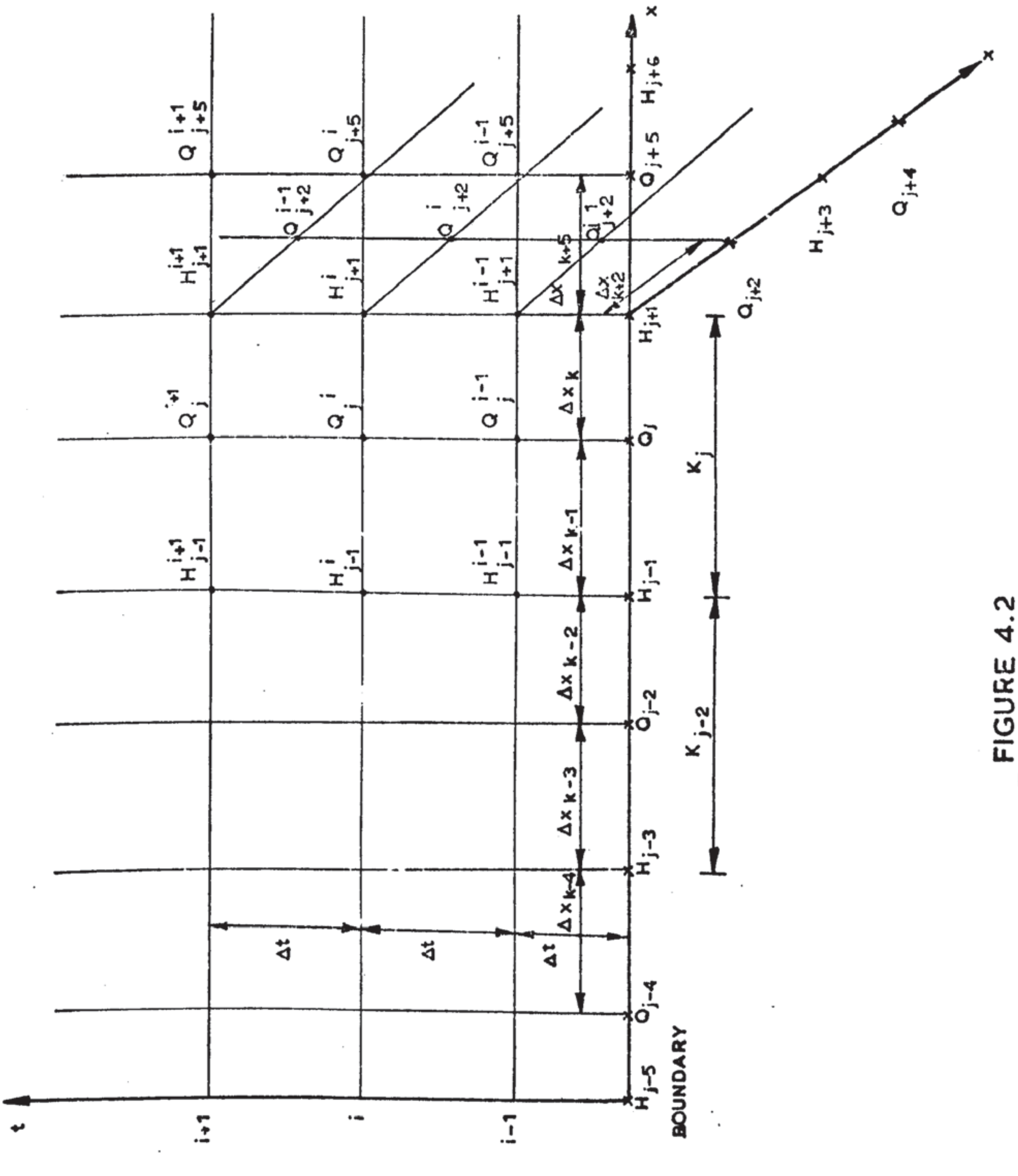

t
i+1
i
i-1
BOUNDARY
x
x
H_{j-5}
Q_{j-4}
H_{j-3}
Q_{j-2}
H_{j-1}
Q_j
H_{j+1}
Q_{j+5}
H_{j+6}
Q_{j+2}
H_{j+3}
Q_{j+4}
Δt
Δt
Δt
Δx_{k-4}
Δx_{k-3}
Δx_{k-2}
Δx_{k-1}
Δx_k
Δx_{k+5}
Δx_{k+2}
K_j
K_{j-2}
H^{i+1}_{j-1}
H^{i}_{j-1}
H^{i-1}_{j-1}
Q^{i+1}_{j}
Q^{i}_{j}
Q^{i-1}_{j}
H^{i+1}_{j+1}
H^{i}_{j+1}
H^{i-1}_{j+1}
Q^{i-1}_{j+2}
Q^{i}_{j+2}
Q^{i-1}_{j+2}
Q^{i+1}_{j+5}
Q^{i}_{j+5}
Q^{i-1}_{j+5}

FIGURE 4.2

H_{j-1} to H_{j+1}, Q^i_{j+5} is flowing from H_{j+1} to H_{j+6} and Q^i_{j+2} is flowing from H_{j+1} to H_{j+3} then :

$$\frac{\delta V^i_j}{\delta x} = \frac{\dfrac{V^i_{j+5} + V^i_{j+2}}{2} - V^i_{j-2}}{\dfrac{\Delta x_{k+5} + \Delta x_{k+2}}{2} + K_j + \Delta x_{k-2}} \tag{4.2.7}$$

If only one of the flows Q^i_{j+2} or Q^i_{j+5} is flowing in the same direction as Q^i_j, i.e., if say Q^i_{j+5} is flowing from H_{j+3} to H_{j+1} then :

$$\frac{\delta V^i_j}{\delta x} = \frac{V^i_{j+5} - V^i_{j-2}}{\Delta x_{k+5} + K_j + \Delta x_{k-2}} \tag{4.2.8}$$

If both Q^i_{j+2} and Q^i_{j+5} are flowing towards H_{j+1} then eq. (4.2.7) is used. The above procedure would be the same for the backward value in $\frac{\delta V}{\delta x}$ if there was a junction at H_{j-1}.

Earlier versions of this method simply took an average of the flows around a junction (again except the flow under consideration) irrespective of direction but it was felt that the above procedure would eliminate problems where the situation shown in Fig. 4.3 arose.

Now Q^i_{j+2} is positive and Q^i_{j+3} negative with respect to H^i_{j+1}, and so if averages were simply taken about H^i_{j+1}, $\delta V/\delta x$ would become :

$$\frac{\delta V^i_j}{\delta x} = \frac{\dfrac{V^i_{j+2} - V^i_{j+3}}{2} - V^i_{j-2}}{\delta x} \tag{4.2.9}$$

in which case V^i_{j+2} and V^i_{j+3} would tend to cancel each other out. If however, the procedure as explained earlier is adopted then $\delta V/\delta x$ becomes :

$$\frac{\delta V_j^{\,i}}{\delta x} = \frac{V_{j+2}^{\,i} - V_{j-2}^{\,i}}{\delta x} \tag{4.2.10}$$

which would appear to be a better estimate of this term which in any case is usually small compared with the others[9]and[11].

When a head controlled boundary is met with, as say H_{j-5} in Fig. 4.2 then a forward difference quotient is used to estimate the $\delta V/\delta x$ term. In this case the term becomes

$$\frac{\delta V_{j-4}^{\,i}}{\delta x} = \frac{V_{j-2}^{\,i} - V_{j-4}^{\,i}}{\Delta x_{k-3} + \Delta x_{k-4}} \tag{4.2.11}$$

It can be seen on examination of eq. (4.2.4) that the friction term is linearised as the velocity after thetime interval multiplied by the velocity before the time interval. Earlier models used the velocity before the time interval for both values but oscillations were evident in the solution similar to those described earlier by Liggett and Woolhiser.

4.3 Second Explicit Method

This method was developed so that some of the problems regarding the evaluation of the $\delta V/\delta x$ term in the dynamic equation may be avoided.

It is clear that the continuity equation (2.1.2) may be expressed in the following manner

$$B_T \frac{\partial H}{\partial t} + Ac \frac{\partial V}{\partial x} + V \frac{\partial Ac}{\partial x} = q \tag{4.3.1}$$

Now $\dfrac{\partial Ac}{\partial x} = \dfrac{dA}{dh}\dfrac{\partial h}{\partial x} = B_c \dfrac{\partial h}{\partial x}$

ahd so (4.3.1) becomes

$$\frac{\partial V}{\partial x} = \frac{q}{Ac} - \frac{B_T}{Ac}\frac{\partial H}{\partial t} - V\frac{Bc}{Ac}\frac{\partial h}{\partial x} \tag{4.3.2}$$

which on substitution into (4.2.1) gives

$$0 = \frac{1}{g}\frac{\partial V}{\partial t} - \frac{V\ B_T}{Ac\ g}\frac{\partial H}{\partial t} - \frac{V^2 Bc}{Ac\ g}\frac{\partial h}{\partial x} + \frac{\partial H}{\partial x} + \frac{V|V|}{C^2 R} + \frac{2Vq}{Ac\ g} \qquad (4.3.3)$$

The above equation together with eq. (4.2.3) are the two basic equations for this method. The procedure is to first sweep down the channel network to evaluate all H_j^{i+1} using quantities of flow at time i in accordance with eq. (4.2.5). A second pass is then made down the system to evaluate all Q_j^{i+1} using the following finite difference form of eq. (4.3.3), (see Fig.4.2).

$$0 = \frac{(V_j^{i+1} - V_j^i)}{g\ \Delta t} - \frac{(V_j^{i+1} + V_j^i)\ B_T}{4\ Ac\ g}\left[(H_{j-1}^{i+1} + H_{j+1}^{i+1}) - (H_{j-1}^{i} + H_{j+1}^{i})\right]$$

$$+ \frac{V_j^{i+1}|V_j^i|}{C^2R} - \frac{V_j^{i+1}\ V_j^i\ Bc}{Ac\ g\ 2k_j}\left[(h_{j+1}^{i+1} + h_{j+1}^{i}) - (h_{j-1}^{i+1} + h_{j-1}^{i})\right] + \frac{(V_j^{i+1} + V_j^i)\ \bar{q}_j}{Ac\ g}$$

$$+ \left[(H_{j+1}^{i+1} + H_{j+1}^{i}) - (H_{j-1}^{i+1} + H_{j-1}^{i})\right]\frac{1}{2\ k_j} \qquad (4.3.4)$$

and $Q_j^{i+1} = V_j^{i+1}$ Ac.

In eq. (4.3.4) the parameters Ac, Bc, B_T, C and R are evaluated using as an average depth, the value :

$$(H_{j+1}^{i+1} + H_{j-1}^{i+1} + H_{j+1}^{i} + H_{j-1}^{i})\ \frac{1}{4} \qquad (4.3.5)$$

and similarly $\bar{q}_j$ is $(q_k^{i+1} + q_{k-1}^{i+1} + q_k^i + q_{k-1}^i)\ \frac{1}{4}$ (4.3.6)

4.4 Implicit Method using Gauss-Seidel

The basic dynamic equation used is eq. (2.5.4), which will be repeated here :

$$0 = \frac{1}{Ac\ g}\frac{\partial Q}{\partial t} - \frac{Q}{Ac^2 g}(Bc+B_T)\frac{\partial H}{\partial t} - \frac{Q^2}{Ac^3 g}\frac{\partial A}{\partial x} + \frac{\partial H}{\partial x} + \frac{Q|Q|}{C^2 Ac^2 R} + \frac{2Qq}{Ac^2 g} \qquad (4.4.1)$$

The basic continuity equation is as eq. (4.2.3). For the system shown in Fig. 4.2 the finite difference form of these equations become :

$$0 = \frac{1}{Ac\ g\ \Delta t}(Q_j^{i+1} - Q_j^{i}) - \frac{(Q_j^{i} + Q_j^{i+1})(Bc + B_T)}{4Ac^2 g\ \Delta t}\Big[(H_{j-1}^{i+1} + H_{j+1}^{i+1}) -$$

$$(H_{j-1}^{i} + H_{j+1}^{i})\Big] - \frac{Q_j^{i+1}\ Q_j^{i}\ \delta A}{Ac^3 g\ Kj} + \Big[(H_{j+1}^{i+1} + H_{j+1}^{i}) - (H_{j-1}^{i+1} + H_{j-1}^{i})\Big]\frac{1}{2kj}$$

$$+ \frac{Q_j^{i+1}\ Q_j^{i}}{C^2 Ac^2 R} + \frac{(Q_j^{i+1} + Q_j^{i})}{Ac^2\ g}\bar{q}_j \qquad (4.4.2)$$

in which δA is the difference between the area at j using $\frac{(H_{j+1}^{i+1} + H_{j+1}^{i})}{2}$ and using $\frac{(H_{j-1}^{i+1} + H_{j-1}^{i})}{2}$, $\bar{q}$ is as eq. (4.3.6) and the parameters Bc, B_T, Ac, C and R are calculated using an average depth according to eq. (4.3.5). The continuity equation is

$$0 = \frac{H_{j+1}^{i+1} - H_{j+1}^{i}}{\Delta t} + \sum_{m=0,2,5} \frac{\{(Q_{j+m}^{i+1} + Q_{j+m}^{i}) - (q_{k+m}^{i+1} + q_{k+m}^{i})\ \Delta x_{k+m}\}}{\sum\limits_{m=0,2,5} (SA_{j+m}^{i+1} + SA_{j+m}^{i})} \qquad (4.4.3)$$

Equation (4.4.2) may be reduced to

$$(Q_j^{i+1} - Q_j^{i}) - \alpha\Big[(H_{j-1}^{i+1} + H_{j+1}^{i+1}) - (H_{j-1}^{i} + H_{j+1}^{i})\Big] - \eta\ Q_j^{i+1} + \mu\ Q_j^{i+1}$$

$$+ \beta\Big[(H_{j+1}^{i+1} + H_{j+1}^{i}) - (H_{j-1}^{i+1} + H_{j-1}^{i})\Big] + q(Q_j^{i+1} + Q_j^{i}) = 0 \qquad (4.4.4)$$

in which $\alpha = \frac{(Bc + B_T)(Q_j^{i} + Q_j^{i+1})}{4\ Ac}$, $\beta = \frac{gAc\ \Delta t}{2kj}$, $\eta = \frac{Q_j^{i}\ \delta A\ \Delta t}{Ac^2\ kj}$, $\mu = \frac{|Q_j^{i}|\ g\ \Delta t}{C^2 Ac\ R}$

and $q = \frac{\bar{q}_j\ \Delta t}{Ac}$. Rearranging (4.4.4) gives

$$Q_j^{i+1}(1 + \mu + q - \eta) - H_{j-1}^{i+1}(\alpha + \beta) - H_{j+1}^{i+1}(\alpha - \beta) =$$

$$Q_j^{i}(1 - q) - H_{j-1}^{i}(\alpha - \beta) - H_{j+1}^{i}(\alpha + \beta) \qquad (4.4.5)$$

If $(1 + \mu + q - \eta) = \lambda$, $\alpha + \beta = \gamma$, $\alpha - \beta = \sigma$ and $1 - q = \zeta$ then eq. (4.4.5) becomes

$$Q_j^{i+1} \lambda - H_{j-1}^{i+1} \gamma - H_{j+1}^{i+1} \sigma = Q_j^{\ i} \zeta - H_{j-1}^{\ i} \sigma - H_{j+1}^{\ i} \gamma \qquad (4.4.6)$$

Equation (4.4.3) may be expressed as

$$\sum_{m=0,2,5} Q_{j+m}^{i+1} + \frac{SA_{j+1}}{\Delta t} H_{j+1}^{i+1} = q_{j+1} \Delta x - \sum_{m=0,2,5} Q_{j+m}^{\ i} + \frac{SA_{j+1}}{\Delta t} H_{\ j+1}^{\ i} \qquad (4.4.7)$$

in which $SA_{j+m} = \sum_{m=0,2,5} (SA_{j+m}^{i+1} + SA_{j+m}^{\ i})$ and $q_{j+1}\Delta x = \sum_{m=0,2,5} (q_{j+m}^{i+1} + q_{j+m}^{\ i})\Delta x_{k+m}$

The right hand sides of equations (4.4.6) and (4.4.7) may now be expressed in terms the parameter B, in which $Bj = Q_j^{\ i} \zeta - H_{j-1}^{\ i} \sigma - H_{j+1}^{\ i} \gamma$

and $B_{j+1} = q_{j+1}\Delta x - \sum_{m=0,2,5} Q_{j+m}^{\ i} + \varepsilon_{j+1} H_{j+1}^{\ i}$ (4.4.8)

where $\varepsilon_{j+1} = \frac{SA_{j+1}}{\Delta t}$

Writing equations (4.4.6) and (4.4.7) in matrix form gives

$$\begin{bmatrix} -\gamma_{j-2} & \lambda_{j-2} & -\sigma_{j-2} & & & & & & & \\ & 1 & \varepsilon_{j-1} & 1 & & & & & & \\ & & -\gamma j & \lambda j & -\sigma j & & & & & \\ & & & 1 & \varepsilon_{j+1} & 1 & & & & 1 \\ & & & & -\gamma_{j+2} & \lambda_{j+2} & -\sigma_{j+2} & & & \\ & & & & & 1 & \varepsilon_{j+3} & 1 & & \\ & & & & & & & & & \\ & & & & & & & & & \end{bmatrix} \begin{bmatrix} Q_{j-2} \\ H_{j-1} \\ Q_j \\ H_{j+1} \\ Q_{j+2} \\ H_{j+3} \\ Q_{j+4} \\ Q_{j+5} \end{bmatrix}^{(i+1)} = \begin{bmatrix} B_{j-2} \\ B_{j-1} \\ B_j \\ B_{j+1} \\ Bj+2 \\ B_{j+3} \\ B_{j+4} \\ B_{j+5} \end{bmatrix} \qquad (4.4.9)$$

which gives the system

$$\bar{A} \ \bar{x} = \bar{B} \qquad (4.4.10)$$

where the matrix $\bar{A}$ and vector $\bar{B}$ also depend on the vector of unknowns $\bar{x}$.

<u>Gauss-Seidal</u>

In $\bar{A} \bar{x} = \bar{B}$ let $\bar{A} = \bar{D} - \bar{L} - \bar{U}$ in which $\bar{D}$ is a diagonal matrix whose elements are the diagonal elements of $\bar{A}$ and all other elements are zero, $\bar{L}$ and $\bar{U}$ are lower and upper triangular matrices respectively, with null diagonals.

The Gauss-Seidal iteration is then defined as

$$\bar{D}\, x^{(r+1)} = \bar{B} + \bar{U}\, x^{(r)} + \bar{L}x^{(r+1)} \qquad (4.4.11)$$

and therefore $x^{(r+1)} = (\overline{D - L})^{-1}\, \bar{U}\, x^{(r)} + (\overline{D - L})^{-1}\, \bar{B}$ (4.4.12)

the superscript r refers to values of x that were calculated on the last iteration and r+1 to values that are to be calculated in the next iteration.

In the implicit system above, the unknown Q_j^{i+1} is then determined from eq. (4.4.11) to be

$$Q_j^{i+1} = \frac{1}{\lambda j}\left[Bj + \gamma j\, H_{j-1}^{i+1} + \sigma j\, H_{j+1}^{i+1}\right] \qquad (4.4.13)$$

in which the (r+1) th value of H_{j-1}^{i+1} has been determined on the previous line. Also,

$$H_{j+1}^{i+1} = \frac{1}{\varepsilon j{+}1}\left[B_{j+1} + Q_j^{i+1} + Q_{j+2}^{i+1} + Q_{j+5}^{i+1}\right] \qquad (4.4.14)$$

where again, the (r+1) th value of Q_j^{i+1} has been determined onthe previous line, by eq. (4.4.13). Again it is emphasised that the channel flows are to be multiplied by +1 or -1 to take into account direction.

The iterative procedure is started off by first setting all H_j^{i+1} and Q_j^{i+1} equal to H_j^{i} and Q_j^{i} except at a boundary at which H_j^{i+1} and Q_j^{i+1} is set to its correct value at t + Δt and thus requires no iteration. The Gauss-Seidel procedure is then entered and new values of H_j^{i+1} and Q_j^{i+1} are calculated, using the most recent values of H_j^{i+1} and Q_j^{i+1} both for the actual calculation ie., equations (4.4.13) and (4.4.14) and in the evaluation of the co-efficients γ, λ, σ and ε. The iteration procedure is recycled until the difference between subsequent values of H_j^{i+1} and Q_j^{i+1} fall within a tolerance.

It can be seen that this method requires little storage and that junctions can be handled as easily as a straight channel. Also, the method does not require that the nodes are read into the computer consecutively but may be read in a random manner, thus displacing elements away from the basic tridiagonal structure.

4.5 Implicit Method using Double Sweep

This method uses the same basic equations and finite difference schemes as the previous section 4.4. The treatment of the finite difference equations (4.4.2) and (4.4.3) is identical up to and including eq. (4.4.10), the only difference is in the solution of this last equation.

As stated in section 3.2, it is possible to transform the resulting matrix $\bar{A}$ into upper triangular form by the Double Sweep method. In this application two vectors are sought, $\bar{K}$ and $\bar{M}$ which would reduce the system (4.4.9) to :

$$\begin{bmatrix} 1 \; - M_1 & & & & & & \\ & & & & & & \\ & 1 - M_{j-2} & & & & & \\ & & 1 - M_{j-1} & & & & \\ & & & 1 - M_j & & & \\ & & & & 1 - M_{j+1} & & \\ & & & & & & \\ & & & & & 1 - M_{N-1} & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \\ x_{j-2} \\ x_{j-1} \\ x_j \\ x_{j+1} \\ \\ \\ x_N \end{bmatrix} = \begin{bmatrix} K_1 \\ \\ K_{j-2} \\ K_{j-1} \\ K_j \\ K_{j+1} \\ \\ \\ K_N \end{bmatrix} \qquad (4.5.1)$$

i.e., $x_j = K_j + M_j \; x_{j+1}$ (4.5.2)

Once all M_j and K_j have been found then x_N can be determined, i.e., $x_N = K_N$, and so $x_{N-1} = K_{N-1} + M_{N-1} \; x_N$ can be found and so on back to x_1.

For a single channel in which the nodes are read into the computer consecutively then this is no problem and the recursive relationships become

$$K_j = \frac{B_j - a_j \; K_{j-1}}{b_j + a_j \; M_{j-1}} \qquad (4.5.3)$$

$$\text{and } M_j = \frac{-C_j}{b_j + a_j \; M_{j-1}} \qquad (4.5.4)$$

in which $a_j = -\gamma_j$, $b_j = \lambda_j$ and $C_j = -\sigma_j$ for a quantity node and $a_j = C_j = 1$

and $b_j = \varepsilon_j$ for a head node. If however, the nodes are not read into the computer consecutively or junctions exist, then off diagonal elements occur such as in the system (4.4.9). In this case the off diagonal elements multiplied by their latest unknown values, i.e., H_j^{i+1} or Q_j^{i+1} and are then taken over to the right hand side to join with $\bar{B}$. For the node H_{j+1} in Fig. 4.2, eq. (4.5.3) becomes

$$K_{j+1} = \frac{(B_{j+1} - Q_{j+5}^{i+1}) - K_j}{\varepsilon_{j+1} + M_j} \qquad (4.5.5)$$

In general this equation is

$$K_j = \frac{(B_j - \mathrm{CONST}) - a_j\, K_{j-1}}{b_j + a_j\, M_{j1}} \qquad (4.5.6)$$

If nodes are not read into the computer consecutively then some of the parameters a_j or C_j may be zero as they apply only to a tridiagonal system. The parameter CONST in eq. (4.5.6) is equal to the sum of the off diagonal elements multiplied by their latest values in x. The equation (4.5.4) applies to all cases.

The iterative process is thus; all x_j^{i+1} are set to x_j^{i}, except for the boundaries as described previously and the matrix $\bar{A}$ and vector $\bar{B}$ are calculated accordingly.

The vectors $\bar{K}$ and $\bar{M}$ are determined on the downward sweep using equations (4.5.4) and (4.5.6), and then all x_j^{i+1} are determined on the upward sweep. $\bar{A}$ and $\bar{B}$ are then updated and the process is repeated until differences between calculated values of x_j^{i+1} fall within a tolerance. At boundaries, B_j is set to the correct value of x_j^{i+1} and b_j is set to unity with $a_j = C_j = 0$.

4.6 Implicit Method using a Sparse Matrix Technique

This method is again identical to sections 4.4 and 4.5 up to the solution of the system (4.4.9). The method of inversion of the matrix A uses a program written by A. Jennings [10] and solves the matrix by a compact elimination technique specifically designed for sparse matrices. The matrix $\bar{A}$ is read as a one dimensional array in which the last element of a row is the right hand side B_j. To do this, procedures were written that presented the elements of the system in the order required. Each row is preceded by minus its row number and following this is the column number of an element and then its value. The output vector x overwrites the matrix A. The iterative procedure for this method is as section 4.5.

4.7 Nonlinear Method

The following forms the basis of the solution of the implicit finite difference equations by nonlinear procedures. The method itself was abandoned, early in the obtaining of results, in favour of the sparse nonlinear methods which will be described in the next section.

Slight modifications to eq. (4.4.2) gives rise to the following finite difference equation

$$(Q_j^{i+1} - Q_j^{i})-(Q_j^{i} + Q_j^{i+1})\,\frac{(Bc + B_T)}{4\ Ac}\left[(H_{j-1}^{i+1} + H_{j+1}^{i+1})-(H_{j-1}^{i} + H_{j+1}^{i})\right] -$$

$$(Q_j^{i} + Q_j^{i+1})^2\,\frac{\delta A\ \Delta t}{4\ k_j\ Ac^2} + \left[(H_{j+1}^{i+1} + H_{j+1}^{i})-(H_{j-1}^{i+1} + H_{j-1}^{i})\right]\frac{g\ Ac\ \Delta t}{2\ kj}$$

$$+ \frac{(Q_j^{i} + Q_j^{i+1})|Q_j^{i} + Q_j^{i+1}|g\ \Delta t}{4\ C^2 Ac\ R} + (Q_j^{i+1} + Q_j^{i})\,\frac{\bar{q}_j\ \Delta t}{Ac} = 0 \qquad (4.7.1)$$

On comparison with eq. (4.4.2) it can be seen that the only differences be in the nonlinear friction and ∂A/∂x terms and that the equation has been multiplied throughout by g Ac Δt. Equation (4.4.3), on multiplication by Δt, gives

$$(H_{j+1}^{i+1} - H_{j+1}^{i}) + \frac{\sum\limits_{m=0,2,5}\{\,(Q_{j+m}^{i+1} + Q_{j+m}^{i})-(q_{k+m}^{i+1} + q_{k+m}^{i})\ \Delta x_{k+m}\}\ \Delta t}{\sum\limits_{m=0,2,5}(SA_{j+m}^{i+1} + SA_{j+m}^{i})} = 0 \qquad (4.7.2)$$

Equation (4.7.1) is a nonlinear function, designated f_j, of the unknown variables Q_j^{i+1}, H_{j-1}^{i+1} and H_{j+1}^{i+1} and eq. (4.7.2) is another nonlinear function,

f_{j+1}, of the unknown variables Q_{j+m}^{i+1} and H_{j+1}^{i+1} where the number of Q variables depends on the number of connecting channels to H_{j+1}. Solutions to the above equations are found when the functions f_j and f_{j+1} either equal zero or are sufficiently near to it, depending on the accuracy required.

If the above system is considered as a general system such that :

$$
\begin{aligned}
f_1\ (x_1,\ x_2,\ \text{-------}\ x_j\ \text{----------}\ x_n) &= 0\\
f_2\ (x_1,\ x_2,\ \text{-------}\ x_j\ \text{----------}\ x_n) &= 0\\
f_j\ (x_1,\ x_2,\ \text{-------}\ x_j\ \text{----------}\ x_n) &= 0\\
f_n\ (x_1,\ x_2,\ \text{-------}\ x_j\ \text{----------}\ x_n) &= 0
\end{aligned}
\qquad (4.7.3)
$$

written concisely as

$$\bar{f}\ (\bar{x})\ =0 \qquad (4.7.4)$$

in which x is the column vector of the unknown variables $Q_j^{\ i+1}$ and $H_j^{\ i+1}$ and $\bar{f}$ the column vector of the functions f_j. If each of the functions f_j in (4.7.3) are now expanded by a Taylors series about the points x_1, x_2,-- x_j,--- x_n, then the following arises if terms of second and higher order are ignored

$$
\begin{aligned}
f_1(x_1+h_1,\text{---}x_j+h_j,\text{---}x_n+h_n) &= f_1(x_1,\text{--}x_j\text{--}x_n) + h_1\frac{\partial f_1}{\partial x_1} + \text{--}h_j\frac{\partial f_1}{\partial x_j} + \text{--}h_n\frac{\partial f_1}{\partial x_n}\\
f_j(x_1+h_1,\text{---}x_j+h_j\text{---}x_n+h_n) &= f_j(x_1,\text{--}x_j\text{--}x_n) + h_1\frac{\partial f_j}{\partial x_1} + \text{--}h_j\frac{\partial f_j}{\partial x_j} + \text{--}h_n\frac{\partial f_j}{\partial x_n}\\
f_n(x_1+h_1,\text{--}x_j+h_j\text{--}x_n+h_n) &= f_n(x_1,\text{--}x_j\text{--}x_n) + h_1\frac{\partial f_n}{\partial x_1} + \text{--}h_j\frac{\partial f_n}{\partial x_j} + \text{--}h_n\frac{\partial f_n}{\partial x_n}
\end{aligned}
\qquad (4.7.5)
$$

or in vector form

$$
\begin{bmatrix} f_1\\ f_2\\ f_j\\ f_n \end{bmatrix}_{(x+h)}
=
\begin{bmatrix} f_1\\ f_2\\ f_j\\ f_n \end{bmatrix}_{(x)}
+
\begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \text{-----} & \frac{\partial f_1}{\partial x_j} & \text{-----} & \frac{\partial f_1}{\partial x_n}\\
\frac{\partial f_j}{\partial x_1} & \text{-----} & \frac{\partial f_j}{\partial x_j} & \text{-----} & \frac{\partial f_j}{\partial x_n}\\
\frac{\partial f_n}{\partial x_1} & \text{-----} & \frac{\partial f_n}{\partial x_j} & \text{-----} & \frac{\partial f_n}{\partial x_n}
\end{bmatrix}
\begin{bmatrix} h_1\\ h_2\\ h_j\\ h_n \end{bmatrix}
\qquad (4.7.6)
$$

The n x n matrix is the Jacobian $\bar{J}$, so the above can be written as

$$\bar{f}\,(\bar{x} + \bar{h}) = \bar{f}(\bar{x}) + \bar{J}\,(\bar{x})\,\bar{h} \qquad (4.7.7)$$

in which $\bar{h}$ is the vector of differences.

Solutions are sought such that $\bar{f}(\bar{x}+\bar{h}) = 0$, at which eq. (4.7.7) becomes

$$\bar{h} = -\,\bar{J}(\bar{x})^{-1}\,\bar{f}(\bar{x}) \qquad (4.7.8)$$

Now $\bar{h} = \bar{x}_{i+1} - \bar{x}_i$, in which $\bar{x}_i$ is the vector of unknowns obtained from the last iteration, so that

$$\bar{x}_{i+1} = \bar{x}_i - \bar{J}(\bar{x}_i)^{-1}\,\bar{f}(\bar{x}_i) \qquad (4.7.9)$$

and is Newtons n - dimensional method.

Newtons method as such suffers from two serious disadvantages from the point of view of practical calculation. The first of which is that unless a sufficiently good initial estimate of the solutions are known then the method shown in eq. (4.7.9) may fail to converge. The second disadvantage is that in some cases the Jacobian $\bar{J}$ may be difficult to estimate if the functions are complex and also once it has been estimated it needs to be inverted.

Brayden [5] describes a class of methods in which the partial derivatives are not estimated or evaluated directly, but corrections to an approximate inverse of the Jacobian matrix are computed from values of the vector function $\bar{f}$.

If a vector $\bar{p}$ is defined as :

$$\bar{p} = -\,\bar{B}^{-1}\,\bar{f} \qquad (4.7.10)$$

where $\bar{B}$ is an estimate for the Jacobian $\bar{J}$, then Newtons method can be written as

$$\bar{x}_{i+1} = \bar{x}_i + \bar{p}_i \qquad (4.7.11)$$

Convergence will only occur if we are close enough to the solution so a simple modification to (4.7.11) gives

$$\bar{x}_{i+1} = \bar{x}_i + t_i\,\bar{p}_i \qquad (4.7.12)$$

where t_i is a value between one and zero, chosen to prevent the process diverging.

This parameter t is not to be confused with t for time. If the variable $\bar{x}$ is now defined as

$$\bar{x} = \bar{x}_i + t\,\bar{p}_i \tag{4.7.13}$$

where $\bar{x}_i$ and $\bar{p}_i$ will have particular values and t is a variable quantity. The vector $\bar{f}$ will now be functions of the variable t and Broyden shows that it is possible to use these functions to obtain an estimate of the Jacobian. On differentiating each function with respect to t gives :

$$\frac{df_j}{dt} = \sum_{k=1}^{n} \frac{\partial f_j}{\partial x_k} \cdot \frac{dx_k}{dt}, \quad \text{for } j = 1\text{---}n \tag{4.7.14}$$

and so $\dfrac{d\bar{f}}{dt} = \bar{J}\,\bar{p}_i$ (4.7.15)

from eq. (4.7.13).

It is now required to obtain an approximation to $\bar{J}$ at the point $\bar{x}_{i+1}$, so if $\bar{f}_j$ is expanded as a Taylors series about the point t_i,

$$f_j\ (t_i - s) = f_j(t_i) - s\,\frac{df_j}{dt}(t_i) + O(s^2) \tag{4.7.16}$$

and since $\bar{f}_{i+1} = \bar{f}(\bar{x}_{i+1}) = \bar{f}(\bar{x}_i + t_i\,\bar{p}_i) = \bar{f}(t_i)$ in which the vector $\bar{f}$ are functions of t alone, then

$$\bar{f}\ (t_i - s) \simeq \bar{f}_{i+1} - s\,\frac{d\bar{f}}{dt} \tag{4.7.17}$$

if second and higher order terms are neglected. On substitution of eq.(4.7.15) into eq. (4.7.17) we obtain

$$\bar{f}\ (t_i - s) \simeq \bar{f}_{i+1} - s\,\bar{J}\,\bar{p}_i \tag{4.7.18}$$

Broyden now uses eq. (4.7.18) to find a better estimate to the approximate Jacobian B, by chosing B_{i+1} to satisfy the equation :

$$\bar{f}\ (t_i - s) = \bar{f}_{i+1} - s_i\,\bar{B}_{i+1}\,\bar{p}_i \tag{4.7.19}$$

where s_i is a particular value of s chosen at each iteration to minimise the error of the estimate of $\dfrac{d\bar{f}}{dt}$. Broyden stated that it was his philosophy to find an estimate of the inverse Jacobian and not the Jacobian itself. If we now define an estimate to the inverse Jacobian $\bar{H}$, say, as

$$\bar{H}_i = -\bar{B}_i^{-1} \tag{4.7.20}$$

and also define a vector $\bar{y}_i$, such that

$$\bar{y}_i = \bar{f}_{i+1} - \bar{f}\,(t_i - s) \tag{4.7.21}$$

so that eq. (4.7.19) becomes

$$\bar{y}_i = s_i\,\bar{B}_{i+1}\,\bar{p}_i \tag{4.7.22}$$

If eq. (4.7.20) is substituted into equations (4.7.10) and (4.7.22) we obtain

$$\bar{p}_i = \bar{H}_i\,\bar{f}_i \tag{4.7.23}$$

and

$$\bar{H}_{i+1}\,\bar{y}_i = -s_i\,\bar{p}_i \tag{4.7.24}$$

the two above equations define a class of methods, based upon Newtons method, for solving nonlinear algebraic equations. Different methods arise when different assumptions are made on $\bar{H}_{i+1}$ or $\bar{B}_{i+1}$, t_i and s_i.

Broyden concludes that one method is superior to the others and he calls this the "full-step" reducing variation. In this t_i is chosen to reduce the norm of the residuals of $\bar{f}(\bar{x}_i)$ and s_i is put equal to t_i. It remains to place restrictions on $\bar{B}_{i+1}$ so that it may be defined uniquely in eq.(4.7.22). As no information is available about the change in $\bar{f}$ when $\bar{x}$ is changed in a direction other than $\bar{p}_i$, $\bar{B}_{i+1}$ is chosen so that the change in $\bar{f}$ predicted by $\bar{B}_{i+1}$ in a direction $\bar{q}_i$ orthogonal to p_i is the same as would be predicted by Bi. Symbolically, that is

$$\bar{B}_{i+1}\;\bar{q}_i = \bar{B}_i\;\bar{q}_i \tag{4.7.25}$$

where $(\bar{q}_i)^T\;\bar{p}_i = 0$.

This with eq.(4.7.22) is sufficient to define $\bar{B}_{i+1}$ uniquely as

$$\bar{B}_{i+1} = \bar{B}_i + (\bar{y}_i - s_i\,\bar{B}_i\,\bar{p}_i)\frac{p_i^T}{s_i\,p_i^T\,\bar{p}_i} \tag{4.7.26}$$

It is now possible to use Householders formula [5] to express $\bar{H}_{i+1}$ the inverse of $\bar{B}_{i+1}$, in the same terms as above giving :

$$\bar{H}_{i+1} = \bar{H}_i - \frac{(s_i\,\bar{p}_i + \bar{H}_i\,\bar{y}_i)\;\bar{p}_i^{\,T}\,\bar{H}_i}{\bar{p}_i^{\,T}\,\bar{H}_i\,\bar{y}_i} \tag{4.7.27}$$

which the "full-step" version gives

$$\bar{H}_{i+1} = \bar{H}_i - \frac{(\bar{H}_i \bar{y}_i + t_i \bar{p}_i) \bar{p}_i^T \bar{H}_i}{\bar{p}_i^T \bar{H}_i \bar{y}_i} \tag{4.7.28}$$

The algorithm for Broydens full-step method is :

1. Obtain an initial estimate X_o of the solution.
2. Obtain an initial estimate of the negative inverse Jacobian H..
3. Compute $\bar{f}_o = \bar{f}(\bar{X}_o)$.
4. Compute $\bar{p}_i = \bar{H}_i \bar{f}_i$.
5. Select a value t_i such that the norm of $\bar{f}(\bar{x}_i + t_i \bar{p}_i)$ is less than the norm of $\bar{f}(\bar{x}_i)$. In the course of this calculation $\bar{x}_{i+1} = \bar{x}_i + t_i \bar{p}_i$ and $\bar{f}_{i+1} = \bar{f}(\bar{x}_{i+1})$ will have been computed.
6. Test $\bar{f}_{i+1}$ for convergence, if not then
7. Compute $\bar{y}_i = \bar{f}_{i+1} - \bar{f}_i$
8. Compute $\bar{H}_{i+1}$ using eq. (4.7.28).
9. Go to step 4.

Each of the steps in the above algorithm will now be discussed in detail with respect to its application to the solution of the finite difference equations (4.7.1) and (4.7.2).

(i) All the unknown depths and quantities of flow at the advance time step, i.e., H_j^{i+1} and Q_j^{i+1} are set to the values obtained at the end of the last time step, i.e., H_j^i and Q_j^i, except at the boundaries at which H_j^{i+1} and Q_j^{i+1} are set to their correct values.

(ii) The initial estimate of the matrix $\bar{H}$ is obtained by differentiating the equations (4.7.1) and (4.7.2) with respect to each of the unknown variables. It was necessary to treat A_c in eq. (4.7.1) as a function of the unknowns H_{j-1}^{i+1} and H_{j+1}^{i+1}, the partial differentials for this equation become:

$$\frac{\partial f_j}{\partial Q_j^{i+1}} = 1 - \frac{(Bc + B_T)}{4Ac} \left[(H_{j-1}^{i+1} + H_{j+1}^{i+1}) - (H_{j-1}^i + H_{j+1}^i) \right] + (Q_j^i + Q_j^{i+1}) \frac{g\Delta t\ P}{2C^2 Ac^2}$$

$$- \frac{(Q_j^{i+1} + Q_j^i)\ \delta A\ \Delta t}{2\ Ac^2\ k_j} + \frac{\bar{q}_j\ \Delta t}{Ac} \qquad (4.7.29$$

and

$$\frac{\partial f_j}{\partial H_{j\pm1}^{i+1}} = -\frac{(Bc + B_T)}{4\ Ac}(Q_j^{\ i} + Q_j^{i+1})\left[1 - \frac{1}{Ac}\frac{dAc}{dH_{j\pm1}}\right] \pm \frac{g\ Ac\ \Delta t}{2\ k_j}$$

$$- \frac{(Q_j^i + Q_j^{i+1})\ |Q_j^i + Q_j^{i+1}|\ gP\Delta t}{2\ C^2Ac^3}\frac{dAc}{dH_{j\pm1}^{i+1}} + \left[(H_{j+1}^{i+1} + H_{j+1}^{\ i})-(H_{j-1}^{i+1} + H_{j-1}^{\ i})\right]\frac{g\Delta t\ dAc}{2k_j\ dH_{j\pm1}^{i+1}}$$

$$+ (Q_j^{i+1} + Q_j^i)^2\frac{\delta A\ \Delta t}{2Ac^3k_j}\frac{dAc}{dH_{j\pm1}^{i+1}} - \bar{q}_j(Q_j^{i+1} + Q_j^{\ i})\frac{\Delta t}{Ac^2}\frac{dAc}{dH_{j\pm1}^{i+1}} \qquad (4.7.30)$$

In the above it will be noticed that R is replaced by Ac/P, where P is not considered to be a function of $H_{j\pm1}^{i+1}$. Ac is, for the purpose of the above analysis, considered to be equal to $\frac{Bc}{4}$ $(H_{j+1}^{i+1} + H_{j-1}^{i+1} + H_{j+1}^{\ i} + H_{j-1}^{\ i})$, so that :

$$\frac{d\ Ac}{d\ H_{j\pm1}^{i+1}} = \frac{Bc}{4} \qquad (4.7.31)$$

The inclusion of eq. (3.7.31) into eq. (3.7.30) gives

$$\frac{\partial f_j}{\partial H_{j\pm1}^{i+1}} = -\frac{(Bc + B_T)}{4\ Ac}(Q_j^i + Q_j^{i+1})\left[1 - \frac{Bc}{4Ac}\right] - \frac{(Q_j^i + Q_j^{i+1})|Q_j^i + Q_j^{i+1}|\ g\Delta t\ Bc}{8\ C^2Ac^2R}$$

$$\pm \frac{gAc\Delta t}{2\ k_j} + \frac{g\Delta t\ Bc}{8\ k_j}\left[(H_{j+1}^{i+1} + H_{j+1}^{\ i})-(H_{j-1}^{i+1} + H_{j-1}^{\ i})\right] + (Q_j^{i+1} + Q_j^i)^2\frac{\delta A\ \Delta t\ Bc}{8\ Ac^3\ k_j}$$

$$(4.7.32)$$

The differentiation of eq.(4.7.2) with respect to the unknowns is :

$$\frac{\partial f_{j+1}}{\partial H_{j+1}^{i+1}} = 1 \qquad (4.7.33)$$

and $$\frac{\partial f_{j+1}}{\partial Q_{j+m}^{i+1}} = \pm\frac{\Delta t}{SA_{j+1}} \qquad (4.7.34)$$

in which SA_{j+1} is not, for these purposes only, treated as a function of H_{j+1}^{i+1}.

The reciprocal of minus the remaining values was then found and these were inserted into the $\bar{H}$ matrix in their appropriate places to give the initial estimate.

In the early stages of adopting this method a negative unit matrix was used, which, as can be seen on examination, proved to be a very good guess as far as equations (4.7.33) and (4.7.34) were concerned. However, the numerical value of the off diagonal partial derivatives eq. (4.7.32) are in general much higher than the diagonal elements eq. (4.7.29) which was also higher than one. This resulted in an estimate of $\bar{H}$ not being sufficiently close to an estimate that would produce convergence and so the method failed by not locating a lower norm. As there was a need to estimate off diagonal elements the above analytical estimates were resorted to and these proved more successful. Boundaries were dealt with by setting their diagonal elements in $\bar{H}$ equal to one and their off diagonal elements equal to zero, also the value of the function f_j at a boundary was set to zero. This process ensured that the boundary values once obtained in (i) were not affected by the iteration process.

(iii) The vector of residuals $\bar{f}_i$ is estimated using the values shown in 1. In the procedures which calculate these values the elements of $\bar{H}$ are also calculated.

(iv) $\bar{P}_i$ is calculated as 4.

(v) The norm chosen as a measure of the size of the vector of residuals $\bar{f}$ was the square of the Euclidean Norm, i.e., $\sum_{j=1}^{n} f_j^{\,2}$, where n is the total number of nodes. For each value of t_i chosen then $\bar{x}_{i+1}$ is tentively set to $\bar{x}_i + t_i\ \bar{p}_i$ and the norm of $\bar{f}(\bar{x}_{i+1})$ is evaluated and tested to see if it is less than the norm of $\bar{f}(\bar{x}_i)$. If this was so then $\bar{x}_{i+1}$ and $\bar{f}(\bar{x}_{i+1})$ are the values calculated, if not then another t_i is chosen until the inequality is satisfied. The norm reducing policy which proved most efficient was that t_i was first set to one, i.e., the full Newtonian step, t_2 was then calculated to be equal to

$$t_2 = \frac{(1 + 6\theta)^{\frac{1}{2}} - 1}{3\theta} \qquad (4.7.35)$$

where $\theta = \phi(1) / \phi(o)$, $\phi(t)$ being the square of the Euclidean norm of $\bar{f}(t)$. Equation (4.7.35) assumes that $\phi(t)$ is a quadratic function of $\phi(o)$ and a cubic function of $\phi(1)$. The resulting function is then differentiated to obtain the minimum from which t_2 is then chosen to lie between $0 < t_2 < 1$. If $\phi(t_2)$ is not less than $\phi(o)$ then t_3 is chosen to be 0.1, t_4 to be 0.01 and so on until 10^{-9}, after which the $\bar{H}$ matrix was re-estimated according to (ii) using the most up to date values of H_j^{i+1} and Q_j^{i+1}, $\bar{P}_i$ was then calculated as in (4) and the norm reducing process re-entered. If a t_i reached 10^{-9} on this occasion then $\bar{p}_i$ was set to $-\bar{p}_i$ and the norm reducing process was again re-entered. By this means a lower norm was usually located and the occasional resetting of the $\bar{H}$ matrix in this manner actually speeded up the convergence of the problem. The actual setting of $\bar{p}_i$ to $-\bar{p}_i$ was only necessary with this particular nonlinear method and was not found necessary with the following sparse nonlinear methods. Steps (vi) to (ix) are as steps (6) to (7) in the algorithm.

4.8 Sparse Non-Linear Method

The major disadvantage with the previous nonlinear method is the necessity to store the full $\bar{H}$ matrix, which could present problems if large systems are to be modelled. Another disadvantage, equally important as the first as far as convergence is concerned, is in the initial approximation of the $\bar{H}$ matrix, as even by estimating the elements analytically and then finding their reciprocal, this was still not sufficient to provide convergence with large time steps. *Broyden suggested a variation of his method described in section 4.7 which is directly applicable to solving systems of nonlinear equations where the Jacobian is sparse. In this method the inverse is not updated, as an inverse to a sparse matrix is not generally sparse, but computations are based on an approximation to the Jacobian itself. Broyden proves that corrections to this approximation are also sparse. Finally, a comparison is made between this

*ref [6].

method and Newtons for a system of 600 sparse equations. Both methods handled the mildly nonlinear problems well, with Broydens sparse method coming out slightly better in that it required less computational work but more iterations.

It is evident from the above that a sparse matrix inversion procedure is required to evaluate the vector $\bar{p}_i$; The new algorithm, using the same notation as in the previous section, is then :

1. Obtain an initial estimate, $\bar{x}_o$, of the solution.
2. Obtain an initial estimate of the Jacobian $\bar{B}$.
3. Compute $\bar{f}_o = \bar{f}_o\ (\bar{x}_o)$.
4. Compute $\bar{p}_i = -\bar{B}^{-1}\ \bar{f}_i$ using a sparse matrix routine.
5. Compute $\bar{x}_{i+1} = \bar{x}_i + t_i\ \bar{p}_i$ and $\bar{f}_{i+1} = \bar{f}(\bar{x}_i + t_i\ \bar{p}_i)$. Chosing the value of t_i such that $\phi(\bar{x}_{i+1}) < \phi(\bar{x}_i)$, as described in the previous section.
6. Test $\phi(\bar{x}_{i+1})$ for convergence.
7. Compute $\bar{y}_i = \bar{f}_{i+1} - \bar{f}_i$.
8. Compute B_{i+1}, denoted by B_1, where B_1 is

$$\bar{B}_1 = \bar{B} - \sum_{j=1}^{n} \bar{U}_j\ \bar{U}_j^T\ (\bar{B}\ \bar{p} - \bar{y}\ t^{-1})\ \frac{\bar{p}_j^T}{\bar{p}_j^T \bar{p}_j} \tag{4.8.1}$$

and the matrices and vectors without the subscript 1 have values obtained from the vector $\bar{x}_i$.

9. Go to 4.

In step (8) $\bar{p}_j = \bar{S}_j\ \bar{p}$, where S_j is a set of diagonal matrices j=1---n, where the kth diagonal element of Sj is zero if it is known that $\partial f_j / \partial x_k = 0$ and unity otherwise, Uj is the jth column of the unit matrix of order n, where n is the size of the matrix.

On examination of eq. (4.8.1) it can be seen thatthis equation follows from eq. (4.7.26) with $s_i = t_i$, except that the sparseness of the matrix $\bar{B}_{i+1}$ is preserved and that each row of the sparse correction is independently scaled so that on adding to $\bar{B}_i$, $\bar{B}_{i+1}$ will then satisfy the quasi-Newton equation (4.7.22).

The application of the above algorithm to the solution of the unsteady flow equations is identical to the previous nonlinear method, except that the updating formula is eq. (4.8.1) and the calculation of the vector $\bar{p}_i$ requires a sparse matrix inversion technique. This latter requirement gives rise to two variations of the sparse nonlinear method. The first used the "Double Sweep" technique described in section 4.5 to invert the Jacobian, in which elements that occurred off the tridiagonal structure were transferred to the right hand side of the system, and an iteration procedure would then be used. If no such elements occurred then no iterating would be necessary and the $\bar{p}_i$ vector would be obtained after the first back substitution. The second variation used the sparse matrix inversion procedure described in section 4.6 in which the $\bar{p}_i$ vector is, again, given directly after the first back substitution.

In both the above inversion techniques a provision was made to allow for any of the diagonal (pivotal) elements becoming small which would cause an overflow on division and prevent the elimination from continuing. If such an occasion arose then the $\bar{B}$ matrix would be re-estimated analytically as per section 4.7, part (ii). This is the same as the initialisation procedure except that in this method the calculated values are inserted into the $\bar{B}$ matrix and not their reciprocal. The process was then continued from step (4).

The total amount of store given to the $\bar{B}$ matrix was an array having JUNCT + 1 dimensions, where JUNCT = the greatest number of nodes correcting any one particular head node and the length of the array = TNN, where TNN is the total number of nodes. Although this provides for more store than required it was felt that a more exact determination was not required unless storage became a problem. Boundaries were estimated as in the previous section with unity in the diagonal elements and zeros in the off diagonal elements, and again f_j was set to zero. The update calculation eq. (4.8.1) is however, avoided at boundaries due to p_j being zero.

4.9 Program Layout

The computer programs were written so that one basic program would be used and then different methods of solving the finite difference equations could be inserted as procedures within the basic program. Thus there was no difference between a non-linear program and an explicit program except for the computational procedures. The flow diagram for the structure of the basic program is as Fig. 4.4. The computational language used was ALGOL and the programs were run on an ICL 1905E computer.

The cross-sectional information procedures, shown at the top of Fig. 4.4, supply such relevant information of a particular node to the computational procedures as, the total and convective areas and breadths, the surface area of a channel, the hydraulic radius and Chezy C. The two boundary procedures H BOUND and Q BOUND provide depths of flow and quantities of flow depending on the type of boundary and the procedure SIDEQ provides the lateral inflow information.

In explicit methods the two procedures PARAM Q and PARAM H provide new flows and depths of flow at the end of the time interval. These procedures are simply called forward from the time varying computational and output block for each node, depending on its type. In the Gauss-Seidel implicit method new values obtained from the two above procedures will be compared in the computational and output block with values obtained from the previous iteration. When they compare within a tolerance they then become the depths and flows for the end of the time interval. Although the workings of the two procedures PARAM Q and PARAM H vary with each method of solution they are called by the same name. In respect of the Double Sweep and Sparse Matrix methods they produce the co-efficients of the sparse matrix which is then solved by an additional procedure in each case for the unknowns. This additional procedure is called from the computational block repeatedly until the unknowns have converged. No such comparison is made in the final block for the Non-Linear

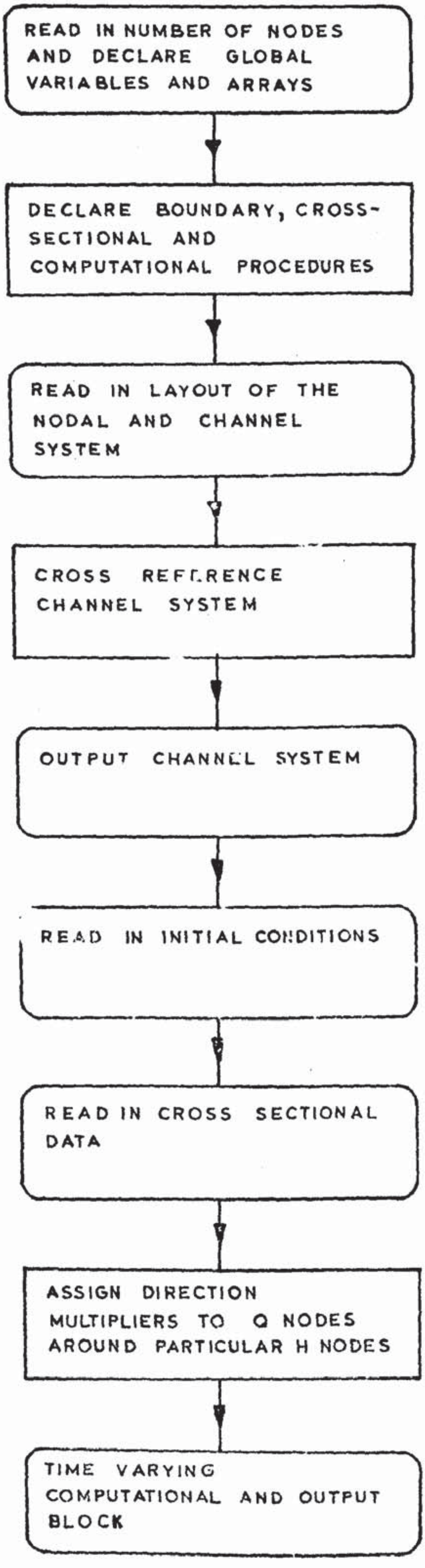

READ IN NUMBER OF NODES AND DECLARE GLOBAL VARIABLES AND ARRAYS
DECLARE BOUNDARY, CROSS-SECTIONAL AND COMPUTATIONAL PROCEDURES
READ IN LAYOUT OF THE NODAL AND CHANNEL SYSTEM
CROSS REFERENCE CHANNEL SYSTEM
OUTPUT CHANNEL SYSTEM
READ IN INITIAL CONDITIONS
READ IN CROSS SECTIONAL DATA
ASSIGN DIRECTION MULTIPLIERS TO Q NODES AROUND PARTICULAR H NODES
TIME VARYING COMPUTATIONAL AND OUTPUT BLOCK

FIGURE 4.4

methods as they contain convergence requirements within themselves. The two procedures provide the values of the vector of residuals and also initial, and when required, subsequent estimates of the Jacobian or its inverse. The procedures are again called from the procedure which solves the resulting non-linear system, which is itself initially called forward from the final block.

The channel system is read into the computer by first reading in the node reference number, then its type, i.e., the number 1 for a head hode and the number 2 for a quantity node, followed by the number of connecting channels to the node. If the number of connecting channels is one and so signifying a boundary node then the type of boundary is read in, again a 1 is used for a head boundary and a 2 if a quantity boundary. For each branch the reference number of the node at the end of the branch is read in and then the branch reference number, followed by its length. The channel system may be read into the computer in any order and the node and channel reference numbers may be of any positive integer value, not necessarily consecutive providing each one of a kind is unique.

In order that each node knows where the nodes connecting to it are in the computer the system must be cross referenced, and so for each node the position which each of the connecting nodes was read in is determined. The system is then printed out. A sample output is shown in Fig. 4.5.

Initial conditions are then read in, in the same order as the nodes. If the node is a head node then its height above a common datum and its initial depth is read in. For a quantity node the quantity of flow per second is read in with the sign convention that flow away from the connecting head node with the smallest reference number is positive and towards it is negative. In the case of the quantity node being a boundary then again the flow is positive if moving away from the node (either the connecting head node or the boundary) with the smallest reference number and vice versa. This sign convention is also adapted with the output.

As no general method of handling the survey data has been developed then it was necessary to have a specific procedure for certain systems. Data for two

NODE		NODE(S)	CONNECTING CHANNEL(S)	LENGTH
1	HB	2	1	3960
2	Q	1	1	3960
		3	2	3960
3	H	2	2	3960
		4	3	3960
4	Q	3	3	3960
		5	4	4280
5	H	4	4	4280
		6	5	3960
6	Q	5	5	3960
		7	6	3960
7	H	6	6	3960
		8	7	4700
8	Q	7	7	4700
		9	8	4700
9	H	8	8	4700
		10	9	4700
10	Q	9	9	4700
		11	10	4700
11	H	10	10	4700
		12	11	3440
12	Q	11	11	3440
		13	12	3440
13	H	12	12	3440
		14	13	3440
14	Q	13	13	3440
		15	14	3440
15	H	14	14	3440
		16	15	4970
16	Q	15	15	4970
		17	16	4970
17	H	16	16	4970
		18	17	4970
18	Q	17	17	4970
		19	18	4970
19	HB	18	18	4970

FIGURE 4.5

of the channel systems was read in as a bottom width and side slope for each node, thus schematizing the cross-sections as trapeziums.

The data was written in the same order as the nodes were read into the computer, although a simple sorting procedure could have been used. Other models were rectangular in shape and having almost constant width, these did not require data inserting separately but were handled by the cross-sectional information procedures.

The last but one block assigns direction multipliers to quantity nodes around a particular head node so that flow from that head node is positive and towards it is negative. This is to ensure that the equations of continuity are applied accurately.

Before the final block is entered then the time step DT and also the total time of the computation is set. On entering the final block the values of the variables Q_j and H_j are output for time t as per Fig. 4.6. The new values of these variables for time $t + \Delta t$ are calculated by calling the correct procedures described above and the process is repeated by outputting these, and then calculating for $t + 2\ \Delta t$ etc., until the total time of computation is reached.

Appendices A and B are copies of the computer programs for the Implicit Gauss-Seidel and Nonlinear Sparse-Sweep methods respectively.

```
TIME[T]= 7200          8 . 0  PM          ITERATIONS   6

       NODE         QUANTITY          DEPTH          BED A.O.D.        TOTAL A.O.D

         1                            23.600           -8.700            14.900
         2         11657.688                           -8.400
         3                            22.364           -8.200            14.164
         4          9940.001                           -7.950
         5                            20.992           -7.700            13.292
         6          8478.186                           -7.400
         7                            19.494           -7.200            12.294
         8          6965.007                           -6.900
         9                            17.324           -6.600            10.724
        10          4760.085                           -6.300
        11                            15.095           -6.000             9.095
        12          3127.334                           -5.800
        13                            13.758           -5.600             8.158
        14          2030.439                           -5.350
        15                            12.949           -5.100             7.849
        16          1690.315                           -4.800
        17                            11.548           -4.500             7.048
        18          -287.227                           -4.200
        19                             9.900           -3.900             6.000
```

FIGURE 4.6

CHAPTER 5

THEORETICAL CONVERGENCE AND STABILITY

5.0 Introduction

Once the two basic partial differential equations for unsteady flow have been transformed into finite difference form, one must consider (a) convergence, i.e., that the solution of the finite difference equations approach the solution of the partial differential equations as the difference steps approach zero; and (b) stability, i.e., that errors introduced into the solution of the finite difference equations due mainly to truncation but also, to a lesser degree, rounding-off remain bounded as the computation progresses.

Where the basic equations have been converted into implicit and nonlinear systems there has resulted, in each case, a set of nonlinear simultaneous equations which require certain mathematical techniques to solve. The convergence of these mathematical techniques depends upon the degree of nonlinearity of the set of equations and also upon the values of the co-efficients in the equations.

5.1 Convergence of the Finite Difference Solutions

Rigorous methods to analyse the convergence of the finite difference approximations to nonlinear partial differential equations are not yet known except for a few particular cases. It is therefore necessary to make certain simplifications and assumptions regarding the nonlinearity, so that convergence may be estimated. Such procedures usually involve linearising the equations and then to applying the relevant techniques to these linearised forms. Meaningful results can only then be expected for the linear systems and serve only as an indication for the nonlinear case.

Strang, in his paper [18] considers the convergence of first order quasi-linear hyperbolic systems. He proves that convergence depends on the stability of the linearised difference equation providing that the equations (a) possess a certain degree of smoothness, which he considers the equations of gradually varied unsteady flow have, and (b) are consistent with the original partial differential equations.

Smith [17] defines consistency to be when the finite difference equations converge to the original partial differential equations as the grid steps tend to zero. Its analysis is similar to the procedure adopted by Liggett and Woolhiser in Section 2.1 in which the order of approximation was found. The diffusing scheme suggested by them may be considered to be inconsistent owing to the presence of the additional term as $\Delta t \& \Delta x$ approach zero.

5.2 Consistency

In the following treatments of consistency the sideflow terms and terms involving channel geometry (i.e., areas and breadths etc.), are assumed, for the sake of simplicity, to vary very little over the intervals considered and can therefore take on average values.

If from a finite difference equation the original partial differential equation is subtracted, then the result is termed the truncation error, T. The first truncation error T_1 for the explicit scheme may then be obtained, for the node j, in Fig. 4.2, by subtracting eq. (4.2.1) from eq. (4.2.4) to give

$$T_1 = \left\{ \frac{(V_j^{i+1} - V_j^i)}{g\,\Delta t} + \frac{V_j^i}{g}\left[\frac{\frac{1}{2}(V_{j+2}^i + V_{j+5}^i) - V_{j-2}^i}{\frac{1}{2}(\Delta x_5 + \Delta x_2) \quad + \Delta x_1}\right] + \frac{(H_{j+1}^i - H_{j-1}^i)}{\Delta x_{K-1} + \Delta x_K}\right.$$

$$\left. + \frac{V_j^{i+1}|V_j{}^i|}{C^2R}\right\} - \left\{\frac{1}{g}\frac{\partial V}{\partial t} + \frac{V\partial V}{g\partial x} + \frac{\partial H}{\partial x} + \frac{V|V|}{C^2R}\right\} \qquad (5.2.1)$$

Equation (4.2.5) is now written for the node H_{j-1} and eq. (2.1.2) is then subtracted from it, to give the second truncation error for this system, i.e.

$$T_2 = \left\{ \frac{(H_{j-1}^{i+1} - H_{j-1}^i)}{\Delta t} + \frac{(Q_j^i - Q_{j-2}^i)}{(SA_j + SA_{j-2})}\right\} - \left\{\frac{\partial H}{\partial t} + \frac{1}{B_T}\frac{\partial Q}{\partial x}\right\} \qquad (5.2.2)$$

The above equation assumes that for both the flows, Q_j and Q_{j-2}, the positive direction is from left to right of the Fig. 4.2. In equation (5.2.1) the velocities V^{i}_{j+2} and V^{i}_{j+5} are both considered to be flowing away from the node H_{j+1}. When each of the variables in this equation are expanded by a Taylor series then the following relationships are found:

$$V^{i+1}_{j} = V(x_j,\ t_i + \Delta t) = V + \Delta t\,\frac{\partial V}{\partial t} + \frac{\Delta t^2}{2!}\frac{\partial^2 V}{\partial t^2} + O(\Delta t^2)$$

$$V^{i}_{j+2} = V(x_j + \Delta x_2,\ t_i) = V + \Delta x_2\,\frac{\partial V}{\partial x} + \frac{\Delta x_2{}^2}{2!}\frac{\partial^2 V}{\partial x^2} + O(\Delta x_2{}^3)$$

$$V^{i}_{j+5} = V(x_j + \Delta x_5,\ t_i) = V + \Delta x_5\,\frac{\partial V}{\partial x} + \frac{\Delta x_5{}^2}{2!}\frac{\partial^2 V}{\partial x^2} + O(\Delta x_5{}^3)$$

$$V^{i}_{j-2} = V(x_j - \Delta x_1,\ t_i) = V - \Delta x_1\,\frac{\partial V}{\partial x} - \frac{\Delta x_1{}^2}{2!}\frac{\partial^2 V}{\partial x^2} - O(\Delta x_1{}^3)$$

$$H^{i}_{j+1} = H(x_j + \Delta x_K,\ t_i) = H + \Delta x_K\,\frac{\partial H}{\partial x} + \frac{\Delta x_K{}^2}{2!}\frac{\partial^2 H}{\partial x^2} + O(\Delta x_K{}^3)$$

$$H^{i}_{j-1} = H(x_j - \Delta x_{K-1},\ t_i) = H - \Delta x_{K-1}\,\frac{\partial H}{\partial x} + \frac{\Delta x_{K-1}{}^2}{2!}\frac{\partial^2 H}{\partial x^2} - O(\Delta x_{K-1}{}^3) \qquad (5.2.3)$$

where $\Delta x_1 = \Delta x_{K-2} + \Delta x_{K-1}$, $\Delta x_2 = \Delta x_K + \Delta x_{K+2}$ and $\Delta x_5 = \Delta x_K + \Delta x_{K+5}$, and the right hand sides of the above relationships are centred at the point (i, j). If these are substituted into eq. (5.2.1) then the following is obtained

$$T_1 = \frac{\Delta t}{2g}\frac{\partial^2 V}{\partial t^2} + O(\Delta t^2) + \left[\frac{\frac{1}{2}(\Delta x_2{}^2 + \Delta x_5{}^2) - \Delta x_1{}^2}{\frac{1}{2}(\Delta x_2 + \Delta x_5) + \Delta x_1}\right]\frac{1}{2!}\frac{\partial^2 V}{\partial x^2}$$

$$+ \frac{1}{2!}(\Delta x_K - \Delta x_{K-1})\frac{\partial^2 H}{\partial x^2} + O(\Delta x^2) + \frac{\Delta t\,|V^{i}_{j}|}{C^2 R}\frac{\partial V}{\partial t} \qquad (5.2.4)$$

On examination of the above equation it can be seen that, by having different section lengths, the approximation to the space derivatives become, depending on the value of the differences in brackets, first order. If the section lengths were equal then the truncation error would be

$$T_1 = O(\Delta t) + O(\Delta x^2) \qquad (5.2.5)$$

otherwise it is

$$T_1 = O(\Delta t) + O(\Delta x) \qquad (5.2.6)$$

If the variables in eq. (5.2.2) are expanded by a Taylor series then the following is obtained

$$H^{i+1}_{j-1} = H(x_{j-1}, t_i + \Delta t) = H + \Delta t \frac{\partial H}{\partial t} + \frac{\Delta t^2}{2!} \frac{\partial^2 H}{\partial t^2} + O(\Delta t^3)$$

$$Q^i_{j-2} = Q(x_{j-1} - \Delta x_{K-2}, t_i) = Q - \Delta x_{K-2} \frac{\partial Q}{\partial x} + \frac{\Delta x_{K-2}^2}{2!} \frac{\partial^2 Q}{\partial x^2} - O(\Delta x_{K+2}^3)$$

$$Q^i_j = Q(x_{j-1} + \Delta x_{K-1}, t_i) = Q + \Delta x_{K-1} \frac{\partial Q}{\partial x} + \frac{\Delta x_{K-1}^2}{2!} \frac{\partial^2 Q}{\partial x^2} + O(x_{K-1}^3) \qquad (5.2.7)$$

in which the right hand sides are centred at the point (i, j-1). On substitution of eq. (5.2.7) into eq. (5.2.2) the value of the second truncation error T_2 is found to be

$$T_2 = \frac{\Delta t}{2!} \frac{\partial^2 H}{\partial t^2} + O(\Delta t^2) + \left[(\Delta x_{K-1} + \Delta x_{K-2}) \frac{\partial Q}{\partial x} + \frac{1}{2!} (\Delta x_{K-1}^2 - \Delta x_{K-2}^2) \frac{\partial^2 Q}{\partial x^2} + O(\Delta x^3)\right] \frac{1}{B_{Tj}\, \Delta x_{K-1} + B_{Tj-2}\, \Delta x_{K-2}} - \frac{\partial Q}{\partial x} \qquad (5.2.8)$$

Now if it is assumed that there is no change in depth between the Q nodes around H_{j+1} and the node H_{j+1} itself, and that it is further assumed that $B_{Tj-2} = B_{Tj} = B_{Tj-1}$ eq. (5.2.8) then becomes

$$T_2 = O(\Delta t) + \frac{1}{2!B_T} (\Delta x_{K-1} - \Delta x_{K-2}) \frac{\partial^2 Q}{\partial x^2} + O(\Delta x^2) \qquad (5.2.9)$$

Again it can be seen that by having different section lengths then the order of approximation to the space derivatives is either $O(\Delta x)$ or $O(\Delta x^2)$ depending on the values in the brackets. The order of approximation of the time derivative is one.

The first truncation error for the second explicit scheme may be obtained by subtracting eq. (4.3.3) from eq. (4.3.4) to give

$$T_1 = \left\{ \frac{(V_j^{i+1} - V_j^i)}{g\,\Delta t} - (V_j^{i+1} + V_j^i)\frac{B_T}{4A_c g\,\Delta t}\left[(H_{j-1}^{i+1} + H_{j+1}^{i+1}) - (H_{j-1}^{i} + H_{j+1}^{i})\right]\right.$$

$$- V_j^{i+1}\,V_j^i\,\frac{B_c}{2A_c g}\left[\frac{(h_{j+1}^{i+1} + h_{j+1}^{i}) - (h_{j-1}^{i+1} + h_{j-1}^{i})}{\Delta x_K + \Delta x_{K-1}}\right] + \frac{1}{2(\Delta x_K + \Delta x_{K-1})}$$

$$\left.\left[(H_{j+1}^{i+1} + H_{j+1}^{i}) - (H_{j-1}^{i+1} + H_{j-1}^{i})\right] + \frac{V_j^{i+1}|V_j^{\,i}|}{C^2R}\right\} - \left\{\frac{1}{g}\frac{\partial V}{\partial t} - \frac{V\,B_T}{A_c\,g}\frac{\partial H}{\partial t} - \frac{V^2B_c}{A_c\,g}\frac{\partial h}{\partial x}\right.$$

$$\left. + \frac{\partial H}{\partial x} + \frac{V|V|}{C^2R}\right\} \tag{5.2.10}$$

As the continuity equation for this method is the same as that used in the first explicit scheme the truncation error T_2 is also the same as eq. (5.2.9).

The variables h and H may be expanded by a Taylor series in the following manner

$$H_{j+1}^{i+1} = H(x_j + \Delta x_K,\ t_i + \Delta t) = H + \Delta x_K\frac{\partial H}{\partial x} + \Delta t\,\frac{\partial H}{\partial t} + \frac{1}{2!}\left[\Delta x_K^2\ \frac{\partial^2 H}{\partial x^2}\right.$$

$$\left. + 2\,\Delta x_K\,\Delta t\,\frac{\partial^2 H}{\partial x\,\partial t} + \Delta t^2\,\frac{\partial^2 H}{\partial t^2}\right] + O(\Delta x_K^{\,3},\ \Delta t^3)$$

$$H_{j-1}^{i+1} = H(x_j + \Delta x_{K-1},\ t_i + \Delta t) = H - \Delta x_{K-1}\frac{\partial H}{\partial x} + \Delta t\,\frac{\partial H}{\partial t} + \frac{1}{2!}\left[\Delta x_{K-1}^{\ \ 2}\ \frac{\partial^2 H}{\partial x^2}\right.$$

$$\left. - 2\,\Delta x_{K-1}\,\Delta t\,\frac{\partial^2 H}{\partial x\partial t} + \Delta t^2\,\frac{\partial^2 H}{\partial t^2}\right] + O(\Delta x_{K-1}^3,\ \Delta t^3) \tag{5.2.11}$$

and similarly for h_{j+1}^{i+1} and h_{j-1}^{i+1}. Expansions for h_{j+1}^{i} and h_{j-1}^{i} are similar to H_{j+1}^{i} and H_{j-1}^{i}.

Using the above and expansions from eq. (5.2.3) the following may be formed

$$\frac{(H_{j+1}^{i+1} + H_{j-1}^{i+1})-(H_{j+1}^{i} + H_{j-1}^{i})}{2\ \Delta t} = \frac{\partial H}{\partial t} + \frac{1}{2}(\Delta x_K - \Delta x_{K-1})\ \frac{\partial^2 H}{\partial x\ \partial t} + \frac{\Delta t}{2}\ \frac{\partial^2 H}{\partial t^2}$$

$$+ O(\Delta x^2,\ \Delta t^2) \qquad (5.2.12)$$

$$\frac{(h_{j+1}^{i+1} + h_{j+1}^{i})-(h_{j-1}^{i+1} + h_{j-1}^{i})}{2\ (\Delta x_K + \Delta x_{K-1})} = \frac{\partial h}{\partial x} + \frac{1}{2}(\Delta x_K - \Delta x_{K-1})\ \frac{\partial^2 h}{\partial x^2} + \Delta t\ \frac{\partial^2 h}{\partial x\ \partial t}$$

$$+ O(\Delta x^2,\ \Delta t^2) \qquad (5.2.13)$$

and similarly

$$\frac{(H_{j+1}^{i+1} + H_{j+1}^{i})-(H_{j-1}^{i+1} + H_{j-1}^{i})}{2\ (\Delta x_K + \Delta x_{K-1})} = \frac{\partial H}{\partial x} + \frac{1}{2}(\Delta x_K - \Delta x_{K-1})\ \frac{\partial^2 h}{\partial x^2} + \Delta t\ \frac{\partial^2 H}{\partial x\ \partial t}$$

$$+ O(\Delta x^2,\ \Delta t^2) \qquad (5.2.14)$$

and as before

$$\frac{(V_j^{i+1} - V_j^{i})}{\Delta t} = \frac{\partial V}{\partial t} + \frac{\Delta t}{2}\ \frac{\partial^2 V}{\partial t^2} + O(\Delta t^2) \qquad (5.2.15)$$

On examination of the above it can be seen that first order terms exist, not only owing to the difference $(\Delta x_K - \Delta x_{K-1})$ but also in Δt owing to the equations being centred on the time t_i line. These terms would vanish (as will be shown later) if the equations were centred at the point x_j, $t_i + \frac{\Delta t}{2}$.

Substitution of the above equations (5.2.13) to (5.2.15) into eq. (5.2.10) gives

$$T_1 = O(\Delta t, \Delta x^2) - \frac{B_T}{2A_c\ g}(\Delta x_K - \Delta x_{K-1})\frac{\partial^2 H}{\partial x\ \partial t}\left[V + O(\Delta t)\right] - \frac{B_T V \Delta t}{2A_c\ g}\frac{\partial H}{\partial t}$$

$$- \frac{V\ B_c\ \Delta t}{2\ A_c\ g}\frac{\partial^2 h}{\partial x\ \partial t}\left[V + O(\Delta t)\right] - \frac{V\ B_c}{A_c\ g}\frac{\partial h}{\partial x}O(\Delta t) + \frac{1}{2}(\Delta x_K - \Delta x_{K-1})\frac{\partial^2 H}{\partial x^2} + \Delta t\frac{\partial^2 H}{\partial x\ \partial t}$$

(5.2.16)

As stated previously the terms involving Δt would vanish if the equations were centred around the point x_j, $t_i + \frac{\Delta t}{2}$, thus employing central differences for the time derivative. The question then arises whether or not the dynamic equation may be considered to be centred at $t_i + \frac{\Delta t}{2}$ or just at t_i, as the new values for h and H at time $t_i + \Delta t$ have only been approximated to first order. If the equation is considered to be centred at time t_i only, then it would appear that the derivatives $\frac{\partial h}{\partial x}$ and $\frac{\partial H}{\partial x}$ would be better approximated by using the values on the line t_i only and not using values at $t_i + \Delta t$ as well. When the second truncation error T_2 is considered this is also of first order in Δt and so the above problem would appear to require no further consideration.

The first truncation error for the implicit scheme may be obtained by subtracting equations (4.4.1) from (4.4.2) to give

$$T_1 = \left\{\frac{1}{A_c g\ \Delta t}(Q_j^{i+1} - Q_j^i) - \frac{(Q_j^i + Q_j^{i+1})(B_c + B_T)}{4\ A_c^2\ g\ \Delta t}\left[(H_{j-1}^{i+1} + H_{j+1}^{i+1}) - (H_{j-1}^i + H_{j+1}^i)\right]\right.$$

$$- \frac{Q_j^{i+1}\ Q_j^i\ B_c}{2\ A_c^3\ g}\left[\frac{(h_{j+1}^{i+1} + h_{j+1}^i) - (h_{j-1}^{i+1} + h_{j-1}^i)}{\Delta x_K + \Delta x_{K-1}}\right] + \frac{1}{2(\Delta x_K + \Delta x_{K-1})}\left[(H_{j+1}^{i+1} + H_{j+1}^i)\right.$$

$$\left.\left. - (H_{j-1}^{i+1} + H_{j-1}^i)\right] + \frac{Q_j^{i+1}|Q_j^i|}{C^2 A_c^2 R}\right\} - \left\{\frac{1}{A_c\ g}\frac{\partial Q}{\partial t} - \frac{Q}{A_c^2 g}(B_c + B_T)\frac{\partial H}{\partial t} - \frac{Q^2 B_c}{A_c^3 g}\frac{\partial h}{\partial x}\right.$$

$$\left. + \frac{\partial H}{\partial x} + \frac{Q|Q|}{C^2 A_c^2 R}\right\}$$

(5.2.17)

The second truncation error T_2 is determined by subtracting eq. (2.1.2) from eq. (4.4.3) written for the node H_{j-1} in Fig. 4.2;

$$T_2 = \left\{ \frac{(H^{i+1}_{j-1} - H^{i}_{j-1})}{\Delta t} + \frac{(Q^{i+1}_{j} + Q^{i}_{j}) - (Q^{i+1}_{j-2} + Q^{i}_{j-2})}{(SA^{i+1}_{j} + SA^{i}_{j}) + (SA^{i+1}_{j-2} + SA^{i}_{j-2})} \right\} - \left\{ \frac{\partial H}{\partial t} - \frac{1}{B_T} \frac{\partial Q}{\partial x} \right\} \qquad (5.2.18)$$

The variables in eq. (5.2.17) are expanded in a Taylor series about the point j, i + ½ shown in Fig. 5.1 to give

$$Q^{i+1}_{j} = Q(x_j, t' + \frac{\Delta t}{2}) = Q + \frac{\Delta t}{2} \frac{\partial Q}{\partial t} + \frac{\Delta t^2}{8} \frac{\partial^2 Q}{\partial t} + O(\Delta t^3)$$

$$Q^{i}_{j} = Q(x_j, t' - \frac{\Delta t}{2}) = Q - \frac{\Delta t}{2} \frac{\partial Q}{\partial t} + \frac{\Delta t^2}{8} \frac{\partial^2 Q}{\partial t^2} - O(\Delta t^3)$$

$$H^{i+1}_{j+1} = H(x_j + \Delta x_K, t' + \frac{\Delta t}{2}) = H + \Delta x_K \frac{\partial H}{\partial x} + \frac{\Delta}{2} t \frac{\partial H}{\partial t} + \frac{1}{2!} \left[\Delta x_K^{\,2} \frac{\partial^2 H}{\partial x^2} \right.$$

$$\left. + \Delta x_K \, \Delta t \frac{\partial^2 H}{\partial x \, \partial t} + \frac{\Delta t^2}{4} \frac{\partial^2 H}{\partial t^2} \right] + O(\Delta x_K^{\,3}, \Delta t^3)$$

$$H^{i+1}_{j-1} = H(x_j - \Delta x_{K-1}, t' + \frac{\Delta t}{2}) = H - \Delta x_{K-1} \frac{\partial H}{\partial x} + \frac{\Delta t}{2} \frac{\partial H}{\partial t} + \frac{1}{2!} \left[\Delta x_{K-1}^{\,2} \frac{\partial^2 H}{\partial x^2} \right.$$

$$\left. - \Delta x_{K-1} \, \Delta t \frac{\partial^2 H}{\partial x \, \partial t} + \frac{\Delta t^2}{4} \frac{\partial^2 H}{\partial t^2} \right] + O(\Delta x_{K-1}^{\,3}, \Delta t^3) \qquad (5.2.19)$$

where $t' = t_i + \frac{\Delta t}{2}$. Similar expansions may be obtained for the remaining variables H^{i}_{j+1}, H^{i}_{j-1}, h^{i}_{j+1}, h^{i+1}_{j+1} etc.

The following relationships may then be formed

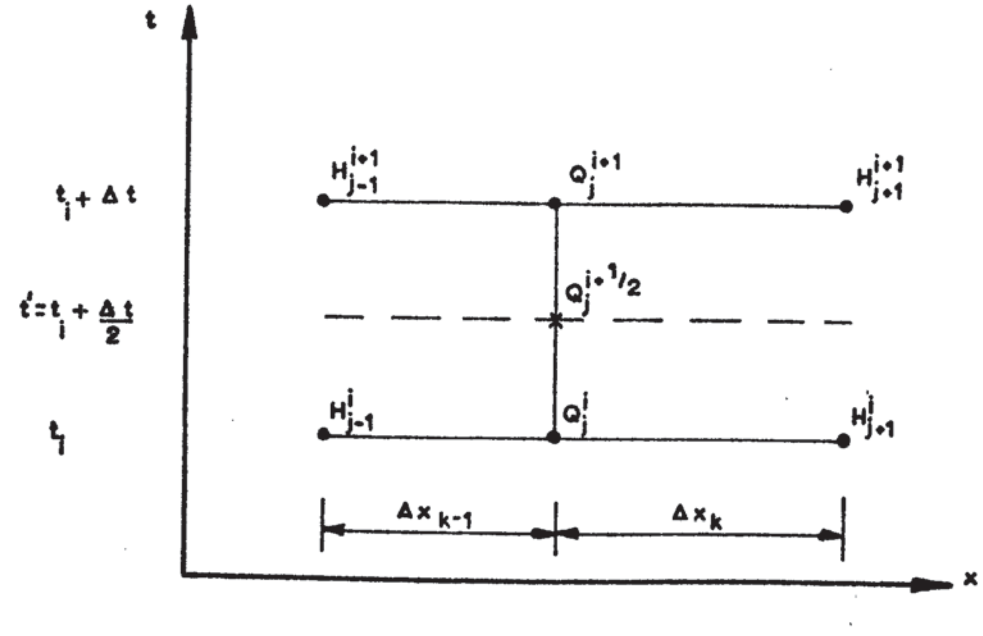


FIGURE 5.1

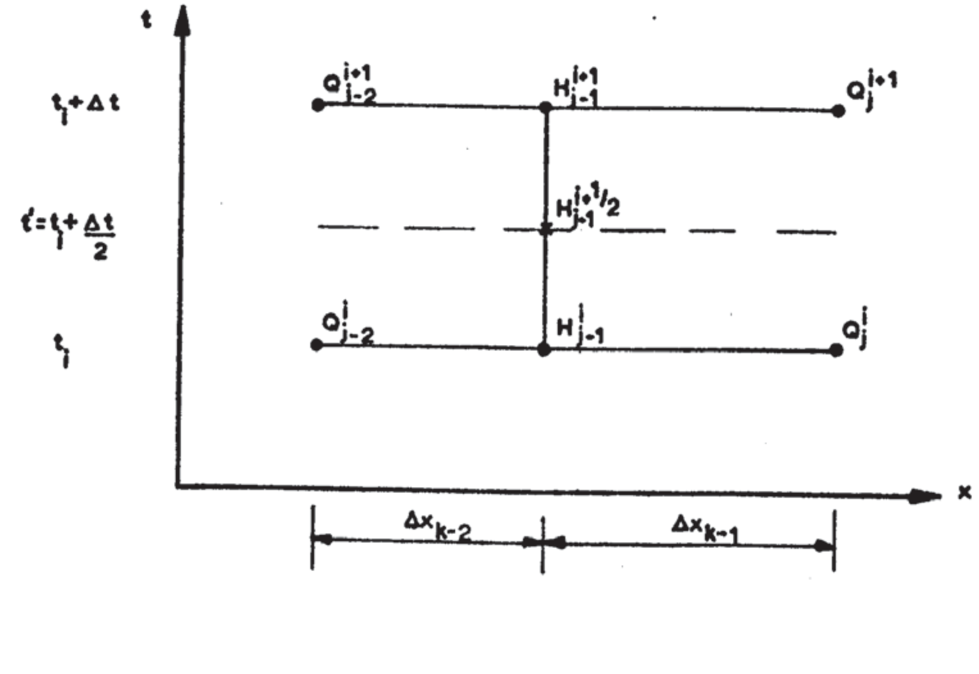


FIGURE 5.2

$$\frac{(Q_j^{i+1} - Q_j^i)}{\Delta t} = \frac{\partial Q}{\partial t} + O(\Delta t^2) \qquad (5.2.20)$$

$$\frac{(Q_j^{i+1} + Q_j^i)}{2} = Q + \frac{\Delta t^2}{8}\frac{\partial^2 Q}{\partial t^2} + O(\Delta t^4) \qquad (5.2.21)$$

$$Q_j^{i+1}\ Q_j^i = Q^2 + \frac{\Delta t^2}{4}\left(\frac{\partial^2 Q}{\partial t^2} - \frac{\partial Q}{\partial t}\right) + O(\Delta t^4) \qquad (5.2.22)$$

$$\frac{(H_{j+1}^{i+1} + H_{j-1}^{i+1}) - (H_{j+1}^{i} + H_{j-1}^{i})}{2\ \Delta t} = \frac{\partial H}{\partial t} + \frac{(\Delta x_K - \Delta x_{K-1})}{2}\frac{\partial^2 H}{\partial x\ \partial t} + O(\Delta x^2, \Delta t^2) \qquad (5.2.23)$$

$$\frac{(H_{j+1}^{i+1} + H_{j+1}^{i}) - (H_{j-1}^{i+1} + H_{j-1}^{i})}{2\ (\Delta x_K + \Delta x_{K-1})} = \frac{\partial H}{\partial x} + \frac{1}{2}(\Delta x_K - \Delta x_{K-1})\frac{\partial^2 H}{\partial x^2} + O(\Delta x^2, \Delta t^2) \qquad (5.2.24)$$

Similar expressions to equations (5.2.23) and (5.2.24) may be obtained for the terms involving h. On substitution of the above into eq. (5.2.17) the value of T_1 becomes

$$T_1 = O(\Delta x^2, \Delta t^2) - \frac{Q}{Ac^2g}(B_c + B_T)(\Delta x_K - \Delta x_{K-1})\frac{\partial^2 H}{\partial x\ \partial t} + \frac{1}{2}(\Delta x_K - \Delta x_{K-1})\frac{\partial^2 H}{\partial x^2}$$

$$- \frac{Q^2\ B_c}{2A_c^3 g}(\Delta x_K - \Delta x_{K-1})\frac{\partial^2 h}{\partial x^2} \qquad (5.2.25)$$

and so the approximation is second order apart from the terms involving the difference $(\Delta x_K - \Delta x_{K-1})$.

Expressions for the variables in eq. (5.2.18) may be obtained by expanding them from the point j-1, i+½ (shown in Fig. 5.2); these then become

$$H^{i+1}_{j-1} = H(x_{j-1}, t' + \frac{\Delta t}{2}) = H + \frac{\Delta t}{2}\frac{\partial H}{\partial t} + \frac{\Delta t^2}{8}\frac{\partial^2 H}{\partial t^2} + O(\Delta t^3)$$

$$H^{i}_{j-1} = H(x_{j-1}, t' - \frac{\Delta t}{2}) = H - \frac{\Delta t}{2}\frac{\partial H}{\partial t} + \frac{\Delta t^2}{8}\frac{\partial^2 H}{\partial t^2} + O(\Delta t^3)$$

$$Q^{i+1}_{j} = Q(x_{j-1} + \Delta x_{K-1}, t' + \frac{\Delta t}{2}) = Q + \Delta x_{K-1}\frac{\partial Q}{\partial x} + \frac{\Delta t}{2}\frac{\partial Q}{\partial t} +$$

$$\frac{1}{2!}\left[\Delta x_{K-1}^2\frac{\partial^2 Q}{\partial x^2} + \Delta x_{K-1}\,\Delta t\,\frac{\partial^2 Q}{\partial x\,\partial t} + \frac{\Delta t^2}{4}\frac{\partial^2 Q}{\partial t^2}\right] + O(\Delta x_{K-1}^3, \Delta t^3)$$

$$Q^{i+1}_{j-2} = Q(x_{j-1} - \Delta x_{K-2}, t' + \frac{\Delta t}{2}) = Q - \Delta x_{K-2}\frac{\partial Q}{\partial x} + \frac{\Delta t}{2}\frac{\partial Q}{\partial t} +$$

$$\frac{1}{2!}\left[\Delta x_{K-2}^2\frac{\partial^2 Q}{\partial x^2} - \Delta x_{K-2}\,\Delta t\,\frac{\partial^2 Q}{\partial x\,\partial t} + \frac{\Delta t^2}{4}\frac{\partial^2 Q}{\partial t^2}\right] - O(\Delta x_{K-2}^3, \Delta t^3) \qquad (5.2.26)$$

and similarly for Q^i_j and Q^i_{j-2}.

As stated at the beginning of this section, the geometrical properties are assumed to vary very little compared with H and consequently take on average values.

Hence; $B^{i+1}_{Tj} = B^{i}_{Tj} = B^{i+1}_{Tj-2} = B^{i}_{Tj-2} = B_{Tj-1}$.

$$\sum_{m=0,-2} (SA^{i+1}_{j+m} + SA^{i}_{j+m}) = 2\, B_{Tj-1}\,(\Delta x_{K-1} + \Delta x_{K-2}) \qquad (5.2.27)$$

Substitution of these together with equations (5.2.26) and similar into eq. (5.2.18) gives

$$T_2 = O(\Delta x^2, \Delta t^2) + \frac{1}{2\,B_T}(\Delta x_{K-1} - \Delta x_{K-2})\frac{\partial^2 Q}{\partial x^2} \qquad (5.2.28)$$

and again the conclusion is that T_2 is of second order apart from the term involving the differences in section lengths.

As stated in Chapter 4, the only difference between the nonlinear schemes, eq. (4.7.1) and the implicit schemes, eq. (4.4.2) is in the $\frac{\partial A}{\partial x}$ and friction terms. In the nonlinear methods the values Q^2 and $Q|Q|$ are approximated by $\frac{(Q_j^{i+1} + Q_j^i)^2}{4}$ and $\frac{(Q_j^{i+1} + Q_j^i)|Q_j^{i+1} + Q_j^i|}{4}$ respectively, instead of $Q_j^{i+1}\, Q_j^i$ and $Q_j^{i+1}|Q_j^i|$ in the linear implicit methods. As the equations may again be considered to be centred around the point j, i+½ the order of the approximation $\frac{(Q_j^{i+1} + Q_j^i)}{2}$ is determined by eq. (5.2.21), and so

$$\frac{(Q_j^{i+1} + Q_j^i)^2}{4} = Q^2 + \frac{\Delta t^2}{4}\, Q\, \frac{\partial^2 Q}{\partial t^2} + O(\Delta t^4) \qquad (5.2.29)$$

which, like the approximation to $Q_j^{i+1}\, Q_j^i$, is second order. This means that the truncation error T_1 for the nonlinear system is the same as that for the implicit system. Also, as the same equation of continuity is used then the truncation error T_2 is also the same. Thus for all of the finite difference systems considered it has been shown that they are consistent with the original partial differential equations.

5.3 Stability

The stability analysis of linearised finite difference equations is normally carried out by the von Neuman technique. The conditions developed are however, considered necessary for stability, but not sufficient. The method expresses an itital line of errors in terms of a finite Fourier series and the growth or decay of a function that reduces to this series for time t = 0 is then investigated. The analysis is applied to a localised portion of the x - t plane in which the co-efficients of the linearised system are given average values for that portion of the plane and are then assumed constant. It is then hoped that if a scheme is proved stable in this simple linearised form when applied to a localised portion of the plane then it will be also stable in its complex nonlinear form, for the whole system. Strelkoff [19] says this is largely true, and also that stability analyses are carried out, not so much to ensure that a given scheme will be stable, but to permit the researcher to discard those schemes which can be shown unstable from the start. The following analysis is thus applied to a straight portion of the channel system (i.e., not including junctions) and that the portion is divided up into equal lengths.

The complex exponential form of the Fourier series is

$$\sum A_n e^{\frac{in\pi x}{L}} \tag{5.3.1}$$

where $i = \sqrt{-1}$ and L is the interval throughout which the junction is defined. It is useful to use the notation Q(pl, qk) instead of Q_j^i, then the value x/L may be expressed as $\frac{pl}{Nl}$, (p = 0, 1, --- N), where Nl = L. If the parameter σ_n is then defined as $\frac{n\pi}{Nl}$ then $A_n e^{in\pi}$ becomes $A_n e^{i\sigma_n pl}$.

The errors along the line t = 0, between x = 0 and Nl may now be expressed by $E(pl) = E_p$, such that

$$E_p = \sum_{n=0}^{N} A_n e^{i\sigma_n pl} \tag{5.3.2}$$

and gives (N+1) equations to determine the (N+1) unknown constants A_o, A_1----A_n uniquely. As the finite difference equations are now linear, and therefore separate solutions are additive, it is sufficient to consider the propagation of the error due to a single term only. The propagation of this error with respect to time is then assumed to be

$$E_{p,q} = A\ e^{i\sigma l}\ e^{\beta t} = A\ e^{i\sigma l}\ e^{\beta qk} = A\ e^{i\sigma l \zeta q} \tag{5.3.3}$$

where $\zeta = e^{\beta k}$ and α is a complex constant. The term $e^{\beta t}$ is chosen such that it vanishes when $t = 0$. The error will not increase as t increases providing $| \zeta | \leq 1$.

Let the variables V, Q and H at a particular point be

$$V = V^* + \bar{\bar{V}}e^{i\sigma l\zeta q}$$

$$Q = Q^* + \tilde{Q}e^{i\sigma l\zeta q}$$

$$H = H^* + \bar{\bar{H}}e^{i\sigma l\zeta q} \qquad (5.3.4)$$

respectively, where V*, Q* and H* are the exact solutions of the difference equations. When these are substituted into the difference equations there arises a set of relations, involving the error terms only, that are identical to the relations involving V, Q and H, as the equations are linear.

If the sideflow term is ignored then the finite difference equations for the first explicit method, equations (4.2.4) and (4.2.5), may be linearised in the following manner

$$(V_j^{i+1} - V_j^i) + a^* V_j^i + b^* (H_{j+1}^i - H_{j-1}^i) + d^* V_j^{i+1} = 0 \qquad (5.3.5)$$

and $$c^* (V_{j+2}^i - V_j^i) + (H_{j+1}^{i+1} - H_{j+1}^i) = 0 \qquad (5.3.6)$$

where $a^* = \Delta t \frac{\delta V}{\delta x}$ $\quad b^* = \frac{g\ \Delta t}{2\ \Delta x}$ $\quad d^* = \frac{g\Delta t |V|}{C^2 R}$ and $c^* = \frac{A\ \ \Delta t}{2\ B_T \Delta x}$.

On substitution of the terms in eq. (5.3.4) into equations (5.3.5) and (5.3.6), the following is given

$$\bar{V} \zeta\ (1+d^*) + \bar{V}(a^*-1) + \bar{H}b^*(e^{i\sigma l} - e^{-i\sigma l}) = 0 \qquad (5.3.7)$$

and $$\bar{V}c^*(e^{2i\sigma l}-1) + \bar{H}e^{i\sigma l}(\zeta-1) = 0 \qquad (5.3.8)$$

which on elimination of $\bar{V}$ and $\bar{H}$ and using the fact that $e^{i\sigma l} - e^{-i\sigma l} = 2i \sin \sigma l$

gives the following quadratic in ζ

$$\zeta^2(1+d^*) + \zeta\left[(a^*-1)-(1+d^*)\right] + 4c^*b^*\sin^2\sigma l - (a^*-1) = 0 \qquad (5.3.9)$$

If the solution of a general quadratic in ζ is considered to be

$$\zeta_{1,2} = -P \pm \sqrt{P^2 - B} \qquad (5.3.10)$$

where B is real and positive and P is real, then for $|\ \zeta\ | \leqslant 1$ it can be shown that B and P have to satisfy the following inequalities

$$0 \leqslant B \leqslant 1 \qquad (5.3.11)$$

and

$$-\left(\frac{1+B}{2}\right) \leqslant P \leqslant \left(\frac{1+B}{2}\right) \qquad (5.3.12)$$

In eq. (5.3.9) P and B are

$$P = \frac{a^* - d^* - 2}{2\ (1+d^*)} \qquad (5.3.13)$$

and $$B = \frac{4c^*b^*\sin^2\sigma l + (1-a^*)}{(1+d^*)} \qquad (5.3.14)$$

If it is assumed that $a^* \leqslant 1$ when this term is positive then as d^* is always positive the result is B is positive and P is negative.

Eq. (5.3.11) gives

$$\frac{4c^*b^*\sin^2\sigma l + (1-a^*)}{(1+d^*)} \leqslant 1 \qquad (5.3.15)$$

which after rearranging and substituting for a^*, b^*, c^* and d^* gives

$$\frac{\Delta t}{\Delta x} < \sqrt{\frac{\Delta t\frac{\delta V}{\delta x} + \frac{g\Delta t\,|V|}{C^2 R}}{c}} \qquad (5.3.16)$$

If Δt is taken out of the square root then

$$\Delta t \quad < \quad \left(\frac{\Delta x}{c}\right)^2 \quad \left[\frac{\delta V}{\delta x} \quad \frac{g|V|}{C^2 R}\right] \tag{5.3.17}$$

As P is negative the only useful inequality from eq. (5.3.12) is

$$P \geq - \quad \frac{(1+B)}{2} \tag{5.3.18}$$

and so

$$\frac{a^* - d^* - 2}{(1+d^*)} \quad \geq \quad - \quad \left[1 + \frac{4c^* b^* \sin^2 \sigma l + (1-a^*)}{(1+d^*)}\right] \tag{5.3.19}$$

to give

$$4c^* b^* \sin^2 \sigma l \geq 0 \tag{5.3.20}$$

which is always satisfied as c*, b* are both positive. The condition for stability of this linearised system according to the above analysis is given by the inequality eq. (5.3.17)

The second explicit method, equations (4.3.4) and (4.3.6) may be linearised to give

$$V_j^{i+1} a^* - V_j^i b^* + (H_{j+1}^{i+1} + H_{j+1}^i) - (H_{j-1}^{i+1} + H_{j-1}^i) = 0 \tag{5.3.21}$$

and $$c^*(V_{j+2}^{\,i} - V_j^i) + (H_{j+1}^{i+1} - H_{j+1}^i) = 0 \tag{5.3.22}$$

where $$a^* = \left[1 - \frac{\Delta t\ B_T}{4\ A_c} \frac{\delta H}{\delta t} - \frac{V\ \Delta t\ B_c}{2\ A_c} \frac{\delta h}{\delta x} + g\ \frac{\Delta t|V|}{C^2 R}\right] \frac{4\ \Delta x}{g\ \Delta t}$$

$$b^* = \left[1 + \frac{\Delta t\ B_T}{4\ A_c} \frac{\delta H}{\delta t}\right] \frac{4\ \Delta x}{g\ \Delta t}$$

and $$c^* = \frac{A_c}{2\ B_T} \ \frac{\Delta t}{\Delta x}$$

If the expressions for V and H given by eq. (5.3.4) are substituted into equations (5.3.21) and (5.3.22) a quadratic in ζ again arises, in which P and B are

$$P = \frac{4c^* \sin^2 \sigma l - (a^* + b^*)}{2\,a^*} \tag{5.3.23}$$

and $$B = \frac{4c^* \sin^2 \sigma l + b^*}{a^*} \tag{5.3.24}$$

If it is assumed that a* and b* are positive, i.e., that

$$\frac{\Delta t}{4}\frac{B_T}{A_c}\frac{\delta H}{\delta t} + V\,\frac{\Delta t}{2}\frac{B_c}{A_c}\frac{\delta h}{\delta x} < 1 + g\,\frac{\Delta t|V|}{C^2R} \quad \text{and also that} \quad \frac{\Delta t}{4}\frac{B_T}{A_c}\frac{\delta H}{\delta t} > -1$$

then B will also be positive as c* is positive.

For $[\,\zeta\,] \leq 1$ then from equations (5.3.11) and (5.3.12) it is found that

$$\frac{4c^* \sin^2 \sigma l + b^*}{a^*} \leq 1 \tag{5.3.25}$$

giving $$\frac{\Delta t}{\Delta x} < \frac{1}{c}\left[2g\,\frac{\Delta t|V|}{C^2R} - \Delta t\frac{B_T}{A_c}\frac{\delta H}{\delta t} - V\,\Delta t\frac{B_c}{A_c}\frac{\delta h}{\delta x}\right]^{\frac{1}{2}} \tag{5.3.26}$$

which when Δt is taken outside of the square root gives

$$\Delta t < \left(\frac{\Delta x}{c}\right)^2 \left[\frac{2g|V|}{C^2R} - \frac{B_T}{A_c}\frac{\delta H}{\delta t} - V\frac{B_c}{A_c}\frac{\delta h}{\delta x}\right] \tag{5.3.27}$$

and $$-1 - \frac{4c^*\sin^2\sigma l + b^*}{a^*} \leq \frac{4c^*\sin^2\sigma l - (a^*+b^*)}{a^*} \leq 1 + \frac{4c^*\sin^2\sigma l + b^*}{a^*} \tag{5.3.28}$$

to give $8c^*\sin^2\sigma l \geq 0$ and $2(a^*+b^*) \geq$ which are both satisfied. The stability condition obtained for the second explicit system is then given by eq. (5.3.27).

The finite difference scheme for the implicit methods, equations (4.4.2) and (4.4.3) are linearised to give

$$Q_j^{i+1}\,a^* - Q_j^i\,b^* + (H_{j+1}^{i+1} + H_{j+1}^i) - (H_{j-1}^{i+1} + H_{j-1}^i) = 0 \tag{5.3.29}$$

and $$c^*\left[(Q_{j+2}^{i+1} + Q_{j+2}^i) - (Q_j^{i+1} + Q_j^i)\right] + (H_{j+1}^{i+1} - H_{j+1}^i) = 0 \tag{5.3.30}$$

in which $$a^* = \left[1 - (B_c + B_T)\frac{\Delta t}{2\,A_c}\frac{\delta H}{\delta t} - Q\,\frac{\Delta t}{A_c^2}\frac{\delta A}{\delta x} + \frac{g\,\Delta t|Q|}{C^2A_cR}\right]\frac{4}{A_cg}\frac{\Delta x}{\Delta t}$$

$$b^* = \left[1 + (B_c + B_T)\frac{\Delta t}{2\,A_c}\frac{\delta H}{\delta t}\right]\frac{4\,\Delta x}{A_c g\,\Delta t} \text{ and } c^* = \frac{B_T\,\Delta t}{4\,\Delta x}$$

When the expressions for Q and H, given by eq. (5.3.4), are substituted into equations (5.3.29) and (5.3.30), then a quadratic in ζ is again obtained and the values of P and B are

$$P = \frac{8c^*\sin^2\sigma l - (a^*+b^*)}{2\left[4c^*\sin^2\sigma l + a^*\right]} \qquad (5.3.31)$$

and $$B = \frac{4c^*\sin^2\sigma l + b^*}{4c^*\sin^2\sigma l + a^*} \qquad (5.3.32)$$

The following assumptions are now made on a* and b*, similar to those made for the second explicit method, to ensure that these are positive, i.e., that $(B_c + B_T)\frac{\Delta t}{2\,A_c}\frac{\delta H}{\delta t} + Q\frac{\Delta t}{A_c{}^2}\frac{\delta A}{\delta x} < 1 + g\frac{\Delta t|Q|}{C^2A_cR}$ and also that $(B_c + B_T)\frac{\Delta t}{2\,A_c}\frac{\delta H}{\delta t} > -1$ when this term is negative. When the above assumptions are fulfilled then B is also positive.

For $|\zeta| < 1$ then from equations (5.3.11) and (5.3.12) the following must hold

$$\frac{4c^*\sin^2\sigma l + b^*}{4c^*\sin^2\sigma l + a^*} < 1 \qquad (5.3.33)$$

to give $a^* \geq b^*$

i.e. $$\frac{g\,\Delta t\,|Q|}{C^2A_c\,R} \geq (B_c+B_T)\frac{\Delta t}{A_c}\frac{\delta H}{\delta t} + \frac{Q\,\Delta t}{A_c{}^2}\frac{\delta A}{\delta x} \qquad (5.3.34)$$

and also that

$$-1 - \frac{4c^*\sin^2\sigma l + b^*}{4c^*\sin^2\sigma l + a^*} < \frac{8c^*\sin^2\sigma l - (a^*+b^*)}{4c^*\sin^2\sigma l + a^*} < 1 + \frac{4c^*\sin^2\sigma l + b^*}{4c^*\sin^2\sigma l + a^*} \qquad (5.3.35)$$

to give $2(a^* + b^*) \geq 0$ and $8c^*\sin^2\sigma l \geq 0$, which are both satisfied. The above system is then stable providing eq. (5.3.34) holds. As stated previously these inequalities are however, not likely to be meaningful in practice as the method is primarily for linear systems.

If the nonlinear friction and $\delta A/\delta x$ terms in the Nonlinear finite difference system are linearised as in the implicit methods, then similar stability conditions to the implicit method arise. If they are broken up into, say, $Q^i_j K$ and $Q^{i+1}_j K$, where $K = \left[g \frac{\Delta t \; |Q|}{2C^2 A_c \; R} - \frac{Q \; \Delta t \;\; \delta A}{2 \; A_c{}^2 \;\; \delta x}\right]$, then this has the result of adding K in a* and subtracting it in b*. This again gives the condition shown by eq. (5.3.34).

In the above analyses it has been assumed that the Bernoulli term, or variations of it, are small in comparison with other terms. This is similar to Dronkers' argument in section 2.2, in which he states that if the Bernoulli term is large then separate equations must be applied for a sudden change in cross-section.

5.4 Convergence of the Numerical Procedures

Solutions to the implicit and nonlinear systems can only be obtained providing the methods of solution of the resulting set of sparse nonlinear simultaneous equations are also convergent. The convergence of the mathematical techniques depends on the functions and their degree of nonlinearity and also on the value of the co-efficients.

It is possible in the case of the modified Gauss-Seidel method to find, as will be shown later, an analytical expression which relates the co-efficients of the resulting linearised system, eq. (4.4.9), to see if convergence is guaranteed. However, similar expressions for the modified Double Sweep and Sparse Matrix methods are extremely difficult to obtain. In the case of the nonlinear methods Broyden describes, in his paper [6] a local convergence proof of the sparse nonlinear method.

The convergence of an iterative system of linear equations may be investigated in the following manner;

The (r+1)th solution to the vector $\bar{u}$ is usually expressed as

$$\bar{u}^{\,(r+1)} = \bar{A}\,\bar{u}^{(r)} + \bar{c} \qquad (5.4.1)$$

where $\bar{A}$ is the iteration matrix and is independent of $\bar{u}$. If the unknown vector $\bar{u}$ has converged to the exact sŏlution then

$$\bar{u} = \bar{A}\,\bar{u} + c \tag{5.4.2}$$

is satisfied. The error vectors $\bar{e}^{(r+1)}$ and $\bar{e}^{(r)}$ may be defined as $\bar{e}^{(r+1} = \bar{u}^{(r+1)} - \bar{u}$, and $\bar{e}^{(r)} = \bar{u}^{(r)} - \bar{u}$, so that substraction of eq.(5.4.2) from eq. (5.4.1) gives

$$\bar{e}^{(r+1)} = \bar{A}\,\bar{e}^{(r)} = \bar{A}\left[\bar{A}\,\bar{e}^{(r-1)}\right] = \bar{A}^{r+1}\,e^{(o)} \tag{5.4.3}$$

If $\bar{A}$ has m linearly independent eigenvectors $\bar{v}_s$ corresponding to the eigenvalues λs, then $\bar{e}^{(o)}$ may be expressed as a linear combination of the eigenvectors in the form

$$\bar{e}^{(o)} = \sum_{s=1}^{m} c_s\,\bar{v}_s \tag{5.4.4}$$

and therefore

$$\bar{e}^{(1)} = \bar{A}\,\bar{e}^{(o)} = \sum c_s \bar{A}\,\bar{v}_s = \sum c_s\,\lambda_s\,\bar{v}_s \tag{5.4.5}$$

as $\bar{A}\,\bar{v}_s = \lambda_s\,\bar{v}_s$ from the definition of an eigenvalue, and so

$$\bar{e}^{(r)} = \sum c_s\,\lambda_s{}^r\,\bar{v}_s \tag{5.4.6}$$

Hence if the modulus of the largest eigenvalue of λ_s, i.e,ρ the spectral radius of $\bar{A}$ is < 1 then $\bar{e}^{(o)}$ will tend to zero as r increases.

The convergence of an iterative system applied to a set of linear simultaneous equations may therefore be estimated providing the value of ρ can be calculated. In certain simple cases it is possible to find analytical expressions for the eigenvalues of a matrix but this is not usually possible for complex sparse systems that occur in practise. Consequently numerical techniques have to be used. In the case of the Gauss-Seidel method then convergence is more easily determined. For the modified Gauss-Seidel method used in this thesis then convergence is gauranteed providing the initial matrix $\bar{A}$ in eq. (4.4.10) is diagonally dominant [20]. This means that the modulus of the diagonal elements must be greater than, or equal to, the sum of the modulii of the off-diagonal elements, with the further condition that for at least one row it must be greater. To satisfy these requirements then for the set of equations (4.4.9) the following two inequalities arise:

$$| \gamma_j | + | \sigma_j | \leqslant | \lambda_j | \tag{5.4.7}$$

and $\varepsilon_j \geqslant N$ (5.4.8)

where N = maximum number of connecting nodes at a particular junction and γ_j, σ_j, λ_j and ε_j are defined in section 4.4. As the elements depend upon the solutions then the inequalities must be fulfilled for each iteration. Equation (5.4.7) gives

$$\left[1 + \frac{Q_j^i \; g \, \Delta t}{C^2 A_c \; R} + q \frac{\Delta t}{A_c} - \frac{Q_j^i}{A_c{}^2} \frac{\delta A}{\delta x} \Delta t\right] \geqslant \left[\frac{(B_c + B_T)}{4 A_c}(Q_j^i + Q_j^{i+1}) + \frac{g \, A_c \, \Delta t}{2(\Delta x_K + \Delta x_{K-1})}\right] + \left[\frac{(B_c + B_T)}{4 A_c}(Q_j^i + Q_j^{i+1}) - \frac{g \, A_c \, \Delta t}{2(\Delta x_{K+} \Delta x_{K-1})}\right] \tag{5.4.9}$$

and eq. (5.4.12) gives the approximate relationship

$$\Delta t \leqslant B_T \, \Delta x \tag{5.4.10}$$

Eq. (5.4.10) is usually fulfilled for natural water courses, however eq.(5.4.9) may not be, thus making the convergence conditional. It can be seen that convergence of this method depends, not only upon the discharge, depths and geometry of the model, but also upon the time step.

CHAPTER SIX

LABORATORY EXPERIMENTS AND APPARATUS

6.0 Introduction

The schematization of a river network into a form suitable for numerical computation introduces many errors. Firstly, points or nodes are chosen along the particular reaches so that their cross-sectional geometry is assumed to be representative of the river, for up to a few miles before and after each point. Secondly the cross-sections themselves are schematised into easily computable shapes; these may be simply rectangles, or more commonly, trapeziums. The exact levels of the river beds may be difficult to obtain if a large amount of silt is present and even when they have been determined they may vary considerably over a short distance, so that averages would have to be obtained. If the Chezy 'C' values are not known for the network then accurate initial conditions must be known so that they may be calculated.

In order that reasonably correct unsteady flow profiles are obtained, then, accurate boundary conditions must be known. If these cannot be assumed periodic then again a set of correct initial conditions must be given.

In order to compare the accuracy of the finite difference systems a laboratory rig was designed. This was done so that the errors mentioned above no longer exist. With a laboratory rig the initial and boundary conditions as well as the frictional resistance, may be determined more accurately than in a real river system.

6.1 Laboratory Rig

The laboratory rig, shown in PLATE 1, was designed to simulate an unsteady flow network, with two upstream quantity controlled boundary conditions and one downstream head controlled boundary condition. The quantities of flow into each upstream channel were recorded by 'turbine' type flow meters and were governed by the valves shown in PLATE 2. Also shown on this plate is the honeycomb baffle which reduced turbulence at the inflow points. The depth of flow at the downstream end was governed by the adjustable weir shown

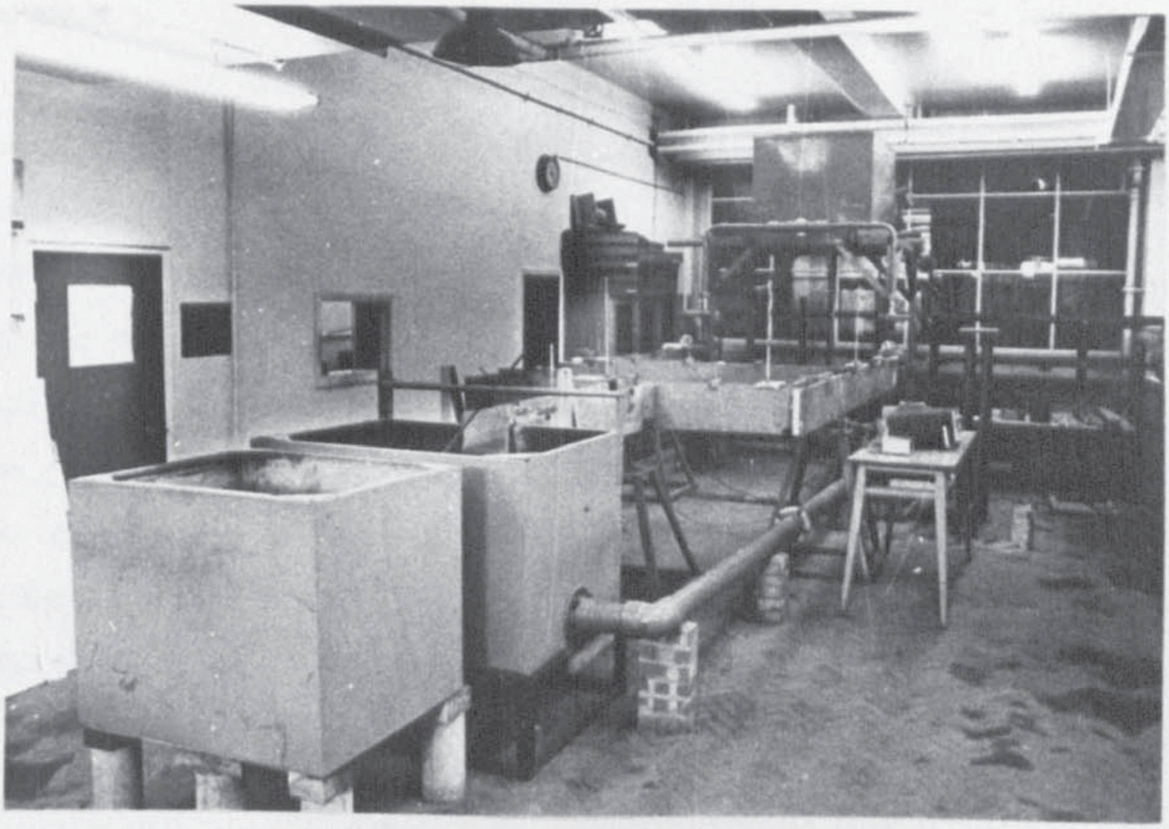


PLATE 1

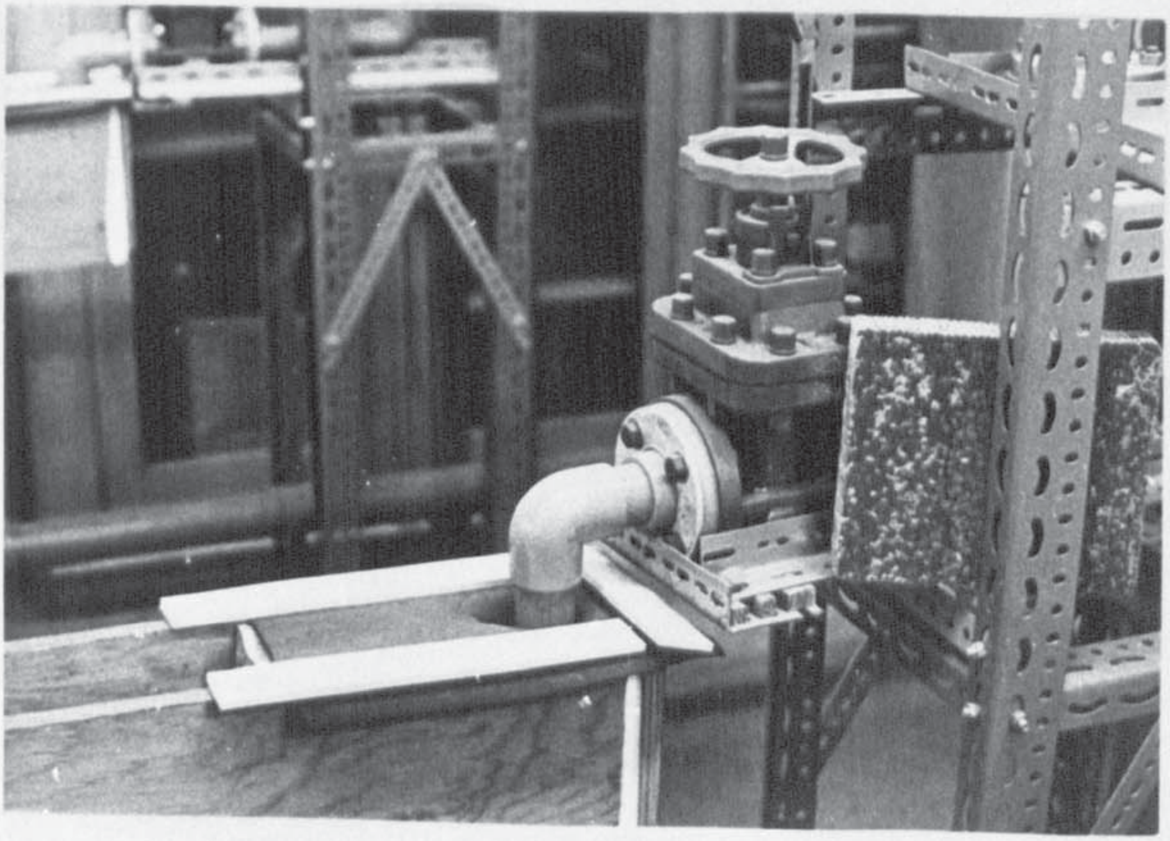


PLATE 2

in PLATE 3.

The basic channel construction was in plywood, coated with yacht varnish and attached to rolled steel channels as shown in Fig. 6.1. The entire construction was then supported on stands, positioned to minimise deflection. Between the stands and the channel structure were jacks which altered the tilt of the channel sections. The flow system, shown schematically in Fig. 6.2, was a closed one. Water from the top, constant head, tank was fed through the flow meters to the channel system. After routing through the channel network the flow was pumped from the bottom collecting tanks along the bottom 6" pipe back up to the top tank. Overflow from the top constant head tank was drawn off via a 6" pipe to the bottom collecting tanks.

Originally glass fibre tanks were positioned above the two channel ends to provide the flow controlled boundaries. These had central overflow pipes (the level of which could be adjusted) inside each tank. When one was suddenly shut off, the water would consequently rise in the tank, thus increasing the flow out of the tank, and so providing an unsteady flow system. The flow conditions from such an arrangement were, however, extremely difficult to calibrate owing to large oscillations of the water level in the tanks at fast flows; and also because of the variety of openings of the outlet valves. The method also required an accurate method of measuring the flow anyway in order to calibrate the system. The turbine meters which were finally installed proved to be a very efficient means of providing and recording unsteady flow boundaries.

6.2 Depth Gauge Development

The depth gauge shown in PLATE 4 and Fig. 6.3 is from a basic design obtained from the Civil Engineering Department, Leeds University. There, the gauge comprised of a complete copper tube which was stopped short of the none conductive base so that water may enter upwards into the gauge, the base being supported by metal strands.

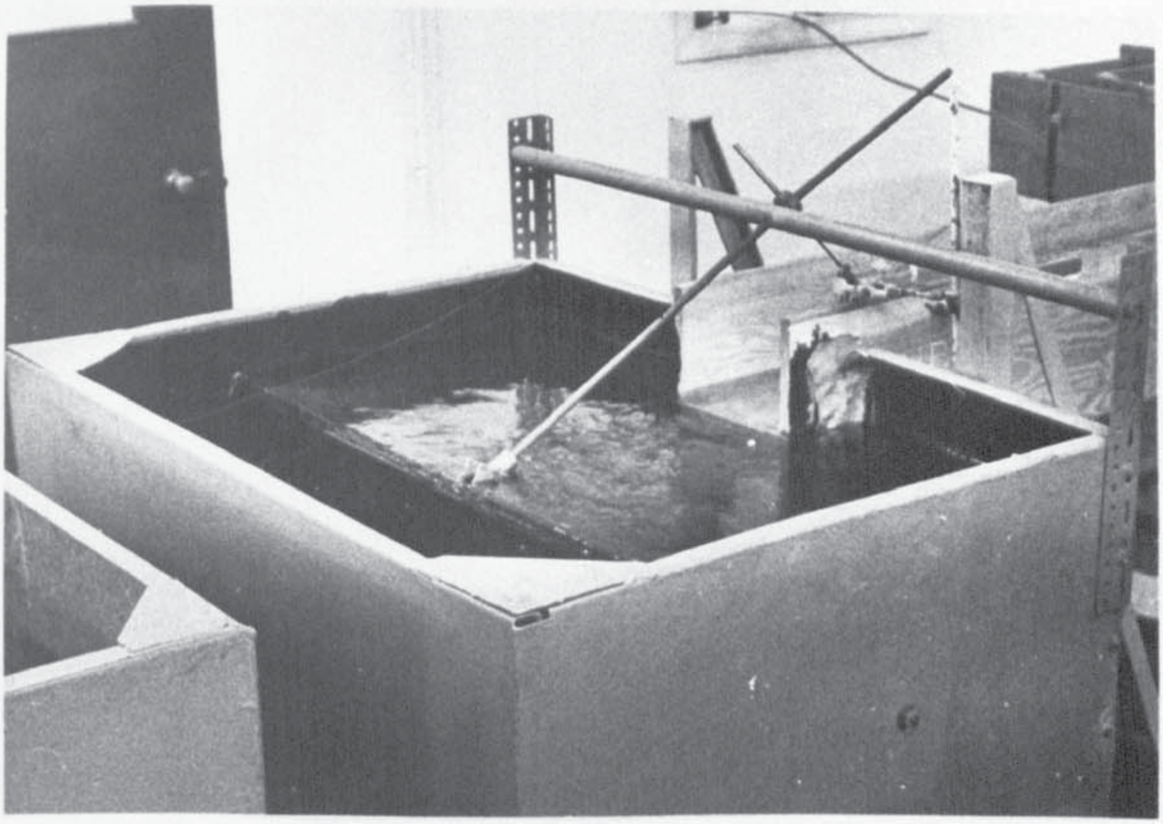


PLATE 3

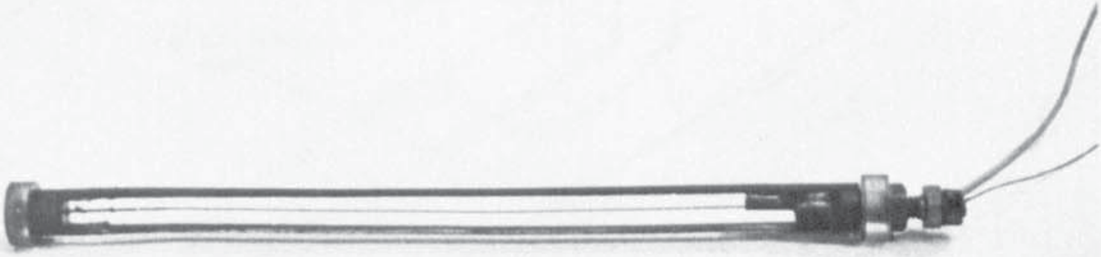


PLATE 4

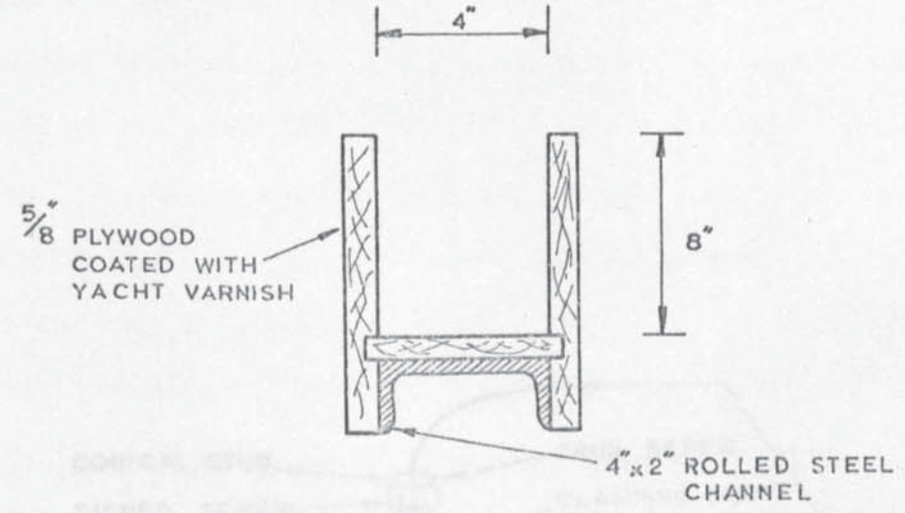

4"
8"
5/8" PLYWOOD COATED WITH YACHT VARNISH
4"x2" ROLLED STEEL CHANNEL

FIGURE 6.1

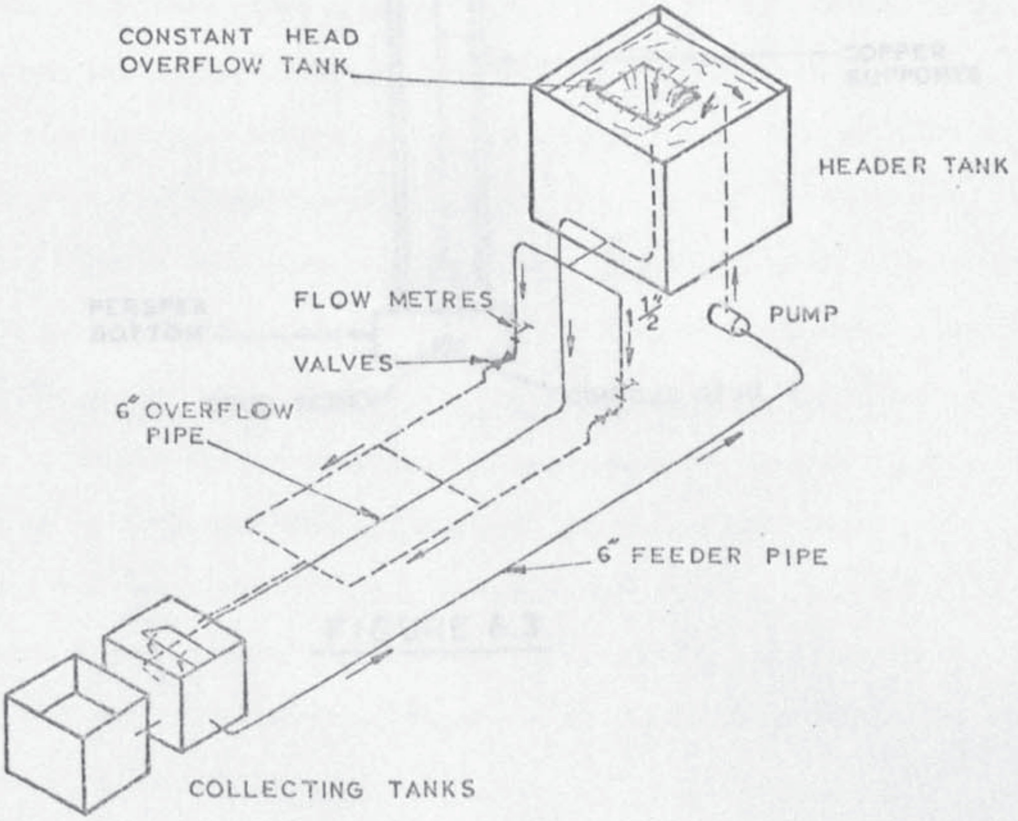


FIGURE 6.2

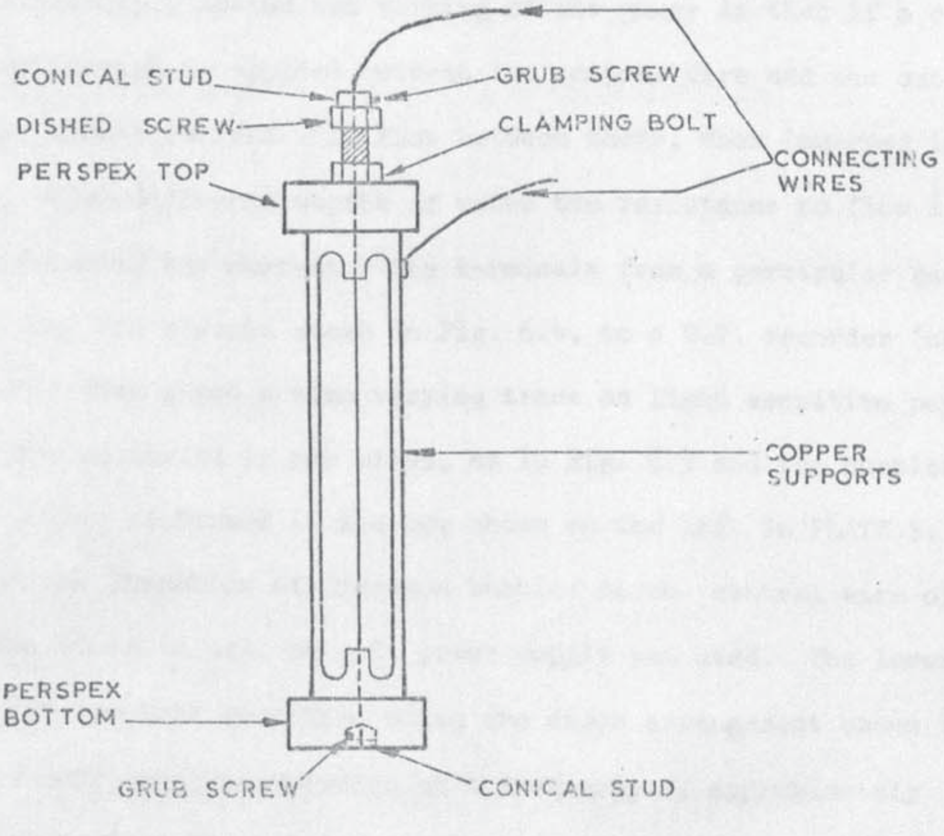

CONICAL STUD
GRUB SCREW
DISHED SCREW
CLAMPING BOLT
PERSPEX TOP
CONNECTING WIRES
COPPER SUPPORTS
PERSPEX BOTTOM
GRUB SCREW
CONICAL STUD

FIGURE 6.3

To reduce the disturbance caused by a full tube, in the relatively fast flowing water of the laboratory rig, the design was modified, so that the tube extended down to the base and four quarter slots were cut in its length. However, this was still found to cause too much backing up of the water in front of the gauge and so two opposite supports were removed. The two remaining supports were al igned with the direction of flow and this reduced the disturbance considerably, also, any time delay that may have existed with the full tube would be eliminated.

The basic principle behind the working of the gauge is that if a constant potential difference is applied between the central wire and the outer copper supports, then a current will flow between these, when immersed in ordinary water. With different depths of water the resistance to flow is altered and so altering the current. The terminals from a particular gauge are connected, via, the circuit shown in Fig. 6.4, to a U.V. recorder (shown in PLATE 5). This then gives a time varying trace on light sensitive paper. Several gauges are connected in par allel, as in Fig. 6.5 and the complete electrical circuitary is housed in the box shown on the left in PLATE 5.

To prevent the formation of hydrogen bubbles on the central wire of the depth gauge, when it is in use, an A.C. power supply was used. The lower half of the supply was then rectified using the diode arrangement shown in Fig. 6.4. The supply was then operated at a frequency of approximately 19,000 c/s to ensure that the galvanometers, whose inertia was capable of responding up to a frequency of 13,000 c/s, gave constant traces.

During the early experimental stages a more common, inexpensive oscillator was use, but this gave problems of voltage drift. This was later changed to a B & K oscillator which has an automatic gain control to ensure that a set voltage output remains constant.

Owing to the high sensitivity of the gauges then even small ripples were recorded on the U.V. trace. To try to damp out these oscillations a type of integrating device was considered. However it was thought, that this would

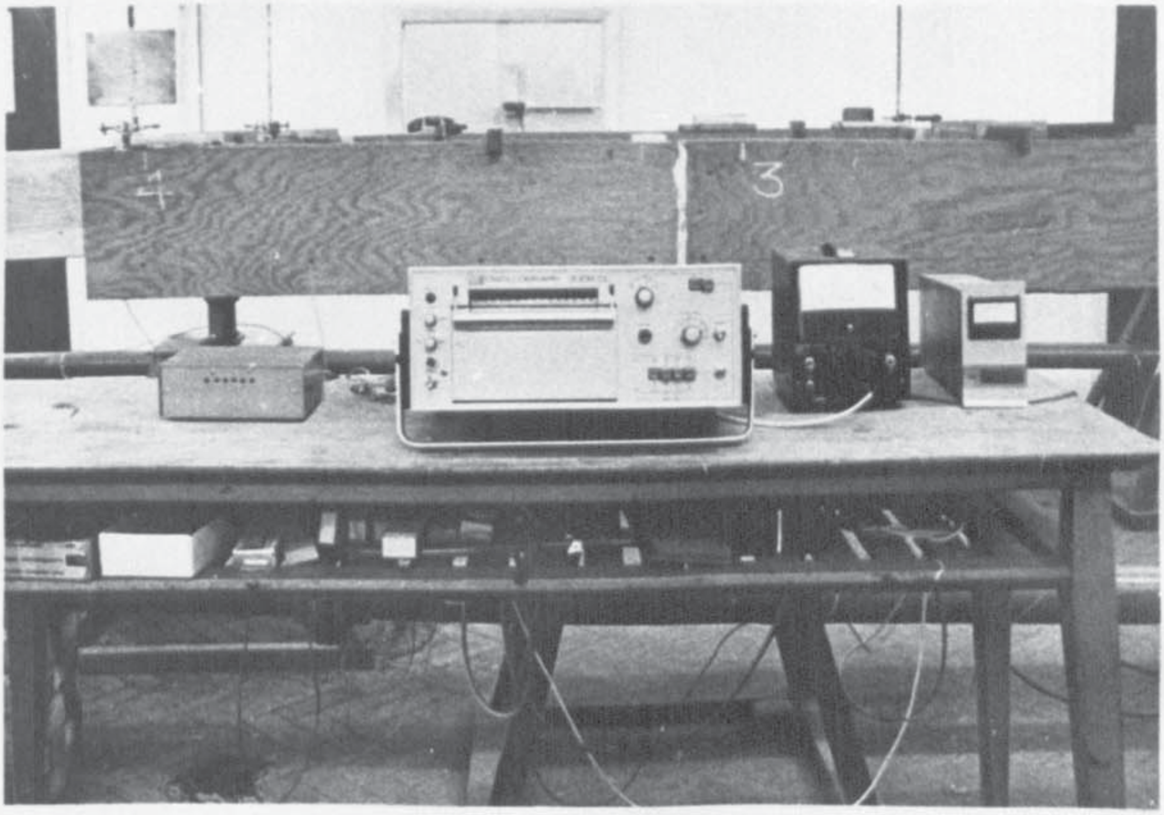


PLATE 5

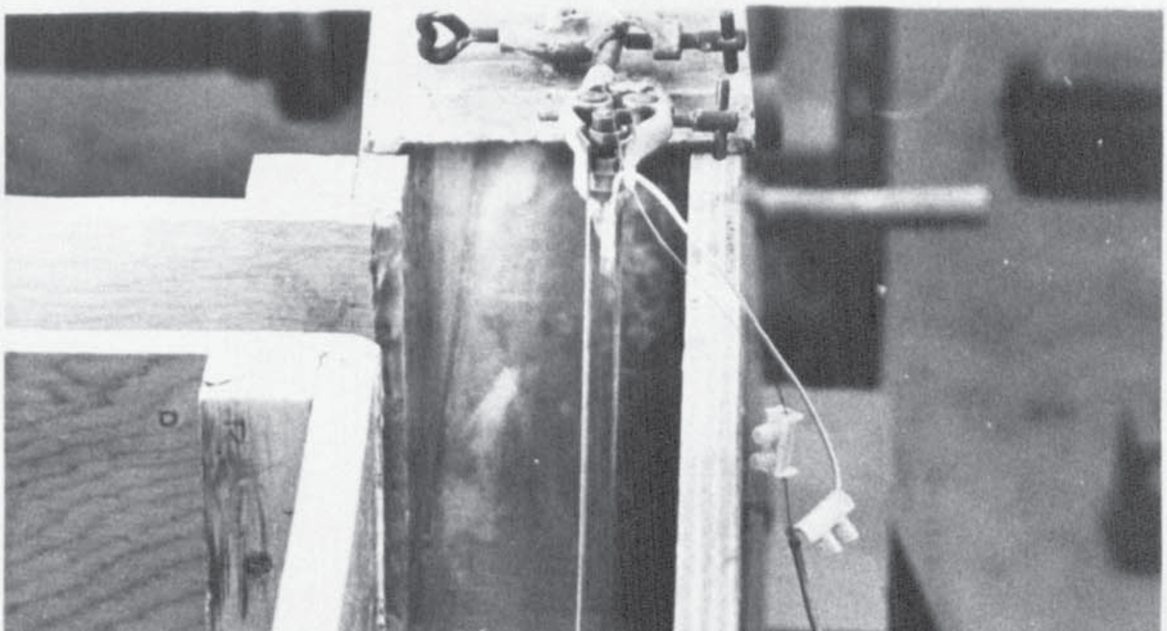


PLATE 6

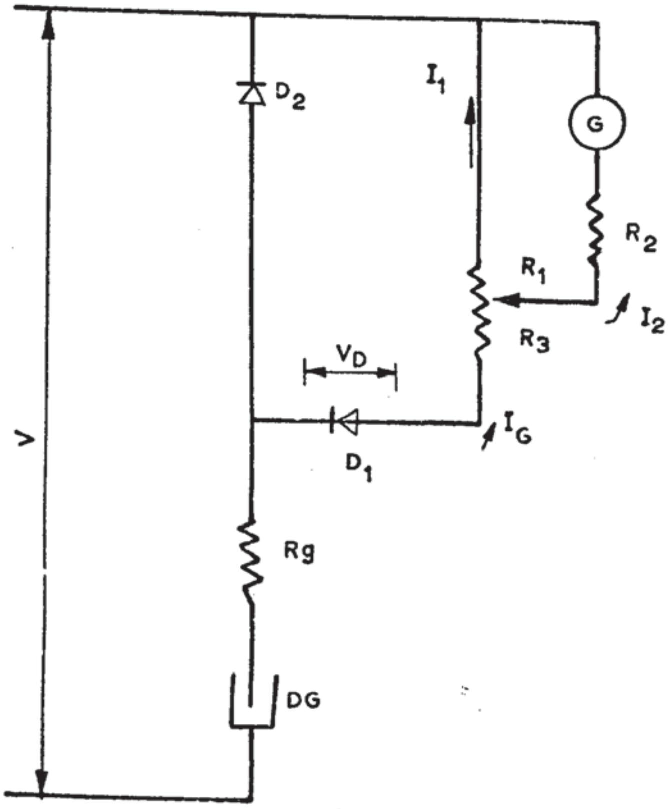


FIGURE 6.4

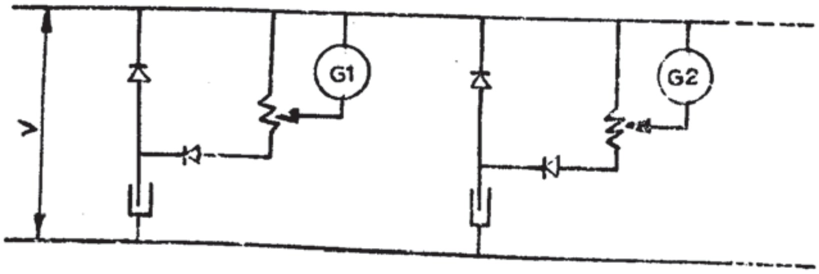


FIGURE 6.5

not only complicate the circuitary and require time to develop, but would also create a time delay between occurrences in the channel and on the recorder. Numerous attempts were made to fit capacitors into the galvanometer circuit to reduce the recording of the ripples but were to no avail. It was therefore decided to obtain the depths by determining a mean through the trace of the ripples. Fig. 6.6 shows a copy of the recording of the first unsteady flow test and PLATE 6 shows a gauge in shallow water in the channel.

6.3 Depth Gauge Design

In Fig. 6.4 DG is the depth gauge, R_G the resistance of the depth gauge D_1 and D_2 are diodes, R_1 and R_3 is a 0 to 10 ohm variable resistor, G is the galvanometer, S.E. Laboratories type B160, and R_2 is the resistance of the galvanometer. I_1, I_2 and I_G are the currents going through the parts of the circuit shown.

From Ohm's Law :

$$I_G = \frac{V - V_D}{R_G + R_c} \qquad (6.3.1)$$

where V_D = voltage drop across the diode D_1, and $R_c = R_3 + \frac{R_1 R_2}{R_1 + R_2}$, V is the supply voltage.

Providing $R_G >> R_c$ then eq. (6.3.1) becomes :

$$I_G = \frac{V - V_D}{R_G} \qquad (6.3.2)$$

If the gauge resistance R_G is assumed to be inversly proportional to the depth of flow, d, also the supply voltage and the voltage drop V_D are constant, then :

$$I_G \propto d \qquad (6.3.3)$$

Now $I_G = I_1 + I_2$ (6.3.4)

and $I_1 R_1 = I_2 R_2$ (6.3.5)

giving $I_1 = I_2 \frac{R_2}{R_1}$ (6.3.6)

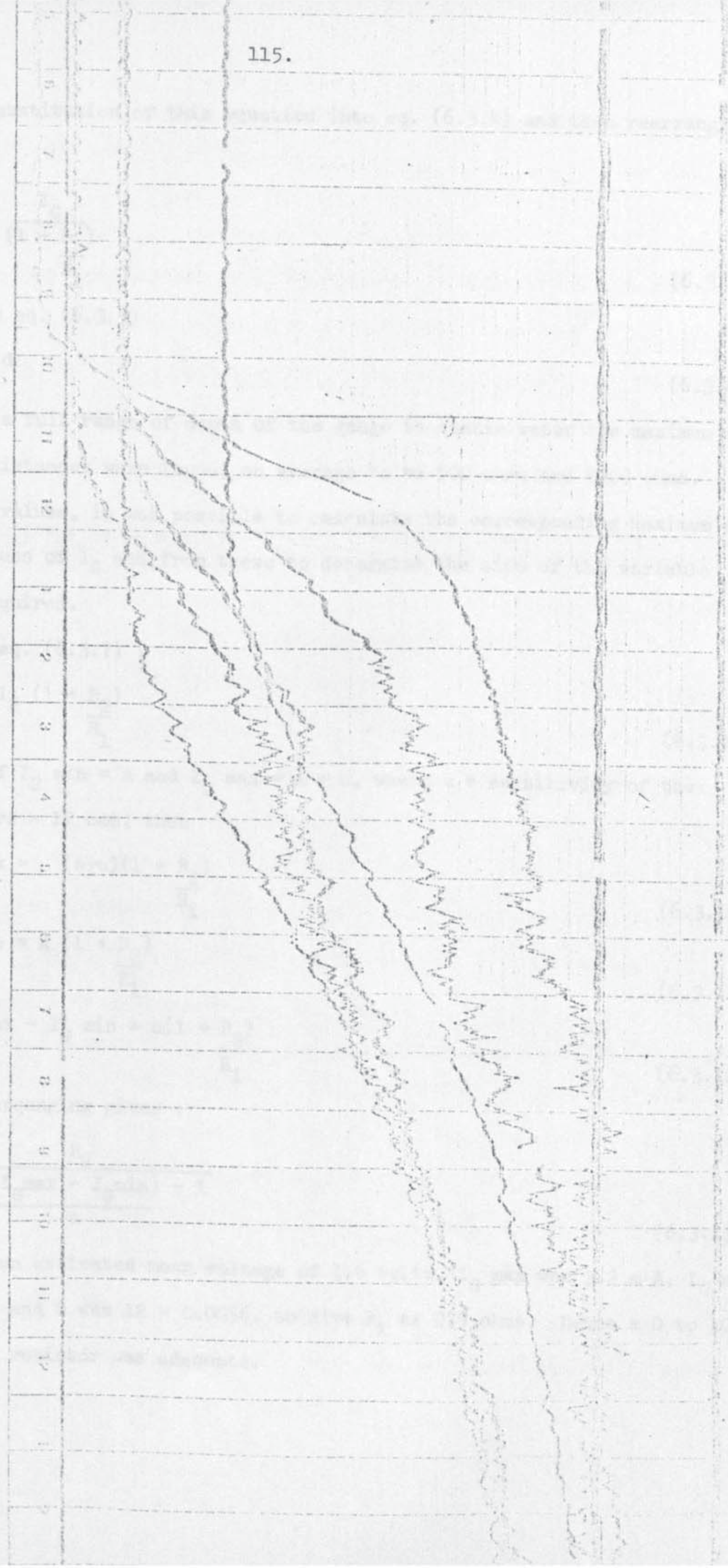


FIGURE 6.6

On substitution of this equation into eq. (6.3.4) and then rearranging gives :

$$I_2 = \frac{I_G}{(1 + \frac{R_2}{R_1})} \tag{6.3.7}$$

and so from eq. (6.3.3)

$$I_2 \propto d \tag{6.3.8}$$

Over a full range of depth of the gauge in static water the maximum and minimum resistances were found, on average to be 500 ohms and 5000 ohms. From these values, it was possible to calculate the corresponding maximum and minimum values of I_G and from these to determine the size of the variable resistor required.

From eq. (6.3.7) ;

$$I_G = I_2 (1 + \frac{R_2}{R_1}) \tag{6.3.9}$$

Now if I_2 min = A and I_2 max = A + u, where u = sensitivity of the galvanometers × 12 cms, then

$$I_G \text{ max} = (A+u)(1 + \frac{R_2}{R_1}) \tag{6.3.10}$$

and $$I_G \text{ min} = A\ (1 + \frac{R_2}{R_1}) \tag{6.3.11}$$

giving $$I_G \text{ max} - I_G \text{ min} = u(1 + \frac{R_2}{R_1}) \tag{6.3.12}$$

which on rearranging gives :

$$R_1 = \frac{R_2}{\frac{(I_G\text{max} - I_G\text{min})}{u} - 1} \tag{6.3.13}$$

Using an estimated mean voltage of 1.6 volts, I_G max was 3.2 m A, I_G min was 0.32 m A and u was 12 × 0.0056, to give R_1 as 2.5 ohms. Hence a 0 to 10 ohm variable resistor was adequate.

6.4 Calibration of Depth Gauges

The gauges were initially checked for linearity in still water. This was done by comparing actual depths with those recorded on the trace. The procedure adopted was to first of all set the oscillator to give the required output that would cover the range of depth required. Marks were then put on the gauges at 3 cms and 15 cms above the base. Each gauge in turn was then placed in still water up to the 3 cms mark and by turning the body of the galvanometer the corresponding spot on the trace was also set to 3 cms. Water was then added to the jar containing the gauge until it reached the 15 cms mark and the variable resistor was altered so that the spot coincided with the 15 cms mark on the U.V. Recorder. If the variation given by the resistor was not sufficient for the total deflection then the output voltage was increased. Conversely, if only a small amount of resistance was required then the adjustment was course, and so the output voltage was lowered. After an adjustment had been made with a variable resistor then that would throw the calibration for the 3 cms mark off and so this would have to be set again by turning the galvanometer. The depth would then have to be checked again at 15 cms and the process repeated until each gauge was correct for both marks. Once this was done it was possible to compare results for intermediate depths and these gave good agreement.

It was found however, that when the gauges were immersed in flowing water the calibration was no longer true and that the flowing water seemed to offer a different resistance to the flow of current than the still water. In order to calibrate the gauges in flowing water a series of tests were devised, the results of which are shown in TABLE 1. Each gauge was first calibrated for still water by the procedure above and was then immersed in flowing water in the channel network. For a series of different flows, ranging from small to large, traces were obtained on the U.V. Recorder and the actual depths were obtained with depth probes having vernier scales.

If the values obtained from the U.V. Recorder are y and the actual depths

TABLE 1

TEST	GAUGE RECORDING (mm)	VELOCITY OF FLOW (M/SEC)	GAUGE CONSTANTS		CALCULATED DEPTH (mm)	ACTUAL DEPTH (mm)	DIFFERENCE (mm)
			m	c			
1	25.0	0.126	1.2137	-15.9017	33.7	33.7	----
	37.5	0.194			44.0	43.8	+0.2
	48.0	0.242			52.7	52.5	+0.2
	55.5	0.285			58.8	59.6	-0.8
	63.5	0.321			65.4	66.1	-0.7
	71.0	0.355			71.6	71.6	----
2	25.5	0.126	1.1538	-13.4984	33.8	33.8	----
	38.5	0.190			45.1	44.7	+0.4
	47.0	0.240			52.4	53.0	-0.6
	56.0	0.283			60.2	59.9	+0.3
	63.0	0.320			66.3	66.2	+0.1
	70.5	0.350			72.8	72.8	----
3	32.5	0.207	1.0798	-11.7733	41.0	41.0	----
	52.0	0.141			59.1	60.2	-1.1
	64.0	0.120			70.2	70.7	-0.5
	76.0	0.314			81.3	81.0	+0.3
	91.0	0.268			95.2	95.1	+0.1
	111.0	0.224			113.7	113.7	----

are x, then providing the current flowing through the galvanometer holds a linear relationship with depth, then the depths from the U.V. recorder may be considered to hold the following law :

$$y = m\,x + c \tag{6.4.1}$$

shown in Fig. 6.7.

In each of the three tests shown in TABLE 1, the initial and deepest depths were used to determine the constants m and c, i.e.,

$$m = \frac{y_1 - y_2}{x_1 - x_2} \tag{6.4.2}$$

and $$c = y_1 - m\,x_1 \tag{6.4.3}$$

Once these values were obtained then x was determined by rearranging eq. (6.4.1) to give :

$$x = \frac{y - c}{m} \tag{6.4.4}$$

These calculated depths (x) were compared with measured depths as shown in the table. It can be seen that in all cases, except one, the differences compare to within 1mm.

In eq. (6.4.4) the variable x is differentiated with respect to each of the three parameters y, c and m to give an estimate of its error. The resulting expression is :

$$\delta x = \frac{\delta y}{m} - \frac{\delta c}{m} - \left(\frac{y - c}{m^2}\right) \delta m \tag{6.4.5}$$

Similarly eqs. (6.4.2) and (6.4.3) are differentiated with respect to each of their components to give :

$$\delta m = \frac{1}{(x_1 - x_2)} \left[\delta y_1 - \delta y_2 - m\,\delta x_1 + m\,\delta x_2\right] \tag{6.4.6}$$

and $$\delta c = \delta y_1 - x_1\,\delta m - m\,\delta x_1 \tag{6.4.7}$$

Substitution of the two above equations into eq. (6.4.5) and rearranging gives

$$\delta x = \frac{\delta y}{m} + \frac{(x_1 - x)}{(x_1 - x_2)}\left[\delta x_2 - \frac{\delta y_2}{m}\right] + \frac{(x - x_2)}{(x_1 - x_2)}\left[\delta x_1 - \frac{\delta y_1}{m}\right] \tag{6.4.8}$$

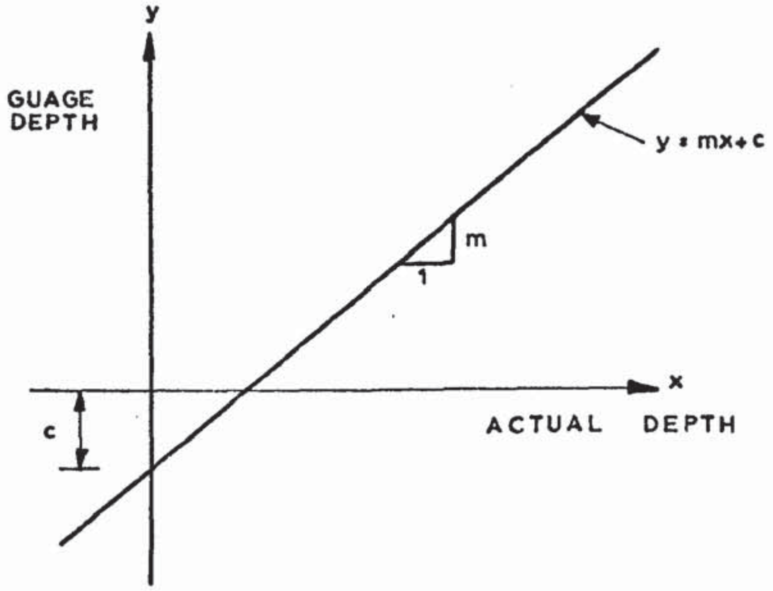


FIGURE 6.7

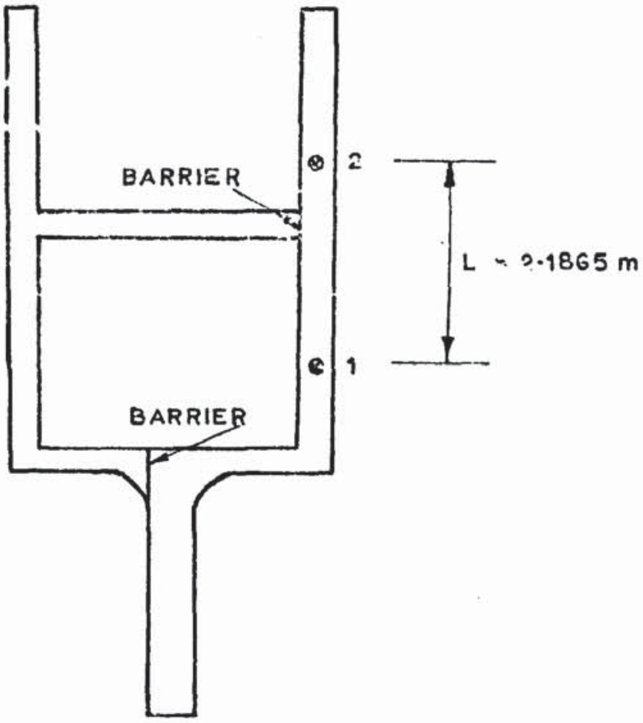


FIGURE 6.8

On examination of eq. (6.4.8) it can be seen that a linear relationship connecting the two points $x = x_1$ and $x = x_2$ exists. At $x = x_1$ we have

$$\delta x_{x=x_1} = \frac{\delta y}{m} + \delta x_1 - \frac{\delta y_1}{m} \qquad (6.4.9)$$

and at $x = x_2$ we have

$$\delta x_{x=x_2} = \frac{\delta y}{m} + \delta x_2 - \frac{\delta y_2}{m} \qquad (6.4.10)$$

On examination of TABLE 1 it can be seen that m is approximately one and so ;

$$|\delta x| < |\delta y| + |\delta x_1| + |\delta y_1| \qquad (6.4.11)$$

and

$$|\delta x| < |\delta y| + |\delta x_2| + |\delta y_2| \qquad (6.4.12)$$

whichever is the greater.

From examination of the traces it was felt that the value of the maximum error involved in each of the elements was ± 0.5 mm. This gives an overall bound on the error of $|\delta x|$ as < 1.5 mm.

6.5 Calibration of Flow Recorders

The flow recorders installed were two turbine type flow meters supplied by Nixon Instrumentation, of Gloucester, England. The meters were made in stainless steel and had a nominal bore of 1½ inches. Electrical pulses from each of the meters were sent to a visual flow indicator which gave a reading of percentage of maximum flow (32 cubic meters / hr.). From this signals were sent to the U.V. recorder to give a time varying trace of each of the boundary flows. The supply to the flow indicator was from a constant voltage supply box, set at 28v D.C.

In order that the flows may be recorded accurately the equivalent sensitivities of the galvanometers were determined. This was done by comparing readings of the U.V. trace with those on the visual indicator (the galvanometers having first been zeroed with zero flow). TABLE 2 shows these results. Meter A is on the right hand side of the apparatus (looking from above) and meter B is on the left. The galvanometer connected to meter A was

TABLE 2

METER	INDICATOR (%)	FLOW GIVEN BY INDICATOR F(M3/hr)	U.V. RECORDER d(cms)	SENSITIVITY OF GALVANOMETER $s=\frac{F}{20d}$ $\left(\frac{m\ A}{cms}\right)$	CALCULATED FLOWS F_c (M3/hr)	DIFFERENCE	
						$F-F_c$ (M3/hr)	$\frac{(F-F_c)100}{F}$ (%)
A	6.0	1.92	1.90	.0505	1.93	-0.01	-0.52
	10.0	3.20	3.15	.0507	3.21	-0.01	-0.31
	10.3	3.29	3.20	.0514	3.26	+ .03	+0.91
	10.8	3.45	3.25	.0530	3.31	+ .14	+4.06
	15.0	4.80	4.85	.0494	4.94	- .14	-2.92
	18.4	5.88	5.75	.0511	5.85	+ .03	+0.51
	20.0	6.40	6.40	.0500	6.52	- .12	-1.88
	24.5	7.84	7.60	.0515	7.74	+ .10	+1.28
	25.0	8.0	7.90	.0506	8.04	- .04	-0.50
	27.0	8.64	8.4	.0514	8.55	+ .09	+1.04
	30.0	9.6	9 .45	.0507	9.62	- .02	-0.21
	30.2	9 .66	9 .40	.0513	9.57	+ .09	+0.93
	32.8	10.49	10.40	.0504	10.59	- .10	-09 5
			AVERAGE	.0509			
B	6.0	1.92	2.05	.0468	1.98	- .06	-3.13
	6.5	2.08	2.20	.0472	2.13	- .05	-2.40
	10.0	3.20	3.30	.0484	3.19	+ .01	+0.31
	15.0	4.80	5.05	.0475	4.88	- .08	-1.67
	20.0	6.40	6.65	.0481	6.42	- .02	-0.31
	25.0	8.00	8.15	.0490	7.87	+ .03	+0.38
	29.5	9.44	9.50	.0496	9.18	+ .26	+2.75
	30.0	9.60	9.80	.0489	9.47	+ .13	+1.35
	35.0	11.20	11.4	.0491	11.01	+ .19	+1.70
			AVERAGE	.0483			

an S.E. Laboratory type B420 and that to meter B a type B450.

If F = flow rate in M^3/hr, s the sensitivity of the galvanometers in m A / cms., k = current output constant of the flow indicator, set at 20 M^3/hr. m A by the manufacturers and d = reading from the U.V. trace in centimeters, then

$$F = s.d.k \qquad (6.5.1)$$

and therefore ;

$$s = \frac{F}{20d} \qquad (6.5.2)$$

From the set of results obtained (shown in TABLE 2) the average sensitivity for each galvanometer (s_a) was determined. Using these two values the equivalent flows from the U.V. recorder readings (F_c) were obtained using

$$F_c = 20\ s_a.d \qquad (6.5.3)$$

Differentiating eq. (6.5.2) with respect to F and d gives

$$ds = \frac{dF}{20d} - \frac{F}{20d^2}\ dd \qquad (6.5.4)$$

The percentage error in s may then be expressed as :

$$\frac{\delta s}{s}\ 100 = \frac{\delta F}{F}\ 100 - \frac{\delta d}{d}\ 100 \qquad (6.5.5)$$

The estimated accuracy in reading the flow off the visual indicator was ± 0.5% of the maximum flow. With d approximately 10 cms (say) at the maximum flow case then $\frac{\delta d}{d}\ 100 \leqslant \pm\ 0.5\%$. The percentage error in s then becomes

$$\frac{\delta s}{s}\ 100 \leqslant 1.0\% \qquad (6.5.6)$$

However, the error in estimating s_a will probably be much smaller than this owing to the averaging process. Differentiating eq. (6.5.3) and putting in difference form gives

$$\delta F_c = 20\ s_a\ \delta d + 20d\ \delta s_a \qquad (6.5.7)$$

giving $\frac{\delta F_c}{F_c}\ 100 = \frac{\delta d}{d}\ 100 + \frac{\delta s_a}{s_a}\ 100$ (6.5.8)

If $\frac{\delta s_a}{s_a} 100 = \frac{\delta s}{s} 100 \leqslant \pm 1.0\%$ and $\frac{\delta d}{s} 100 \leqslant 0.5\%$ (as before) then :

$$\frac{\delta F_c}{F_c} 100 \leqslant \pm 1.5\% \text{ (of the maximum flow)} \tag{6.5.9}$$

but, as stated previously, the error in F_c will probably be much less than this owing to the averaging process.

On examination of TABLE 2 it can be seen that the differences between F and F_c compare to within ± 0.5% of the maximum flow, i.e., ± 0.16 M^3/hr. This, as stated previously is equal to the estimated accuracy in reading the scale on the visual indicator.

For the rate of change of boundary flows required, the manufacturers state that the meters are linear to within ± 0.2% of the maximum flow. This was considered small compared with that shown by eq. (6.5.9).

6.6 Determination of Channel Frictional Resistance

The Hydraulics Research Station in their publication "Charts for the Hydraulic Design of Channels and Pipes", recommend the use of the Colebrook-White equations as being the most accurate in determining the resistance offered to flow. They say that the equation has been found to be applicable to virtually any commercial surface and fluid, over a wide range of conditions. This equation, in the case of pipe flow, is :

$$\frac{1}{\sqrt{\lambda}} = -2 \log_{10} \left(\frac{k_s}{14.8 R} + \frac{2.51}{Re\sqrt{\lambda}} \right) \tag{6.6.1}$$

In which λ is the Darcy-Weisbach friction factor = $2gDs/V^2$ where D is the diameter of circular section of slope s, k_s is a linear measure of the projection of the roughness elements above the channel wall, Re is the Reynolds number, and R the hydraulic radius. Henderson [9], refers to slight modifications that have been made to this formula to render it applicable to open channel flow, also he expresses the resistance in terms of an equivalent Chezy 'C'. This equation is :

$$\frac{C}{\sqrt{8g}} = -2 \log_{10} \left(\frac{k_s}{12 R} + \frac{2.5 C}{Re \sqrt{8g}} \right) \tag{6.6.2}$$

To determine the value of k_s a set of experiments were conducted. The channel network was arranged so that water was flowing down one side only, as in Fig. 6.8.

To do this barriers of the same wood and finish as the rest of the network, were placed in the positions shown. Two depth gauges were then placed in the long straight portion of the channel, a distance of 2.1865 meters apart. A series of results were obtained recording the depths at the points (1) and (2) for different flows. Initially, and at each change in slope, the relative difference in height of the bottom of the channels at the two points were obtained, using an engineers level and a millimeter rule.

The gradually varied steady flow equation may be written as :

$$-\frac{Q^2}{A^3 g}\frac{\partial A}{\partial x} + \frac{\partial h}{\partial x} + \frac{\partial z}{\partial x} + \frac{Q^2}{C^2 A^2 R} = 0 \qquad (6.6.3)$$

This may be rearranged to give :

$$C = \frac{Q}{A\sqrt{R\left[\frac{Q^2}{A^3 g}\frac{\partial A}{\partial x} - \frac{\partial h}{\partial x} - \frac{\partial z}{\partial x}\right]}} \qquad (6.6.4)$$

which may then be expressed in finite difference form as :

$$C = \frac{2\,Q}{(A_1 + A_2)}\sqrt{\frac{2\,L}{(R_1 + R_2)\left[\frac{8\,Q^2}{g(A_1 + A_2)^3}(A_1 - A_2) - (z_1 - z_2) - (h_1 - h_2)\right]}} \qquad (6.6.5)$$

in which the subscripts refer to the positions in Fig. 6.8.

A computer program was written in which the values of C were calculated for the different flows and depths of flow using eq. (6.6.5). With each value of C calculated the program then determined the value of k_s required to give that C using the following equation :

$$k_s = 12R\left[\text{antilog}\left(\frac{-C}{2\sqrt{8g}}\right) - \frac{2.5\,C}{Re\,\sqrt{8g}}\right] \qquad (6.6.6)$$

which is simply eq. (6.6.2) rearranged.

Finally an average k_s was determined for the complete set of results and was found to be 0.055 cms . Using this, the program then calculated, on an iterative basis, a new value of C for each flow and depth of flow using eq.

(6.6.2). The results of the tests are shown in TABLE 3. It can be seen that the friction co-efficients calculated using the average value of k_s compare well with those determined using the gradually varied flow equation.

6.7 Unsteady Flow Tests

Several unsteady flow tests were done with various boundary and initial conditions. Of these, three have been chosen to illustrate the differences between the computed and measured depths. The examples chosen are those which had the greatest amount of varying conditions, i.e., from large depths to small, and vice versa.

The first unsteady flow test was with the downstream weir backing the flow up to a relatively large depth and with constant inflows at the two top boundaries. Unsteady conditions were then propagated by suddenly dropping the weir. The second test was again with initially deep depths and large boundary flows, but this time at approximately the same time the weir was dropped, the valves controlling the boundary flows were also shut off. The third test was with the initial conditions of relatively shallow depths and small flows. The two valves were then turned on full and after a short period of time were then turned off completely, the level of the downstream weir remaining unaltered.

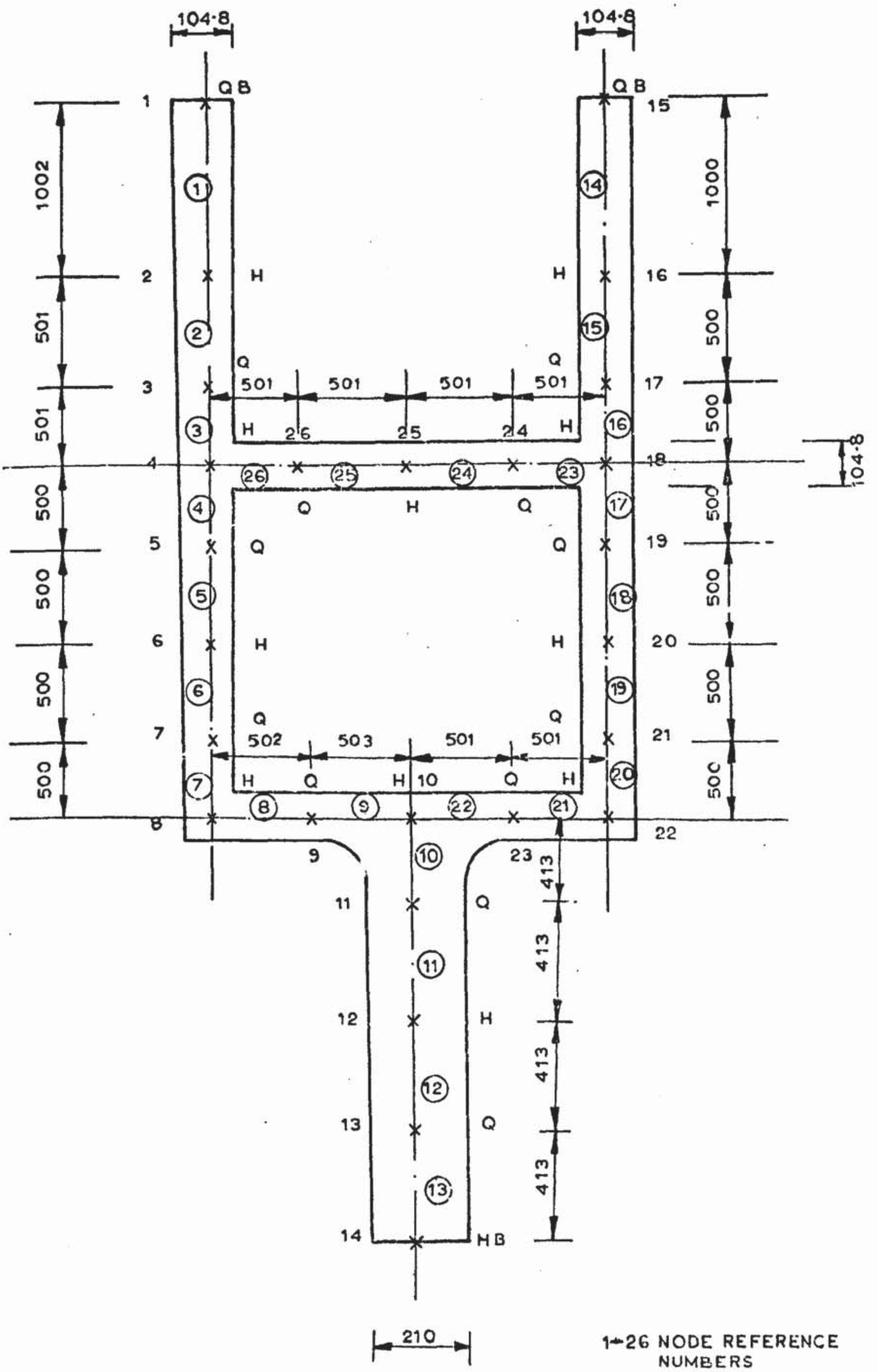
In each of the tests steady state conditions were allowed to prevail before the boundaries were altered. The second and third tests had different slopes and cross slopes on the channel system than the first. To measure these slopes the relative difference in height of the network was recorded using an engineers level and steel millimeter rule. The flow gauges were calibrated before each test by allowing steady conditions to arise for (a) deep depths, and then measuring these depths accurately with vernier depth probes, and then (b) for shallow depths which were again measured with the probes. For the two steady conditions a length of trace was run off the U.V. recorder to relate the actual depths with those recorded.

TABLE 3

DEPTHS (CMS)		FLOW Q	MEASURED VALUES		CHEZY 'C' using K_s =0.055cms
H 1	H 2	(Litres/sec)	CHEZY'C'	K_s (cms)	
9.95	9.36	4.1063	45.928	0.093	49.353
7.29	6.46	2.3172	49.453	0.043	48.048
3.66	2.68	0.5751	46.695	0.030	43.899
6.80	6.05	2.4000	45.234	0.084	47.972
6.10	5.25	1.9556	46.717	0.062	47.429
5.35	4.45	1.5111	45.923	0.063	46.719
4.45	3.45	1.0667	48.174	0.034	45.629
3.30	2.25	0.4444	41.073	0.079	42.987
4.31	3.27	0.8889	47.942	0.032	45.202
5.13	4.17	1.3333	47.128	0.048	46.398
6.61	5.75	2.2222	48.338	0.050	47.787
7.26	6.43	2.6667	49.545	0.044	48.235
			AVERAGE	0.055	

NOTE: z1 = 0, z2 = 1.05 cms for the first 3 results and = 1.15 cms for the rest.

The channel system was schematised according to Fig. 6.9 in which there were a total of 26 nodes. Depth gauges were placed at the junctions, i.e., nodes 4, 8, 10, 18 and 22 and at the downstream boundary, node 14. The computer program which was adapted for the laboratory results were programmed to output the recorded depths as well as the calculated depths at each time step, and multiples of that. This was achieved by inputting as data the constants for the gauges and flow meters and then the time varying results off the U.V. traces.



1→26 NODE REFERENCE NUMBERS

①–㉖ CHANNEL REFERENCE NUMBERS

FIGURE 6.9

ALL DIMENSIONS ARE IN MILLIMETRES

CHAPTER SEVEN

PHYSICAL NETWORKS

7.0 Introduction

To compare the accuracy, computational time and general behaviour of the finite difference systems using more realistic data, a set of unsteady flow situations were tested. The first of these was a purely hypothetical channel with no branches and having a rectangular cross-section of constant breadth. This was initially used to eliminate programming errors in the development of the finite difference systems. The model was set to have initial conditions that were inconsistent with the frictional resistance term and the fixed boundary conditions. Results from the running of this model illustrated the convergence tendencies of the finite difference systems.

The second example was an unbranched section of the River Aire in Yorkshire. Approximately 16 miles of this river was programmed to investigate its tidal propagation.

The last example is a schematization of part of the Ganges delta in which the propagation of tides were again investigated.

7.1 Hypothetical Model

The layout of the model is as shown in Fig. 7.1.

It can be seen that all the section lengths are equal to 100 meters and that there is a constant slope of 1/200. Also it can be seen that the nodes are numbered consecutively and that there are no junctions. The resulting structure of the finite difference equations for this model is then tridiagonal. The breadth of the rectangular cross-sections was a constant 10.16 M and the initial conditions were 1M for the 'H' nodes and 20 M^3/sec. for the 'Q' nodes. Boundary conditions were fixed at 20 M^3/sec. at node (1) and 1m at node (20). The Chezy 'C' value was set at 20 $M^{\frac{1}{2}}$/sec. For the stated slope, boundary flow and friction, the critical depth and normal flow depth are calculated to be 0.734 M and 1.348 M respectively.

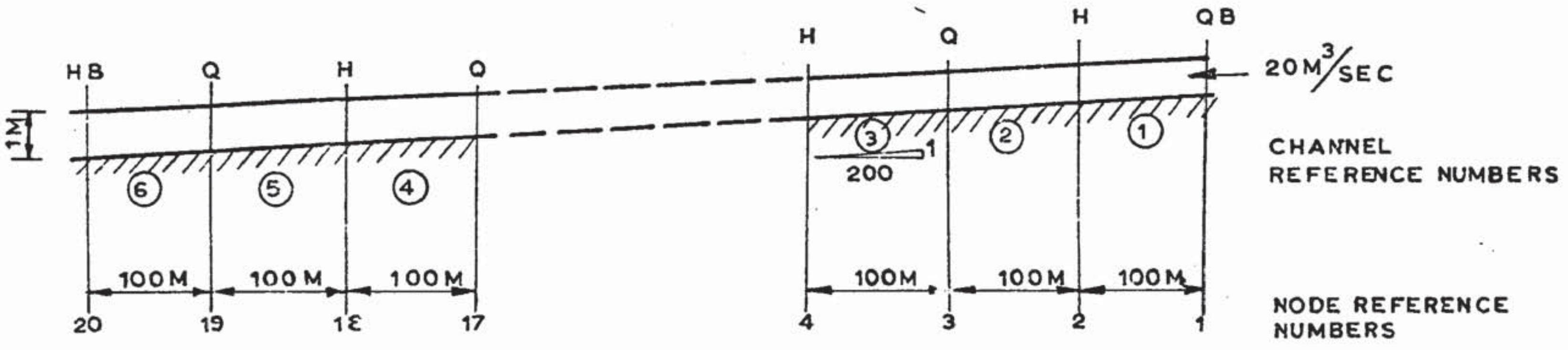

HB
Q
H
Q
H
Q
H
QB
20M³/SEC
1M
⑥
⑤
④
③
1
200
②
①
CHANNEL
REFERENCE NUMBERS
100M
100M
100M
100M
100M
100M
20
19
1ε
17
4
3
2
1
NODE REFERENCE
NUMBERS

FIGURE 7.1

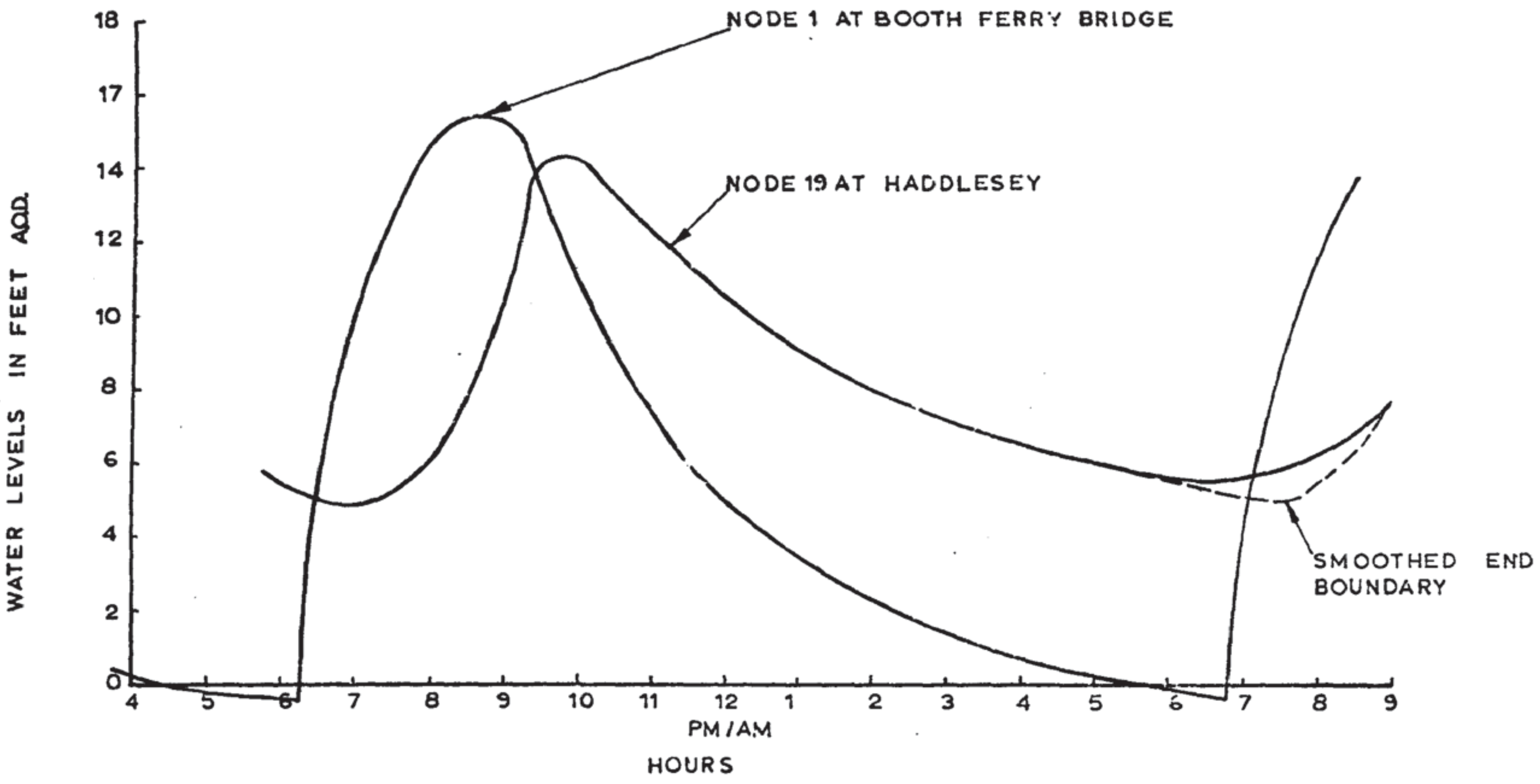
Using the above initial conditions the programs were run until steady state conditions were reached. By using a large range of time steps it was possible, from the results obtained, to determine the time steps at which the systems (a) became unstable, in the case of the explicit methods and (b) did not reach a solution, in the case of the implicit systems. Observations were also made on the behaviour of the results when large time steps were used.

7.2 River Aire

The general locality of the part of the river programmed is shown in Fig. 7.2 and in more detail in Fig. 7.3. Initially 20 nodes were used, of varying section lengths, from the junction with the River Ouse at Booth Ferry Bridge up to the weir at Haddlesey. The positions of the 20 nodes are shown in Fig. 7.2, again as the nodes are numbered consequtively and there are no junctions then the structure of the finite difference systems is tridiagonal.

The tide at Booth Ferry Bridge provided a downstream head controlled boundary and although there was no recording station at Haddlesey, it was initially thought that flows obtained from the Beal weir would suffice. However, it was later found that the charts from which the flows at Beal weir were recorded were inaccurate. The 'Q' node (20) was then abandoned and the model was curtailed at node (19), making this a head controlled boundary. The depth profile for this boundary and for Booth Ferry Bridge were obtained from a chart of river profiles for the tide investigated, i.e., 4th May 1962 and are shown in Fig. 7.4.

The author obtained the survey data for this model from his supervisor who had studied the sediment transport in the river. The cross-sections were schematized as trapeziums by taking measurements of the top width and the width 20ft. below this at frequent intervals along the river. From this data it was possible to obtain graphs of the bottom widths B_o and the overall side slopes m. Smooth curves were put through these points and



BOUNDARY CONDITIONS FOR RIVER AIRE

FIGURE 7.4

values for the node positions are shown in TABLE 4. Bed levels were obtained from echo soundings at different times in the period October 1961 to May 1962. As these levels fluctuated so much along the river at any one time it was necessary for the researcher to do a least squares fit of the results. From these graphs the author estimated the bed levels for the time of the tide in question. The resulting levels are again shown in TABLE 4.

Initial depths were available for the tide of the 4th May 1962 but no initial flows. Also lacking was the value of the Chezy factor 'C'. To obtain this data the programs were first set at the correct initial depths (obtained from the river profiles) and with estimated initial flows. Then using the boundary conditions mentioned previously, the programs were run and set to recycle at periods of 12½ hrs. At the start of each recycle the depths were set to the correct depths but the flows were set to those recently obtained.

The boundary at node (1) was found to be periodic for the period stated but at the upstream boundary, node (19), some smoothing was required (as shown). It was found that results from the 3rd cycle were the same as those from the 2nd and so making further recycling unnecessary. For each value of Chezy 'C' tried the results from the 3rd cycle were compared with the river profiles and it was found that using values of 85ft$^{\frac{1}{2}}$/sec. for the flow upstream and 75ft$^{\frac{1}{2}}$/sec. for flow downstream gave good agreement. It was thought that even better agreement could probably still be obtained by further ~~r~~modification of 'C' but was felt to be unnecessary. During the course of these tests it was found that calculated water levels along the river reach were particularly sensitive to the boundary conditions, but not necessarily to bed levels or to Chezy 'C'.

The resulting initial conditions obtained at the end of the 3rd cycle are shown in TABLE 4. With these values several programs were run using different finite difference schemes and with varying time steps. The

TABLE 4

							INITIAL CONDITIONS	
NODE	TYPE OF NODE	BOTTOM WIDTH B_o(FT)	SIDE SLOPE m	BED LEVEL A.O.D. (FT)	CON-NECTING CHANNEL	LENGTH OF CHANNEL (FT)	FLOW (FT^3/SEC)	BED LEVEL A.O.D. (FT)
1	HB	90.45	6.50	-8.7				-0.4
					1	3960		
2	Q	87.28	5.80	-8.4			-1515	
					2	3960		
3	H	80.95	5.25	-8.2				0.3
					3	3960		
4	Q	76.84	4.80	-7.95			-1432	
					4	4280		
5	H	70.35	4.50	-7.7				0.9
					5	3960		
6	Q	65.55	4.25	-7.4			-1332	
					6	3960		
7	H	63.20	4.00	-7.2				1.3
					7	4700		
8	Q	58.09	3.90	-6.9			-1226	
					8	4700		
9	H	54.92	3.80	-6.6				1.8
					9	4700		
10	Q	50.375	3.75	-6.3			-1116	
					10	4700		
11	H	48.80	3.70	-6.0				2.5
					11	3440		
12	Q	48.12	3.60	-5.8			-1025	
					12	3440		
13	H	46.40	3.50	-5.6				3.0
					13	3440		
14	Q	45.54	3.45	-5.35			-951	
					14	3440		
15	H	44.66	3.40	-5.1				3.5
					15	4970		
16	Q	44.16	3.30	-4.8			-864	
					16	4970		
17	H	43.38	3.25	-4.5				4.4
					17	4970		
18	Q	42.56	3.20	-4.2			-764	
					18	4970		
19	HB	43.91	3.10	-3.9				5.5

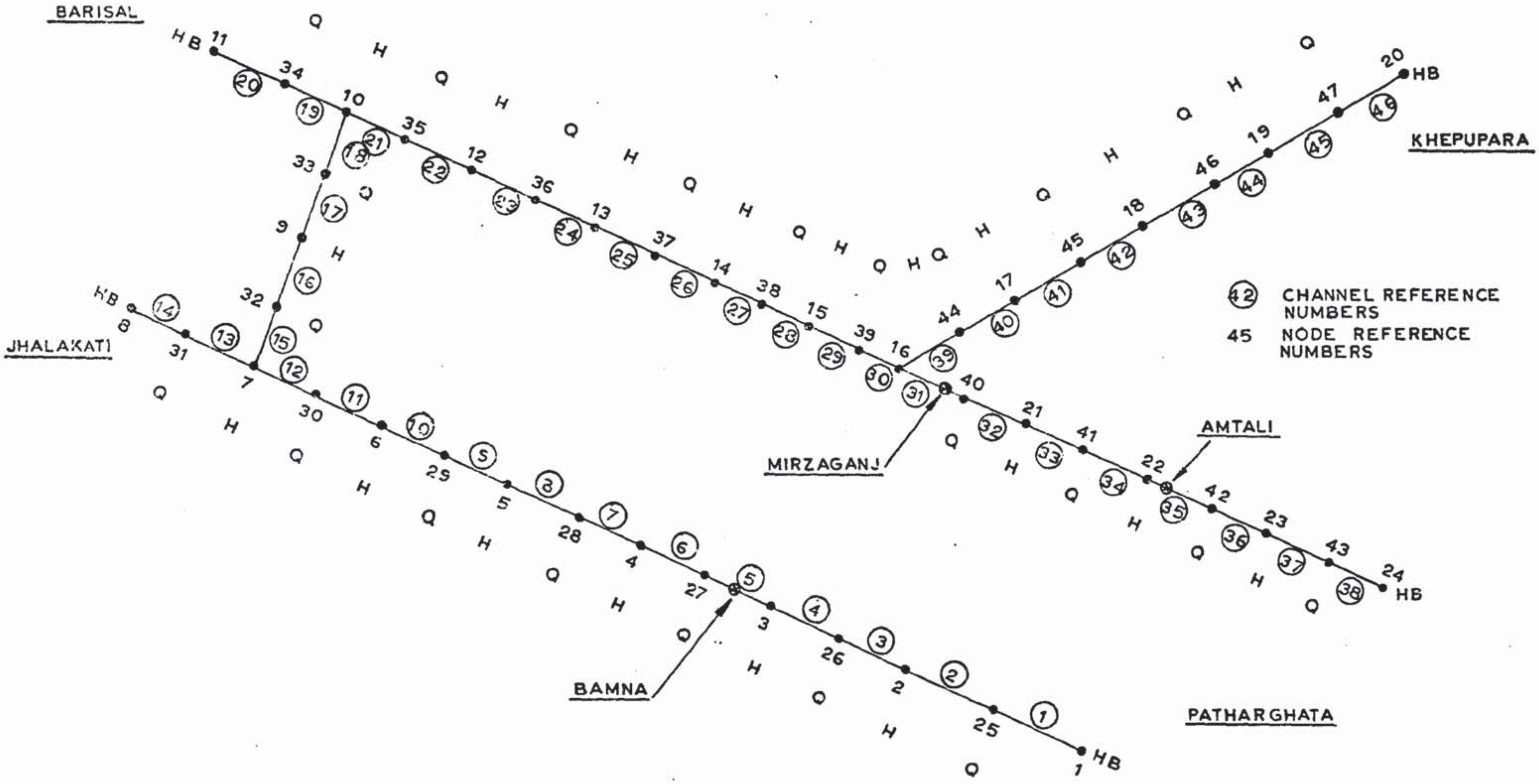
results of these tests will be dealt with later.

7.3 Ganges Delta

The part of the Ganges delta programmed is shown in Fig. 7.5. Fig. 7.6 shows this area in more detail whereas Fig. 7.7 shows the layout of the node and channel system for numerical computation. It can be seen that a total of 47 nodes were used to schematise this network with channel lengths varying from 9,500 ft. to 29,500 ft. The channel cross-section was represented as a rectangle, the breadths of which are shown in TABLE 5. Also shown in this table are the bed levels above water datum and the lengths of the connecting nodes. Fig. 7.8 shows the arrangement of the matrix obtained from the implicit solution of the network.

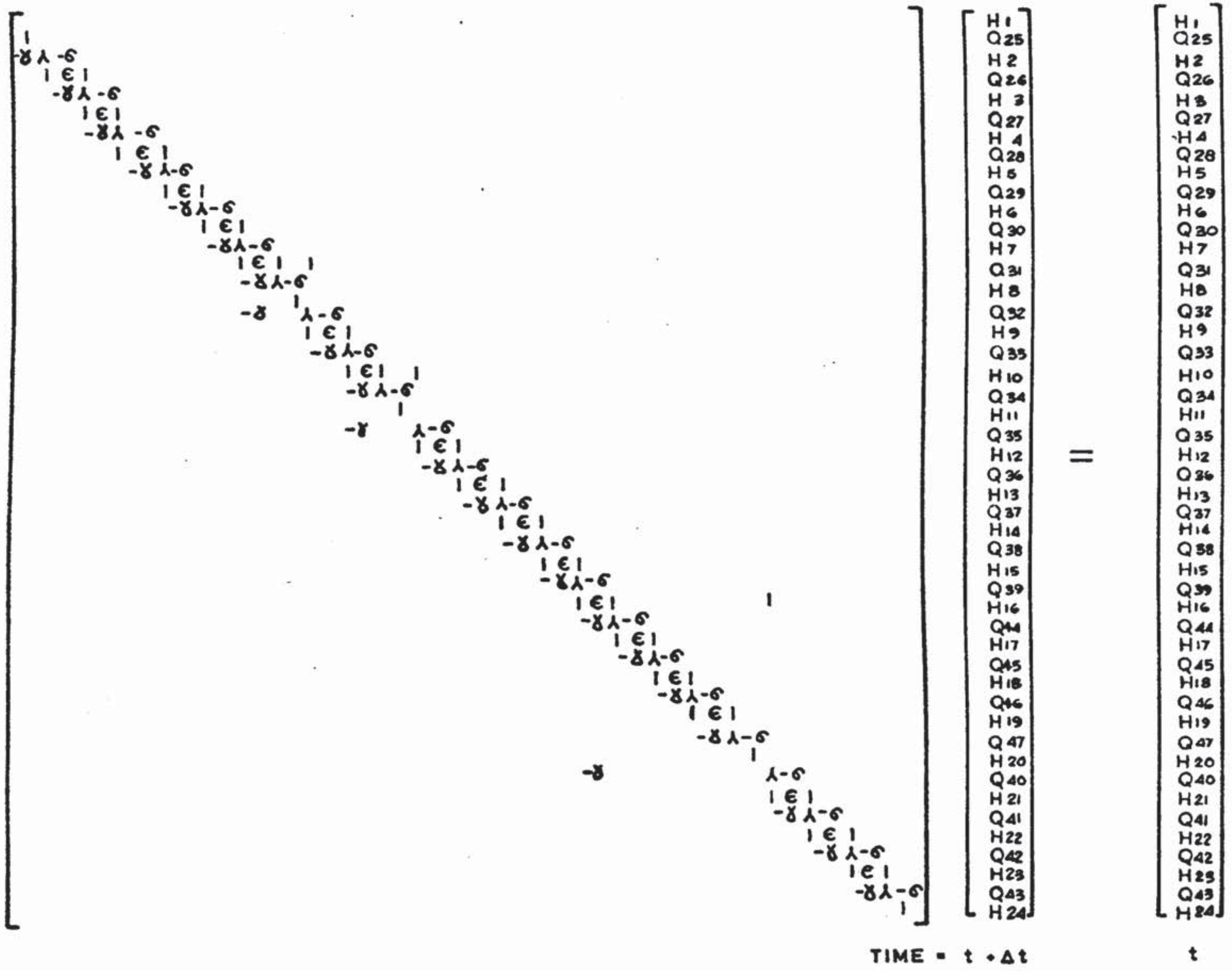
On examination of Fig. 7.7 it can be seen that there are five boundaries all of which are head controlled nodes. The profiles for the tide under study, i.e., that of the 7th February 1968, are shown in Figures 7.9 and 7.10. As no initial conditions were given it was necessary to recycle a program with smoothed boundary data using a tide cycle of 12 hrs. 22½ mins. Again it was found that the 3rd cycle gave the same results as the 2nd and so making further recycling unnecessary. The value of Chezy 'C' recommended to be used was $80\text{ft}^{\frac{1}{2}}$/sec. The initial conditions obtained are shown in TABLE 5. Once these had been obtained subsequent tests on the network were done with the true boundary profiles, i.e., those shown in full in Figures 7.9 and 7.10.

Tests made on this network were similar to those made on the River Aire in that the finite difference systems were compared, with varying time steps, with the recorded water levels. For this purpose actual water levels were known at the stations Bamna, Mirzaganj and Amtali for the tide in question. These locations are also shown in Fig. 7.7.



SCHEMATIC LAYOUT OF PART OF THE TIDAL NETWORK OF THE GANGES DELTA

FIGURE 7.7



NOTE:-
λ, γ, σ AND ε VARY FOR EACH NODE, ALSO THE NODES AROUND HEAD NODES HAVE TO BE MULTIPLIED BY ±,1, DEPENDING ON THE DIRECTION.

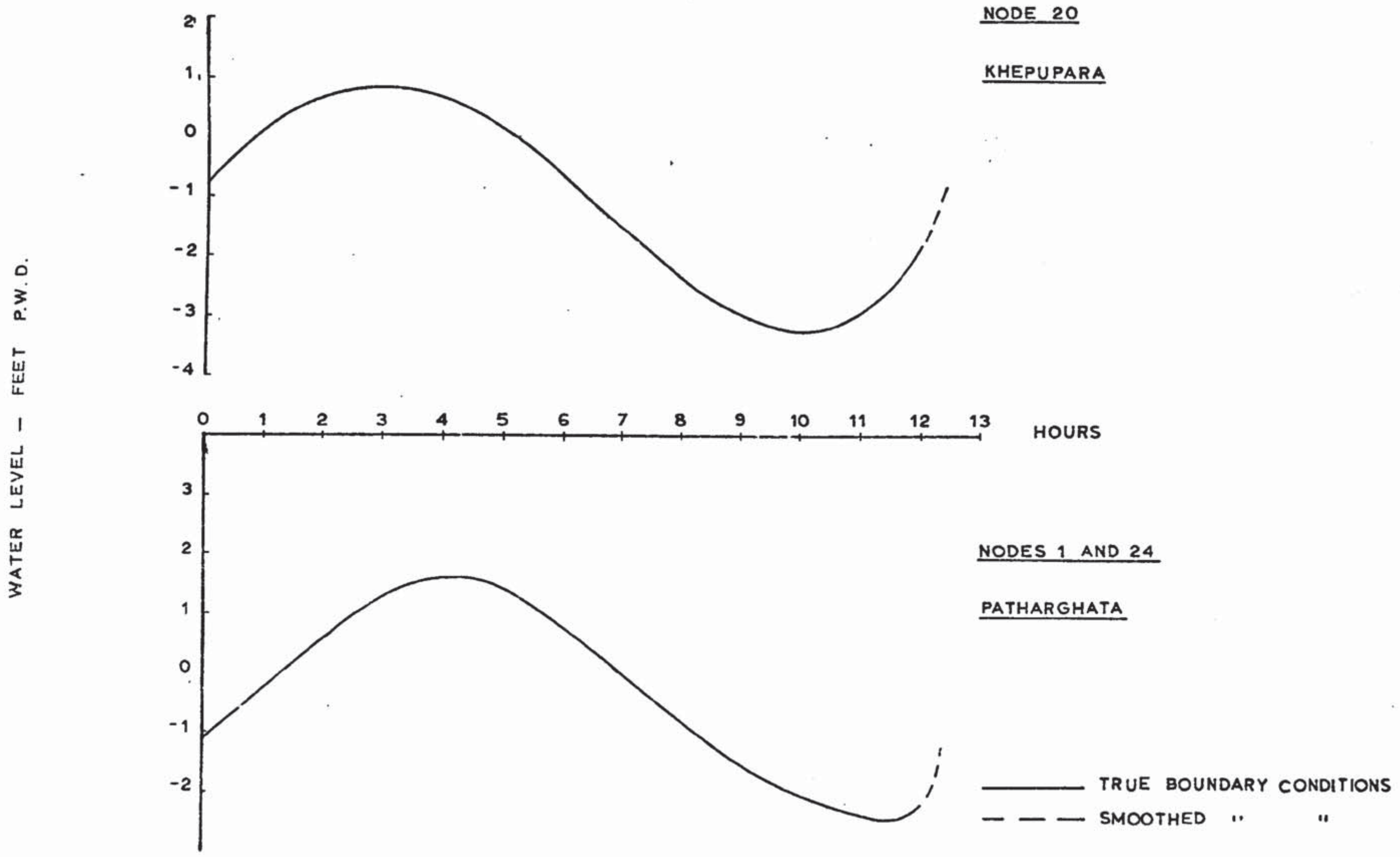
FIGURE 7.8

TABLE 5

NODE	TYPE OF NODE	BREADTH OF CHANNEL (FT)	BED LEVEL P.W.D. (FT)	CON-NECTING CHANNELS	LENGTH OF CHANNEL (FT)	INITIAL CONDITIONS	
						FLOW (FT3/SEC)	WATER LEVEL P.W.D. (FT)
1	HB	4600	-40.0				-1.100
				1	24250		
25	Q	4600				4214	
				2	24250		
2	H	4200	-35.0				-2.038
				3	21050		
26	Q	3800				-71257	
				4	21050		
3	H	3580	-30.0				-2.237
				5	20000		
27	Q	3360				-85672	
				6	20000		
4	H	2940	-27.0				-2.143
				7	22150		
28	Q	2520				-85135	
				8	22150		
5	H	2310	-24.0				-1.585
				9	24250		
29	Q	2100				-77546	
				10	24250		
6	H	2100	-22.0				-0.590
				11	13150		
30	Q	2100				-69151	
				12	13150		
7	H	2000	-20.0				0.294
				13	13150		
31	Q	210				-7405	
				14	13150		
8	HB	210	-30.0				0.650
				15	19750		
32	Q	1900				-53983	
				16	19750		
9	H	2000	-20.0				0.893
				17	15750		
33	Q	2100				-48122	
				18	15750		
10	H	2210	-20.0				1.186
				19	9500		
34	Q	2320				-94062	
				20	9500		
11	HB	2320	-20.0				1.700
				21	23150		
35	Q	1680				53372	
				22	23150		
12	H	1575	-20.0				0.324

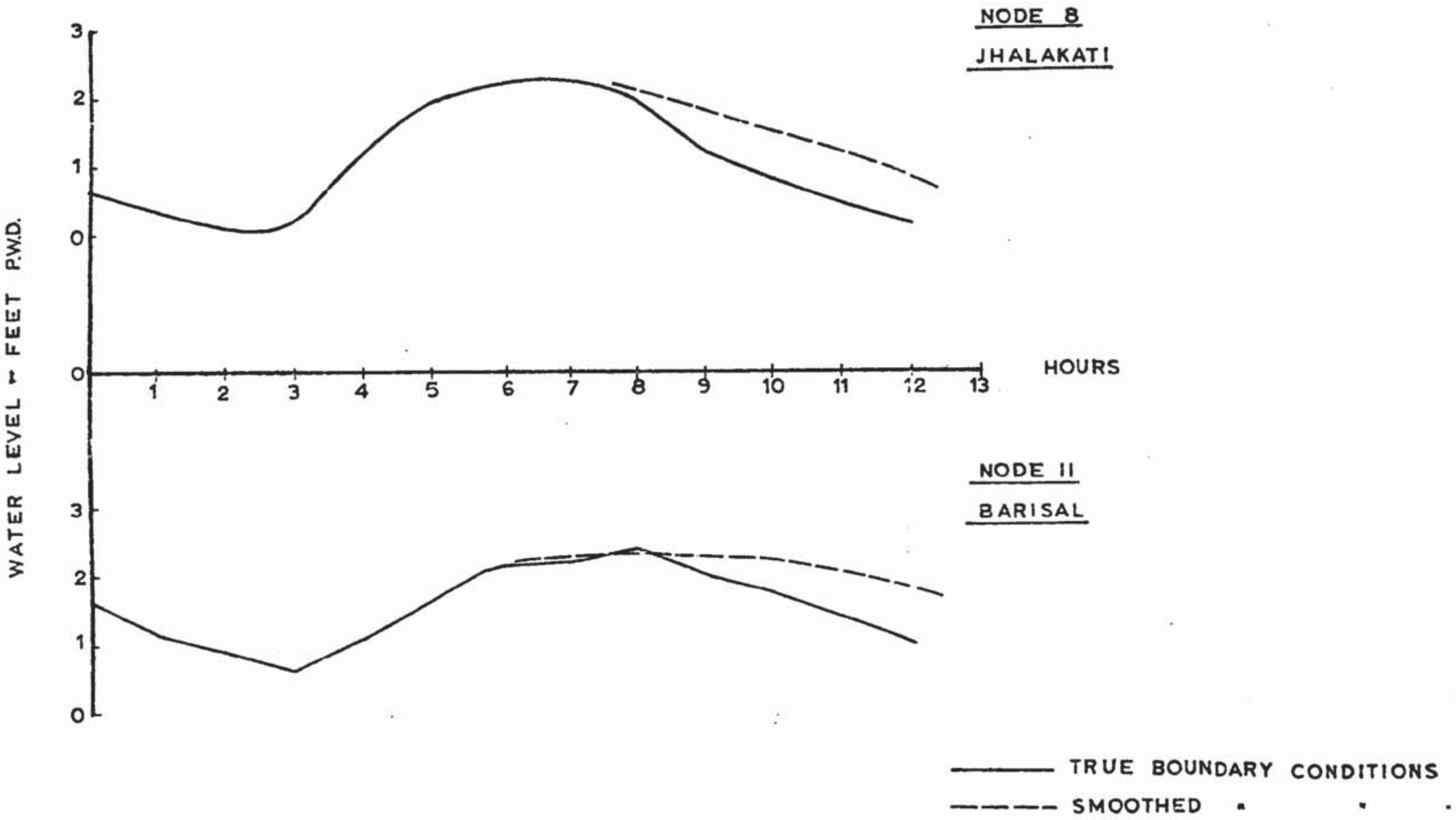
TABLE 5 (CONT'D)

36	Q	1470		23	23150	59842	
13	H	1785	-20.0	24	23150		-1.377
37	Q	2100		25	15800	66471	
14	H	2310	-22.0	26	15800		-2.075
38	Q	2520		27	12600	72361	
15	H	2840	-24.0	28	12600		-2.424
39	Q	3160		29	13700	77697	
16	H	3475	-26.0	30	13700		-2.616
44	Q	2100		39	21050	34053	
17	H	2210	-24.0	40	21050		-2.634
45	Q	2320		41	29500	26134	
18	H	2420	-24.0	42	29500		-2.237
46	Q	2520		43	22100	-22795	
19	H	2625	-30.0	44	22100		-1.579
47	Q	2730		45	17900	-76320	
20	HB	2730	-35.0	46	17900		-0.850
40	Q	3790		31	26400	50312	
21	H	4000	-30.0	32	26400		-2.598
41	Q	4210		33	17900	43636	
22	H	4315	-35.0	34	17900		-2.452
42	Q	4420		35	27400	21871	
23	H	4315	-40.0	36	27400		-1.997
43	Q	4210		37	15800	-74988	
24	HB	4210	-40.0	38	15800		-1.100



BOUNDARY CONDITIONS FOR GANGES DELTA

FIGURE 7.9



BOUNDARY CONDITIONS FOR GANGES DELTA

FIGURE 7.10

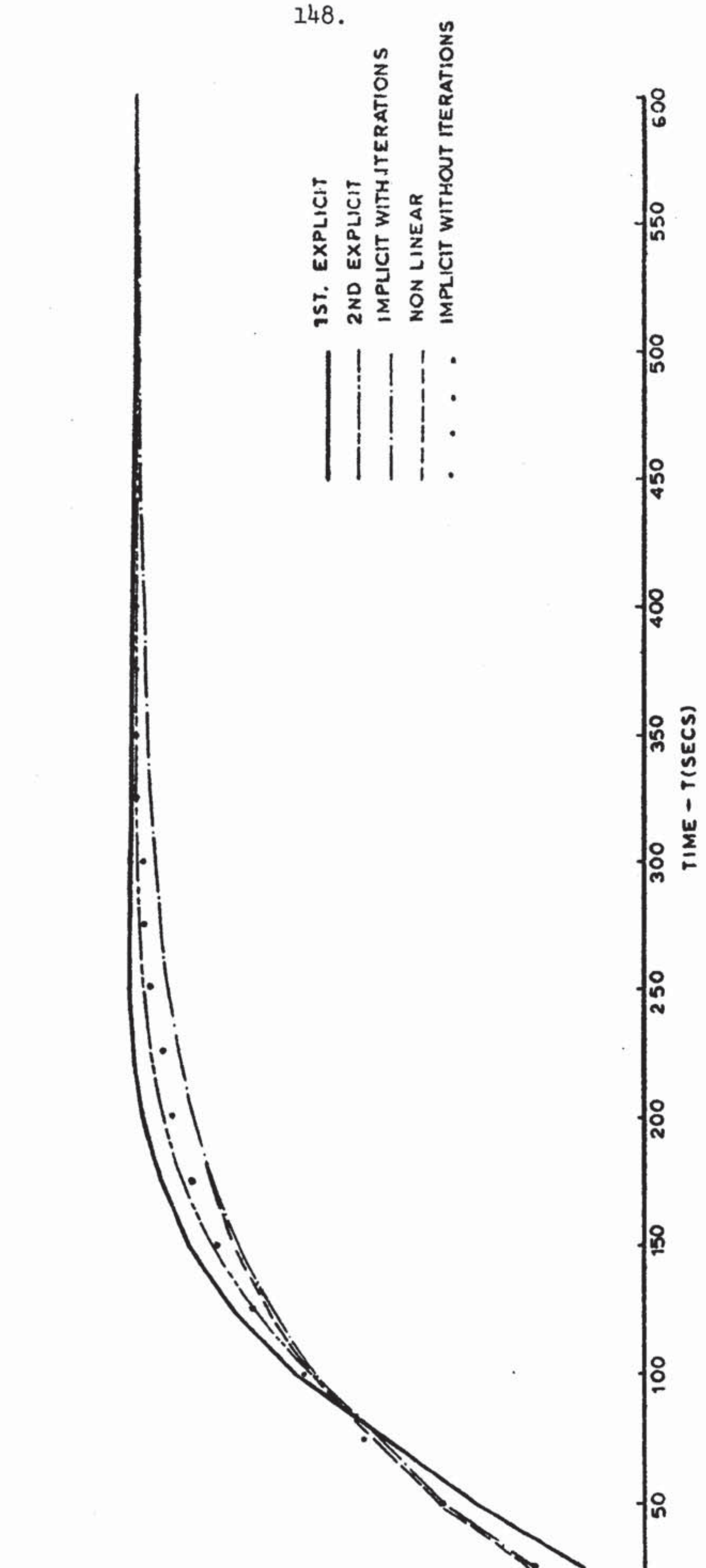
CHAPTER 8

RESULTS

8.0 Introduction

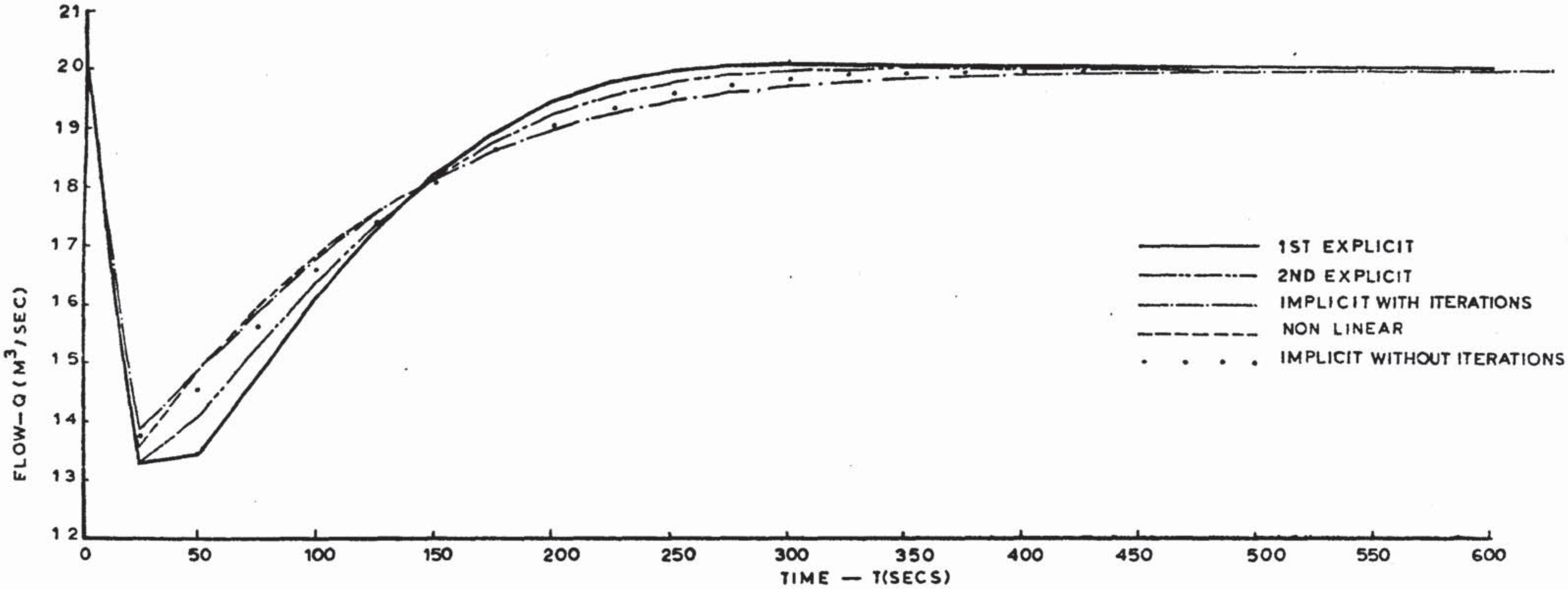
In the following chapter the results obtained from the running of the various numerical systems, described in Chapter 4, on the laboratory and physical models are presented. Comparisons are made, both between the results themselves and, except for the hypothetical model, between recorded results. It is proposed that these comparisons will then lead to recommendations as to which systems may be suitable for the mathematical modelling of unsteady flow situations in open channel networks, in general. Comparisons are made between the systems on the basis of accuracy, economy, general behaviour and stability and convergence.

The degree of accuracy obtained from finite difference systems is primarily a function of the lengths of the difference steps and in this respect the time steps were chosen to be multiples of the Leap Frog stability criteria, eq. (3.2.12). In general section lengths are usually chosen beforehand from experience, with the aims that differences in geometry between consecutive nodes are not great and also that the longitudinal profiles produced will correspond to those recorded. In some instances, as in the Ganges model tested here, then section lengths are determined by the availability of cross-sectional survey data. A further consideration as regards to accuracy is that if time steps are greater than those that can adequately represent the boundary, then again inaccuracy is to be expected. Finally, as mentioned in Chapter 3, it is considered by Dronkers [8] not necessary to update co-efficients in implicit systems, on an iterative basis, as the accuracy obtained is acceptable when considering the inaccuracies inherent in the schematisation of survey data. In this respect modifications were made to the Implicit systems, so that comparisons could then be made between the results from these systems, with constant co-efficients, and with those of the Explicit systems and of the Implicit and Nonlinear systems in which the co-efficients varied.



HYPOTHETICAL MODEL ΔT = 25 SECONDS

FIGURE 8.1



HYPOTHETICAL MODEL ΔT · 25 SECONDS

FIGURE 8.2

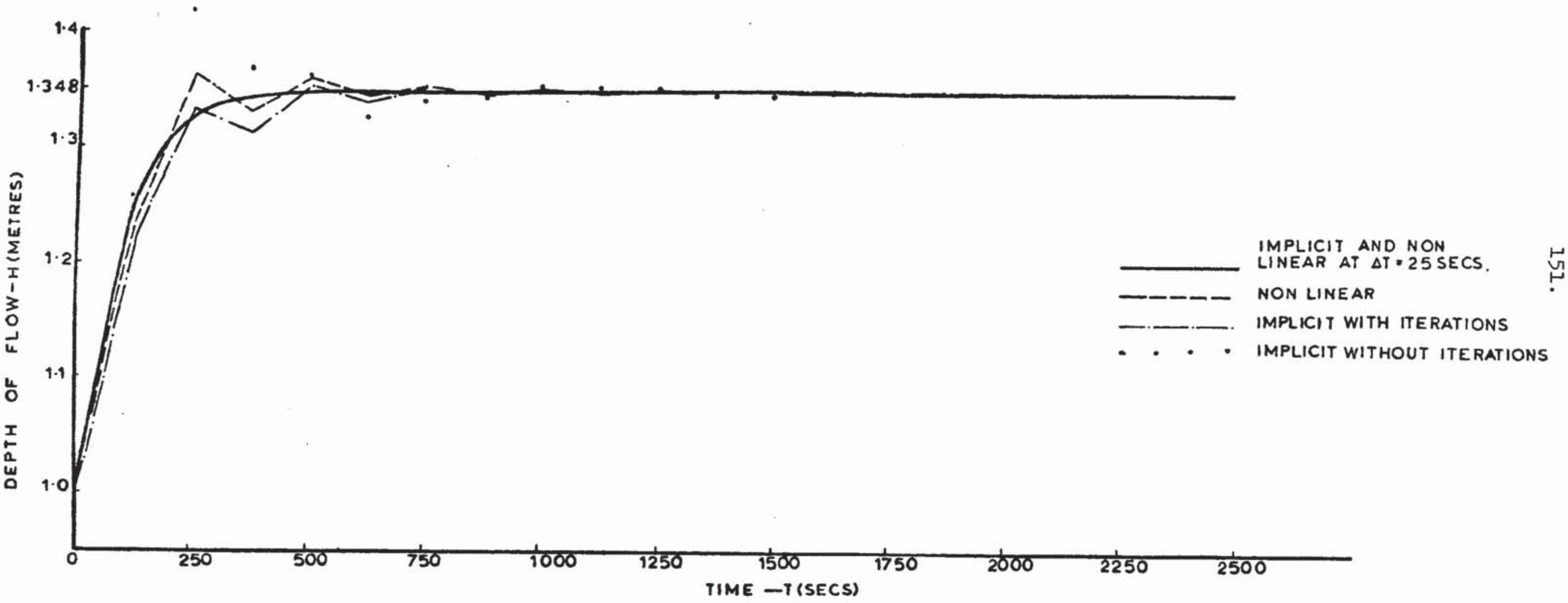
In order to determine the economical aspect then the computer running times were compared for different time steps. Such comparisons may not give an entirely accurate picture as the running times of individual methods depend upon such things (a) the efficiency of the programming used (b) the type of computer and also (c) upon changes in the computer system over the period of time in which results were collected. However, it was thought that such times would illustrate large differences between different methods, and for the same method, illustrate savings in time when using larger time steps. In comparing the running times between explicit, implicit and nonlinear systems then difficulties arise because tolerances are placed on the results of the iterative implicit and nonlinear procedures. The finer the degree of tolerance then the larger is the computer time, whereas results are obtained automatically from explicit systems, by definition.

In the following sections results for each model in turn are discussed, together with the way in which the different systems behaved in general whilst working on each model. Finally, in the last section the numerical times at which the systems either became unstable or did not converge to solutions for each model are discussed in relation to Chapter 5.

8.1 Hypothetical Model

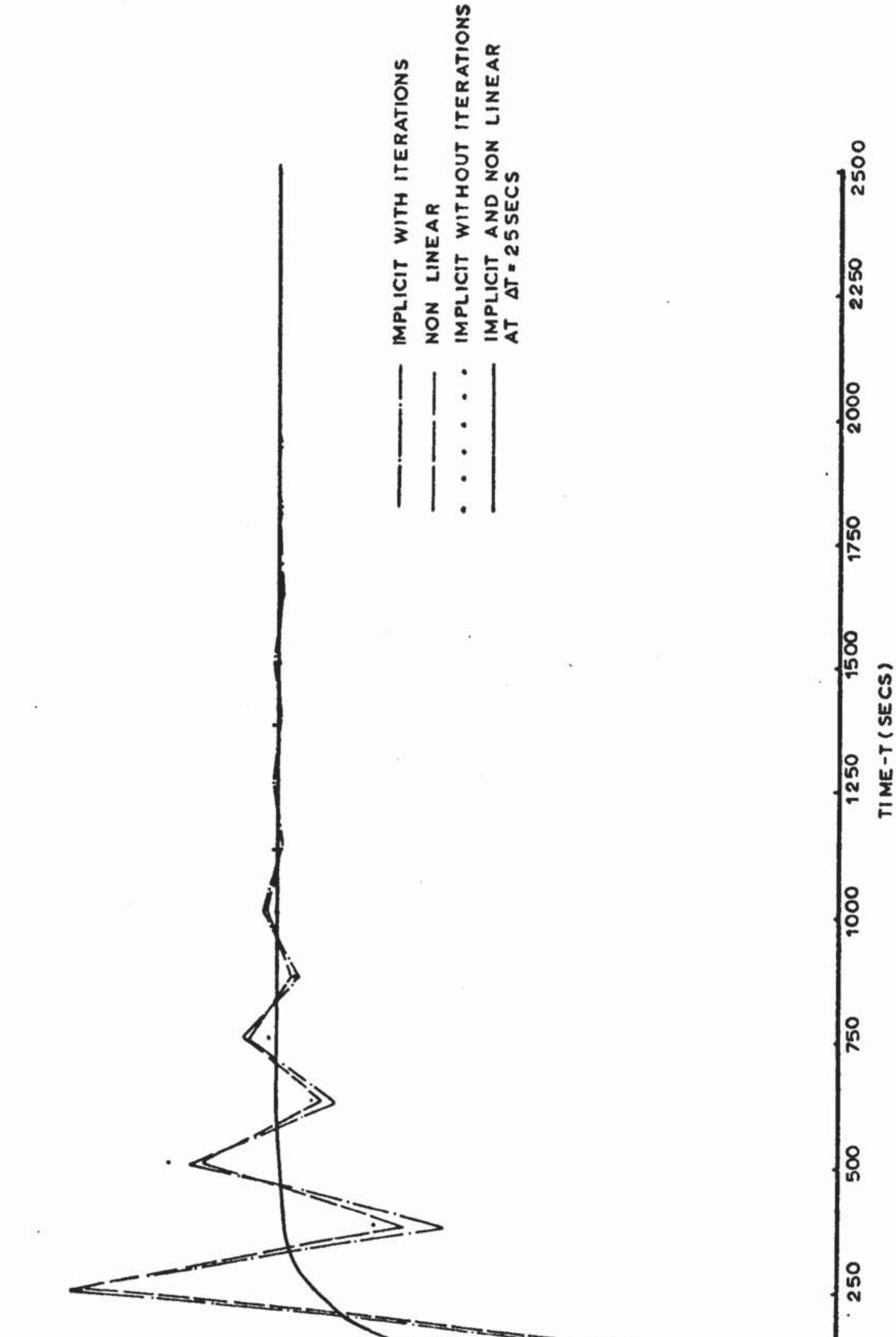
The purpose behind programming this model was, as stated previously, to "iron-out" programming errors whilst developing the different numerical systems. However the results from conducting a series of tests on this model, do provide information on the stability and convergence properties of the systems together with insight into their general behaviour. To illustrate these points, results from nodes 2 and 3 (see Fig. 7.1) were plotted, as it was these nodes that were subject to the greatest amount of variation with time.

Figures 8.1 and 8.2 show the results for nodes 2 and 3, for the time step of 25 seconds, in which results for all of the systems are plotted, i.e., Explicit, Implicit and Nonlinear. On examination of Fig. 8.2 it can be seen that at the end of the first time step the flows are retarded owing to the



HYPOTHETICAL MODEL ΔT = 125 SECONDS

FIGURE 8.3



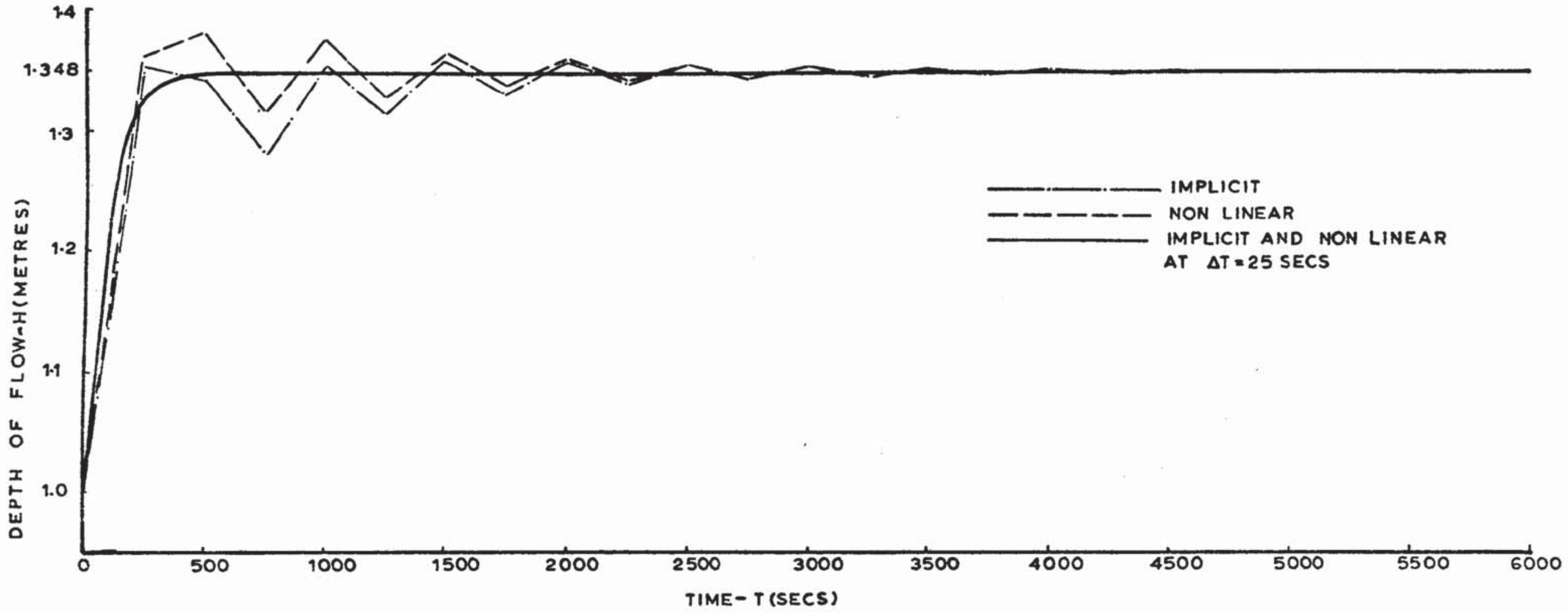
HYPOTHETICAL MODEL ΔT- 125 SECS

FIGURE 8.4

channel friction (it is noted that the normal flow for a depth of 1m with the stated slope and friction is 13.13 M^3/sec). Then, owing to the influence of the constant boundary value of 20 M^3/sec., it can be seen that there is a gradual increase in depth of flow (see Fig. 8.1) and quantity of flow until the steady state condition was reached. With the increase in depth at node 2 then the effect of the boundary was also gradually transmitted downstream until the entire length reached a steady state condition. This consisted of depths of 1.348 M, which as stated in Chapter 7 is the calculated normal depth of flow, for nodes 2 to 12, and then a short length of backwater curve up to the boundary node 20, where the depth was held constant at 1 M. Figures 8.1 and 8.2 show that all of the systems converged well to the steady state with perhaps, the Implicit system with varying co-efficients and the Nonlinear system converging at a slightly slower rate than the others.

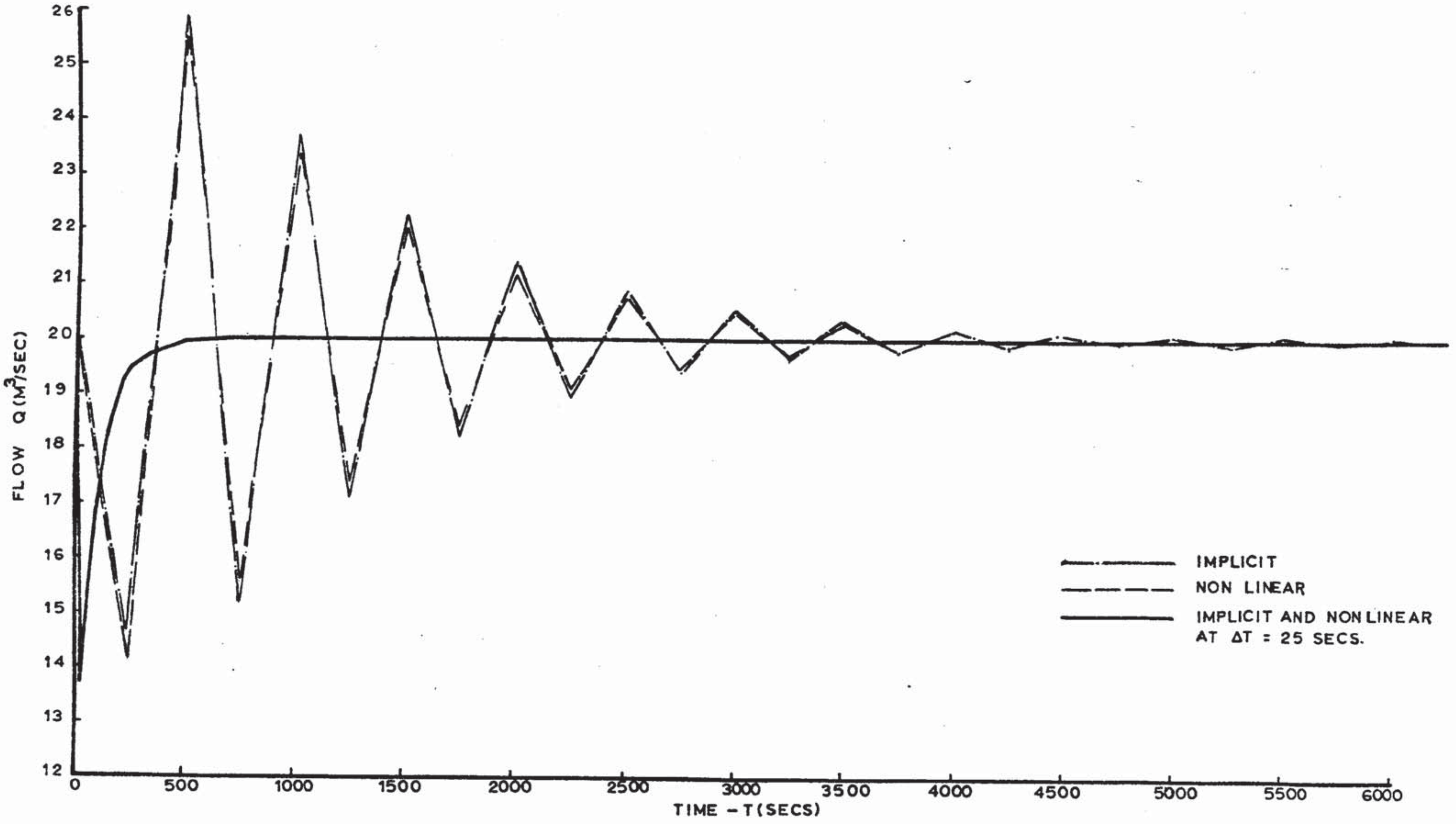
Figures 8.3 and 8.4 show the results obtained for the time step of 125 seconds. Only results from the Implicit and Nonlinear systems are plotted as the Explicit systems become unstable at time steps less than this. Also shown on these figures are the results obtained from the Implicit system with varying co-efficients and Nonlinear system for the time step of 25 seconds, for comparison purposes. The small differences that were present on the "rising limbs" of these results were averaged. On examination of Figures 8.3 and 8.4 it can be seen that oscillations in the results are present owing to the use of the relatively large time step. However, after a period of time, the results converged to the steady state conditions for the entire reach.

In Figures 8.5 and 8.6 only the results for the Implicit system with varying co-efficients and the Nonlinear system are plotted as the Implicit system with constant co-efficients became unstable at a smaller time step. Also plotted on these figures are the results from the Nonlinear system and Implicit system with varying co-efficients for the time step of 25 seconds. On examination it can be seen that the oscillations are greater and more persistent than those for the time step of 125 seconds, but again they converge to the steady



HYPOTHETICAL MODEL ΔT = 250 SECONDS

FIGURE 8.5



HYPOTHETICAL MODEL ΔT 250 SECONDS

FIGURE 8.6

state conditions.

Figures 8.5 and 8.6 show that there is little difference in the results obtained from the Nonlinear and Implicit schemes, with perhaps the Nonlinear programs producing slightly less amplitude in oscillation. Figures 8.3 and 8.4 show that the results from the Implicit system with constant co-efficients had larger oscillations initially but, as the others, they converged to the steady state.

TABLE 6 shows, for each numerical system, the maximum number of iterations required for the systems to converge to solutions, for the time steps shown. Also shown are the time steps at which the systems either became unstable or did not converge to solutions at all. In order to make comparisons with the iterative procedures then an equal tolerance limit must be used and this was arbitrarily set at a value of 10^{-3} for the Implicit Methods and 10^{-6} on the norm for the Nonlinear Methods.

The performance of each of the numerical systems, both with reference to the previously mentioned figures and with TABLE 6, will now be dealt with in turn:

Explicit

The two explicit systems are shown to provide good results, in that they follow the general trend of the results from the other systems and converge to the steady state well, for the time step of 25 seconds. This time step is below the numerical points of instability of the First and Second Explicit systems which were found to be 55 and 111 seconds respectively.

Implicit, Gauss-Seidel

Results from this program were obtained for the 25 second and 125 second time steps but at 150 seconds the system failed by not converging to a solution. It can be seen that the maximum number of iterations required to reach solutions rose sharply from 8 at $\Delta t = 25$ seconds to 31 at $\Delta t = 125$ seconds. At a time step of 147 seconds, i.e., prior to the system failing at $\Delta t = 150$ seconds it required, at one stage 98 iterations to reach a solution.

TABLE 6

NUMERICAL SYSTEM		TIME STEP (SECONDS)	MAXIMUM NUMBER OF ITERATIONS	APPROXIMATE TIME STEP AT WHICH SYSTEM FAILED
EXPLICIT	FIRST	25	1	55
	SECOND	25	1	111
IMPLICIT (with varying co-efficients)	GAUSS-SEIDEL	25	8	150
		125	31	
	DOUBLE-SWEEP	25	4	375
		125	10	
		250	27	
	SPARSE-SWEEP	25	4	375
		125	10	
		250	27	
IMPLICIT (with constant co-efficients)	DOUBLE-SWEEP	25	1	225
		125	1	
	SPARSE-SWEEP	25	1	225
		125	1	
NONLINEAR	DOUBLE-SWEEP	25	4	NOT REACHED
		125	4	
		250	5	
		500	7	
		1000	8	
		2500	7	
	SPARSE-SWEEP	25	4	NOT REACHED
		125	4	
		250	5	
		500	7	
		1000	8	
		2500	7	

Implicit, Double-Sweep

Two variations of this method were tested, the first as described in Chapter 4, was with the matrix $\bar{A}$ in eq. (4.4.10) being updated on an iterative basis, and the second was with this matrix held constant. Results from the first variation show that the system provided solutions up to the time step of 375 seconds after which the method failed by not converging to a solution. At a time step of 350 seconds this method required at one stage, 200 iterations to converge to a solution. However, at smaller time steps the system can be seen to require few iterations. The second variation of this method can be seen to fail at a time step of 225 seconds by the system becoming unstable.

Implicit, Sparse-Sweep

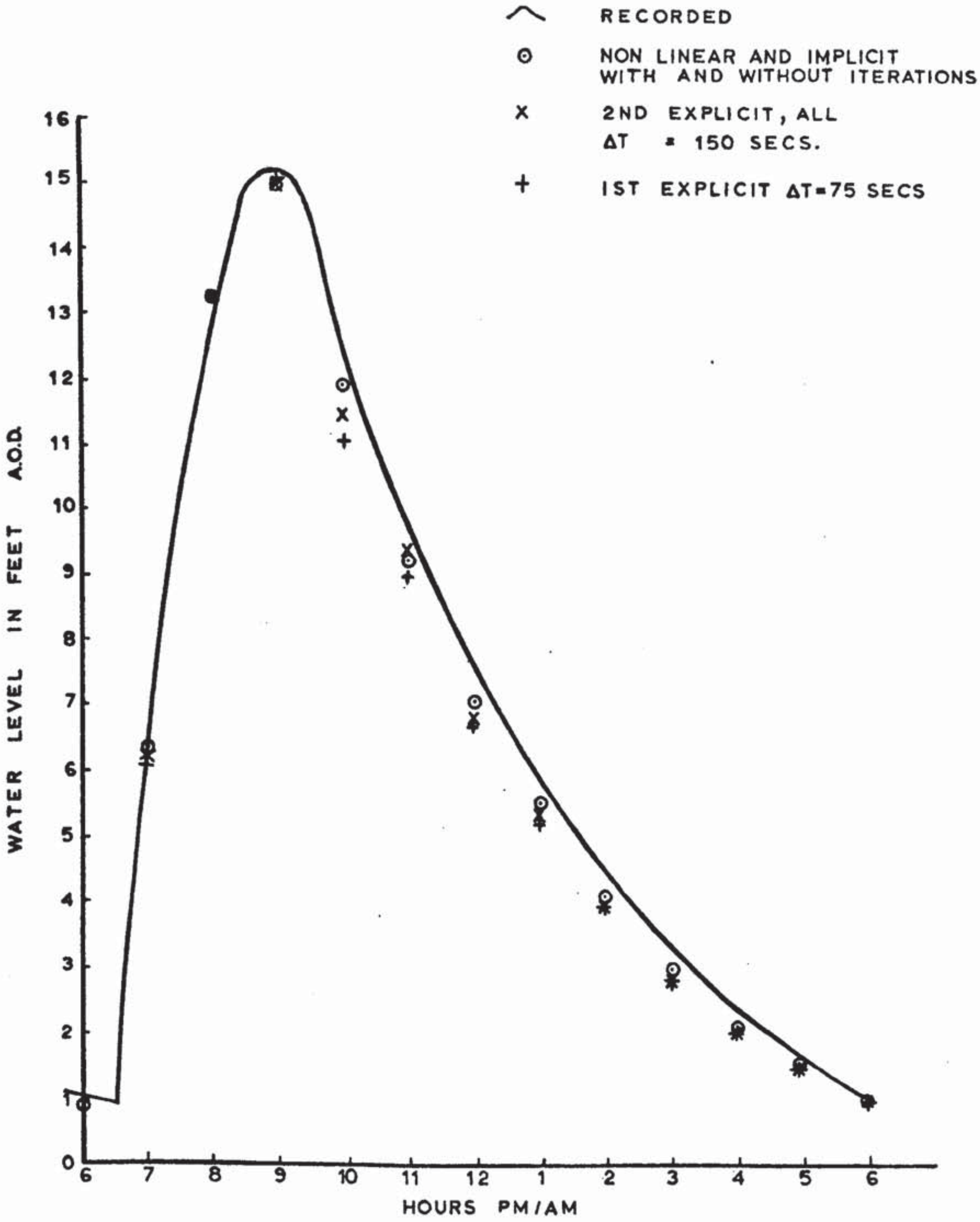
Again the two variations described above were tried with this method. As this method of inversion of the matrix $\bar{A}$ is identical to the above double sweep method, for tridiagonal structures, then it is expected that their behaviour for this model will be the same and on examination of TABLE 6 this is seen to be true.

Nonlinear

Both of the variations of this method described in Chapter 4 are considered here together as, as to be expected, the results are seen, on examination of TABLE 6, to be identical. Results were obtained for time steps up to 2500 seconds, i.e., 100 times greater than the basic time step of 25 seconds. Further increases beyong this point were thought unnecessary. Also, the table shows just how few iterations were required at these large time steps and clearly show the effective convergence characteristics of this method. In both of the variations, and for all the time steps used, it was not necessary for the programs to enter the procedures which chose values of t_i less than one in order to reduce the norm. Thus the full Newtonian step of $t_i = 1$ was used throughout.

8.2 River Aire

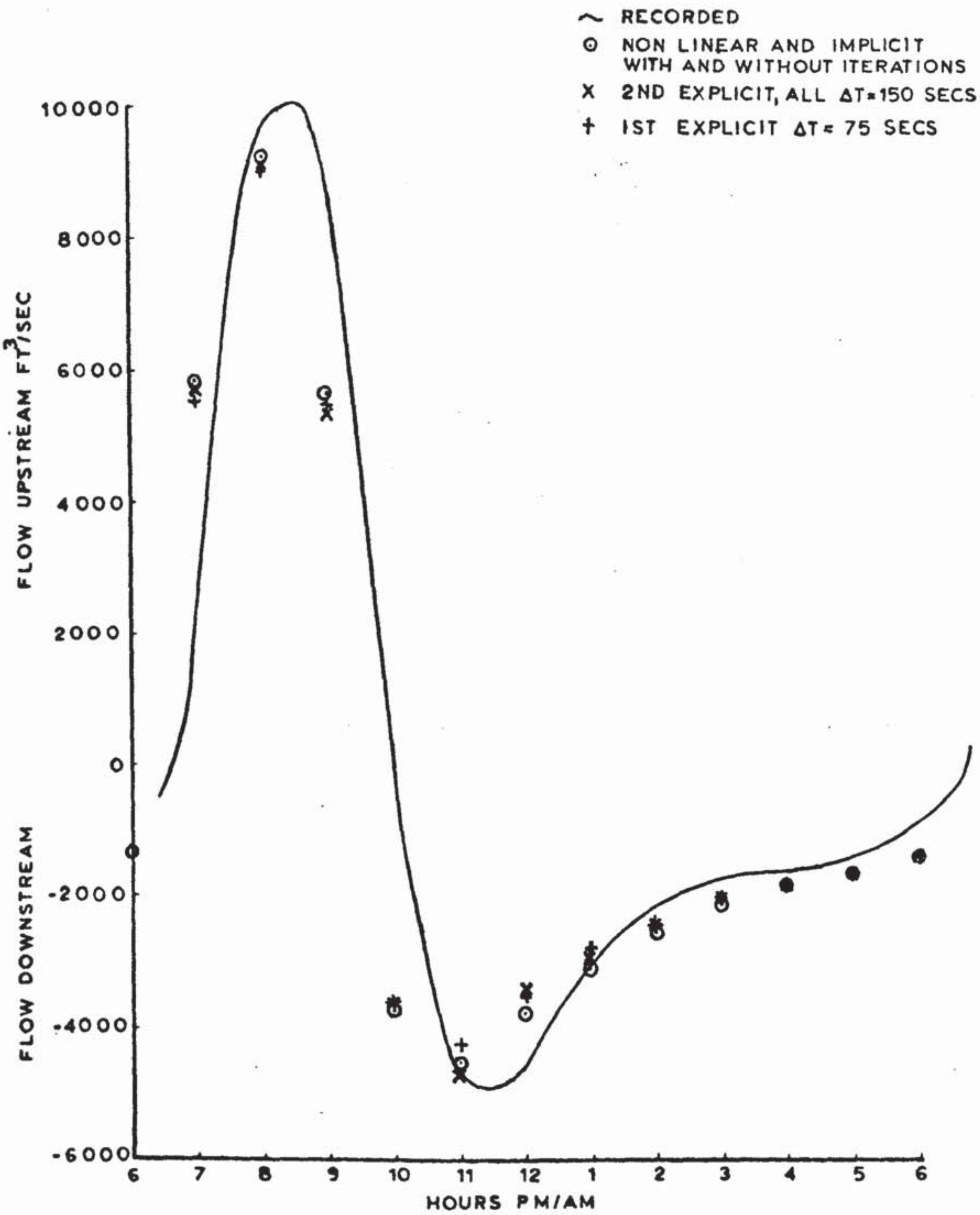
In order to show and compare the results obtained for the River Aire model



DEPTH OF FLOW AT NEWLAND—NODE 5

RIVER AIRE

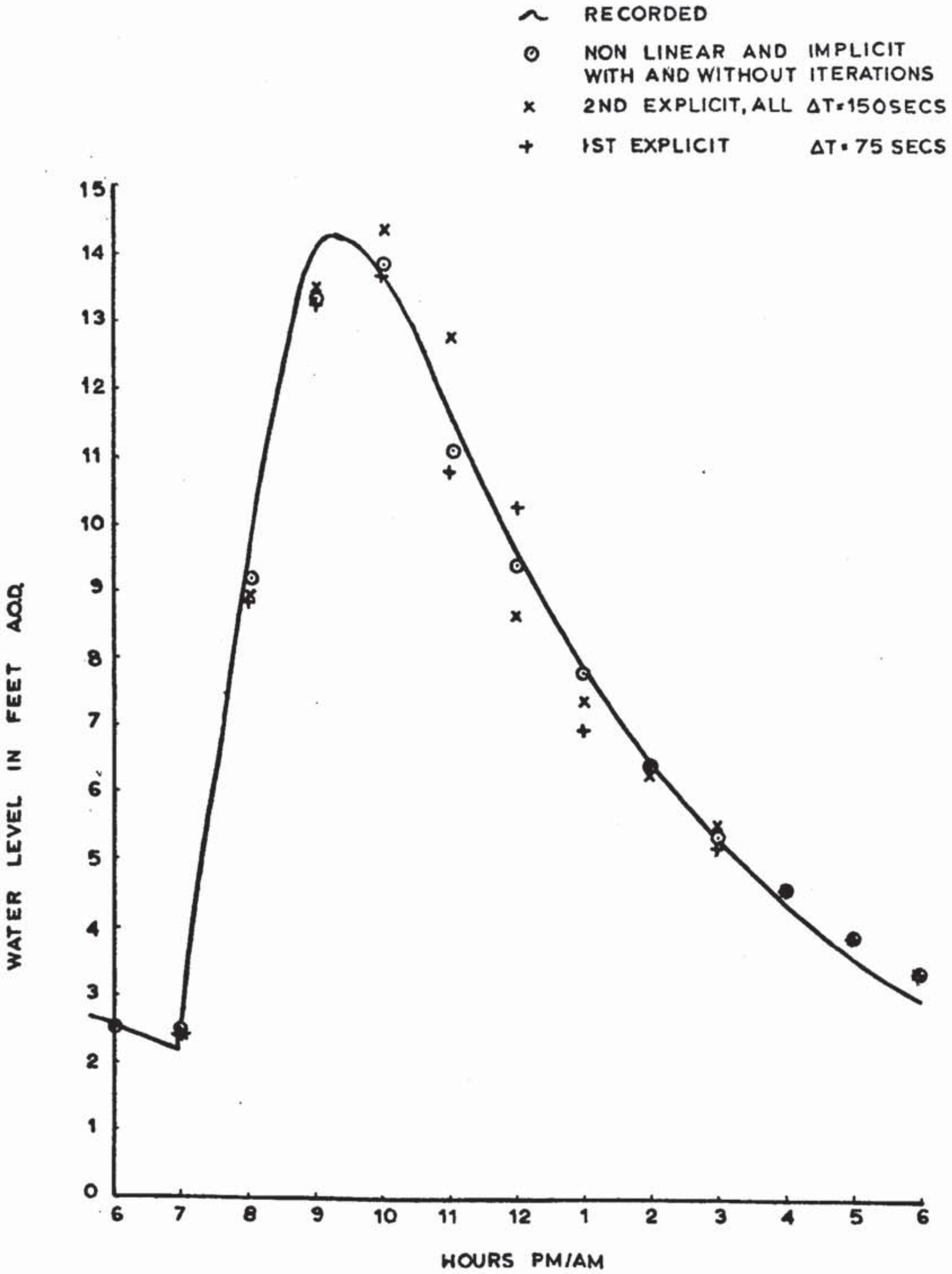
FIGURE 8.7



QUANTITY OF FLOW AT NEWLAND - NODE 5

RIVER AIRE

FIGURE 8.8



DEPTH OF FLOW AT CARLTON – NODE 11

RIVER AIRE

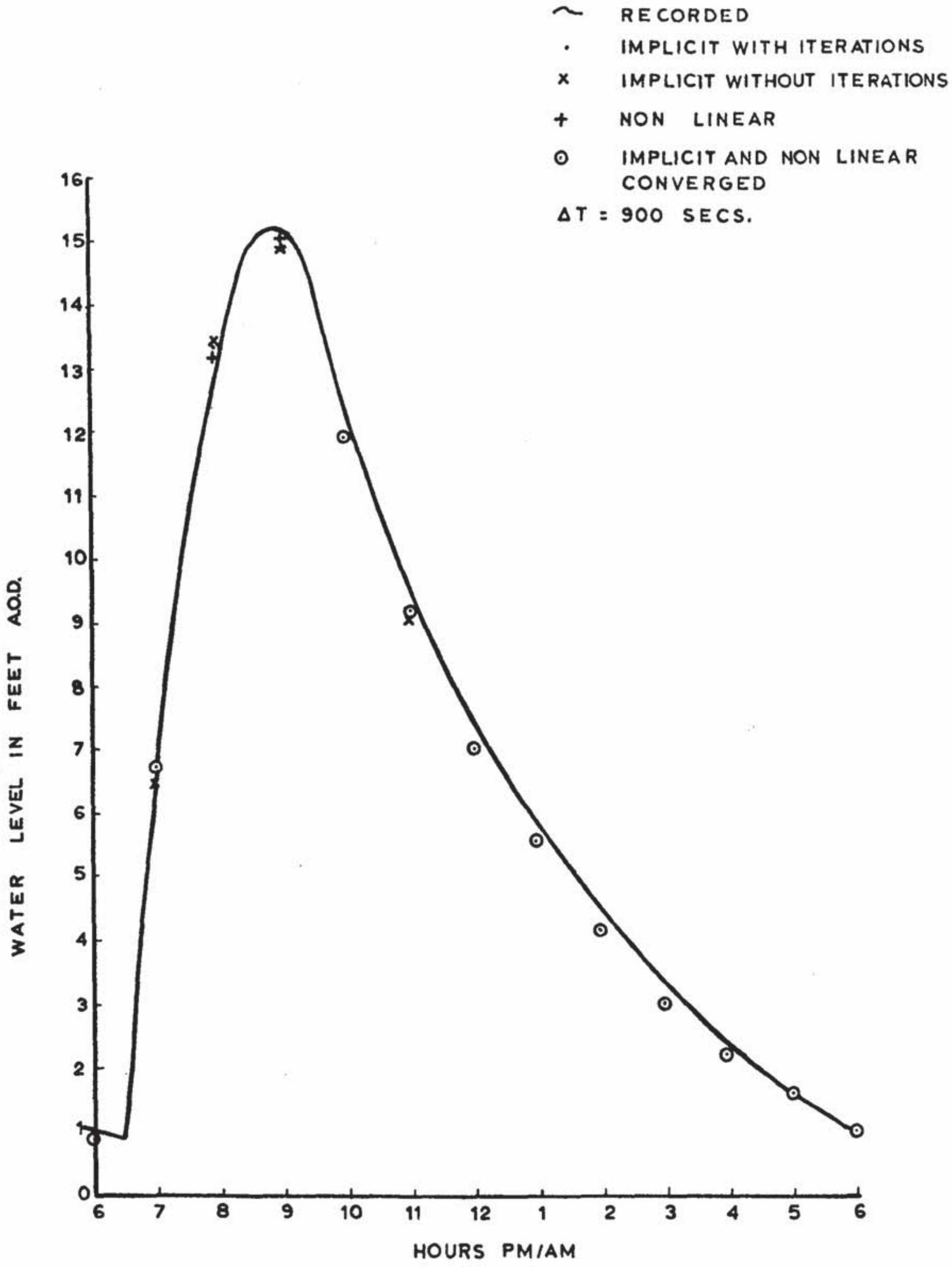
FIGURE 8.9

then these are presented for the two nodes 5 and 11, situated approximately one third and two thirds along the reach programmed. On examination of Fig. 7.3 these are seen to coincide approximately with the positions Newland and Carlton. At both of these positions the depths of flow have been plotted and also at Newland the discharge results are plotted, obtained by interpolation between nodes 4 and 6.

Figures 8.7, 8.8 and 8.9 show the results for the two nodes for the time step of 150 seconds. On these figures, however, it was necessary to show the results from the First Explicit system with the time step of 75 seconds as this system became unstable before 150 seconds. On examination of the figures it can be seen that there is in general, good agreement between the calculated depths from the Nonlinear and Implicit systems with those recorded. The comparison between the calculated discharges and those recorded is however, not quite as good. This may be due to the fact that on the chart from which the recorded discharge was obtained, it was difficult to relate the start of the tide with the time scale used. A shift of the results to the right of approximately ½ hour would appear to give better agreement. It can be seen from Fig. 8.7 that the calculated points have "run-out" quicker than those recorded and although a smaller value of Chezy 'C' was used for the downstream flow than the flow upstream, it is thought that a smaller value still, could have been used. This would then have the effect of retarding the flow. Further, no allowance was made for the presence of bridge piers at Carlton, which would also have the effect of retarding the flow.

As regards to the results obtained from the Explicit systems it can be seen that there is good agreement between these and those obtained from the Nonlinear and Implicit systems at the beginning and ends of the curves, but in the centre, where the depths are greater, there is a large amount of divergence. This, as will be explained later, is due to instability "creeping in" at the deeper depths.

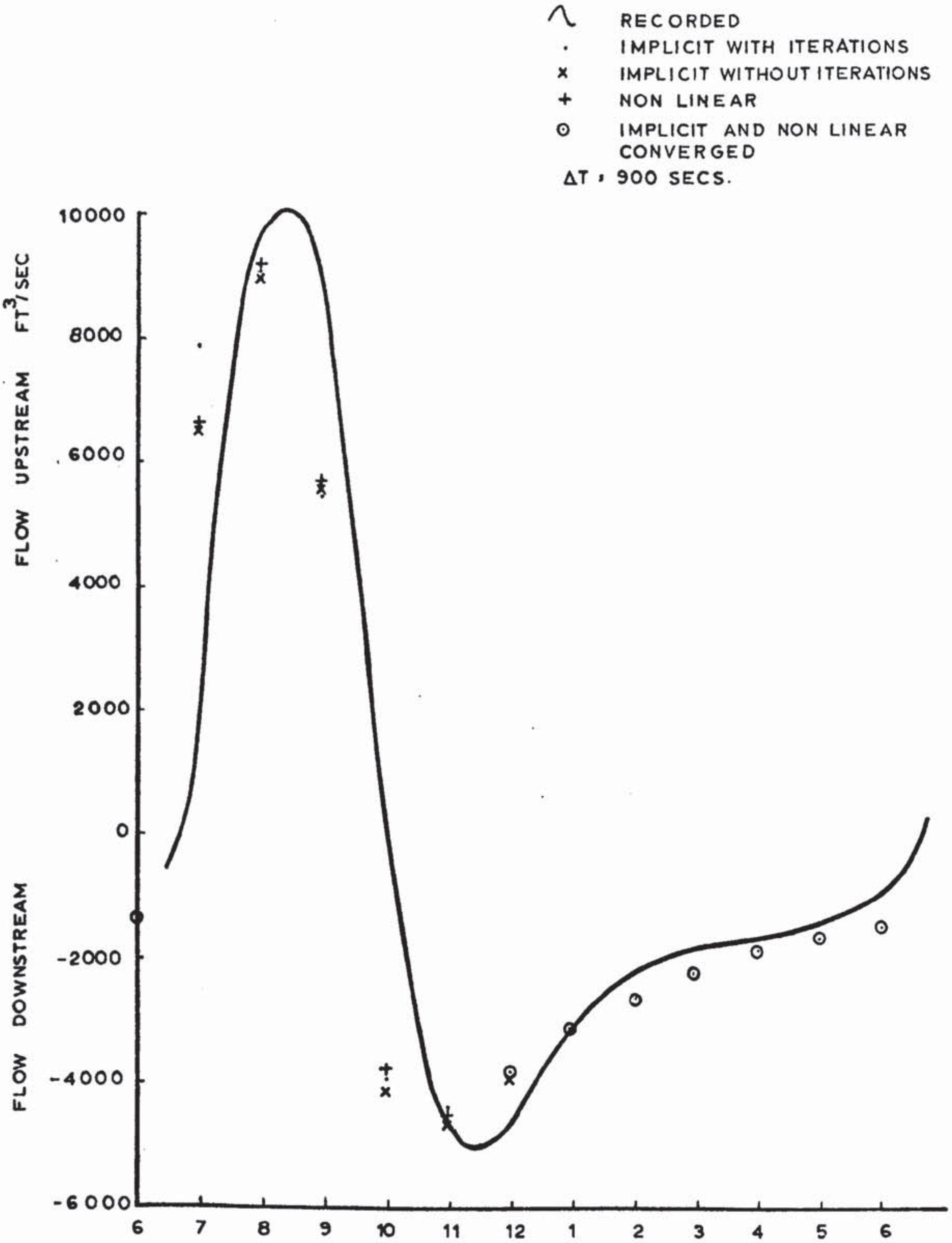
Figures 8.10, 8.11 and 8.12 show the results obtained when a time step of



DEPTH OF FLOW AT NEWLAND - NODE 5

RIVER AIRE

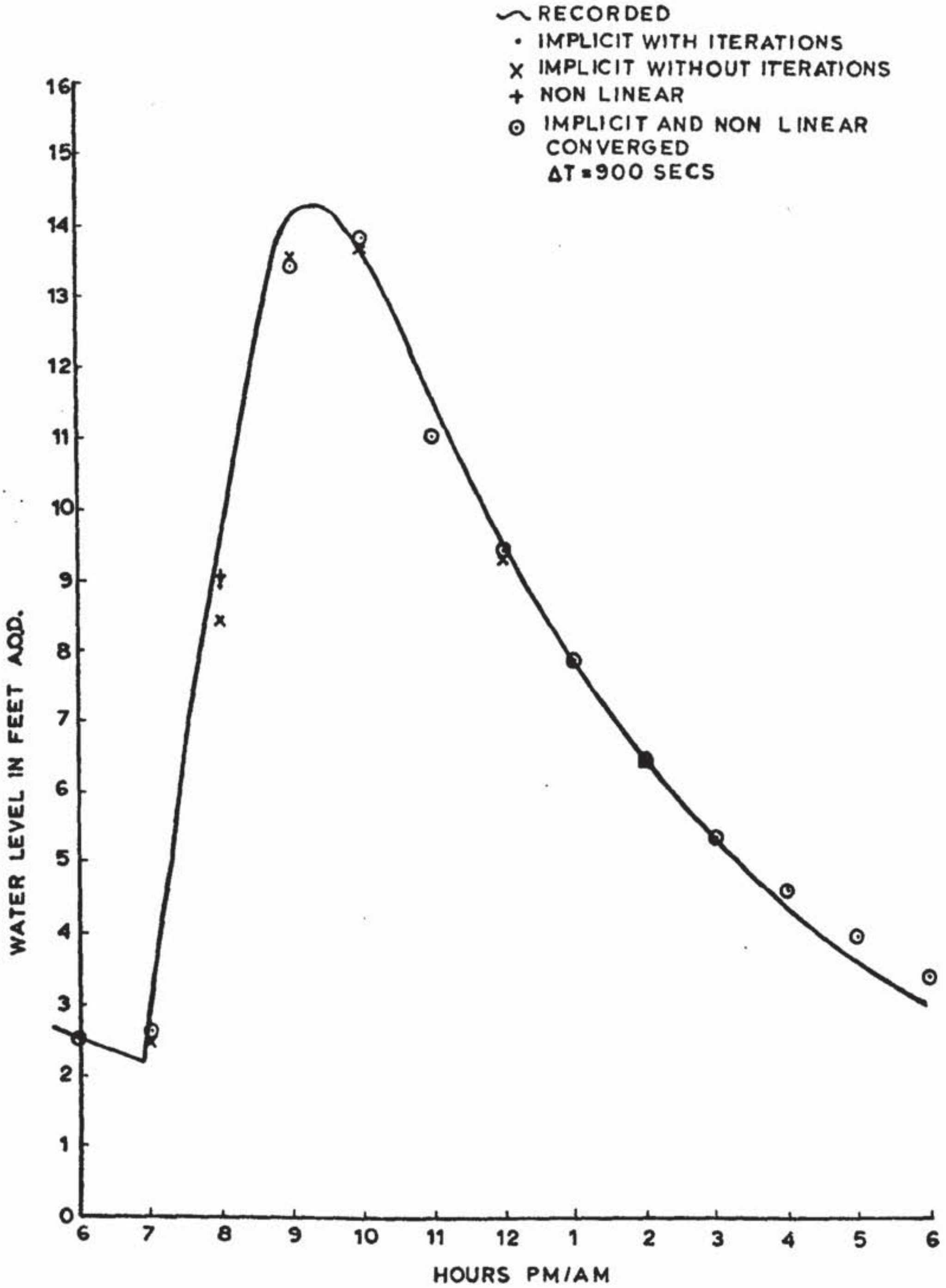
FIGURE 8.10



QUANTITY OF FLOW AT NEWLAND – NODE 5

RIVER AIRE

FIGURE 8.11



DEPTH OF FLOW AT CARLTON-NODE 11

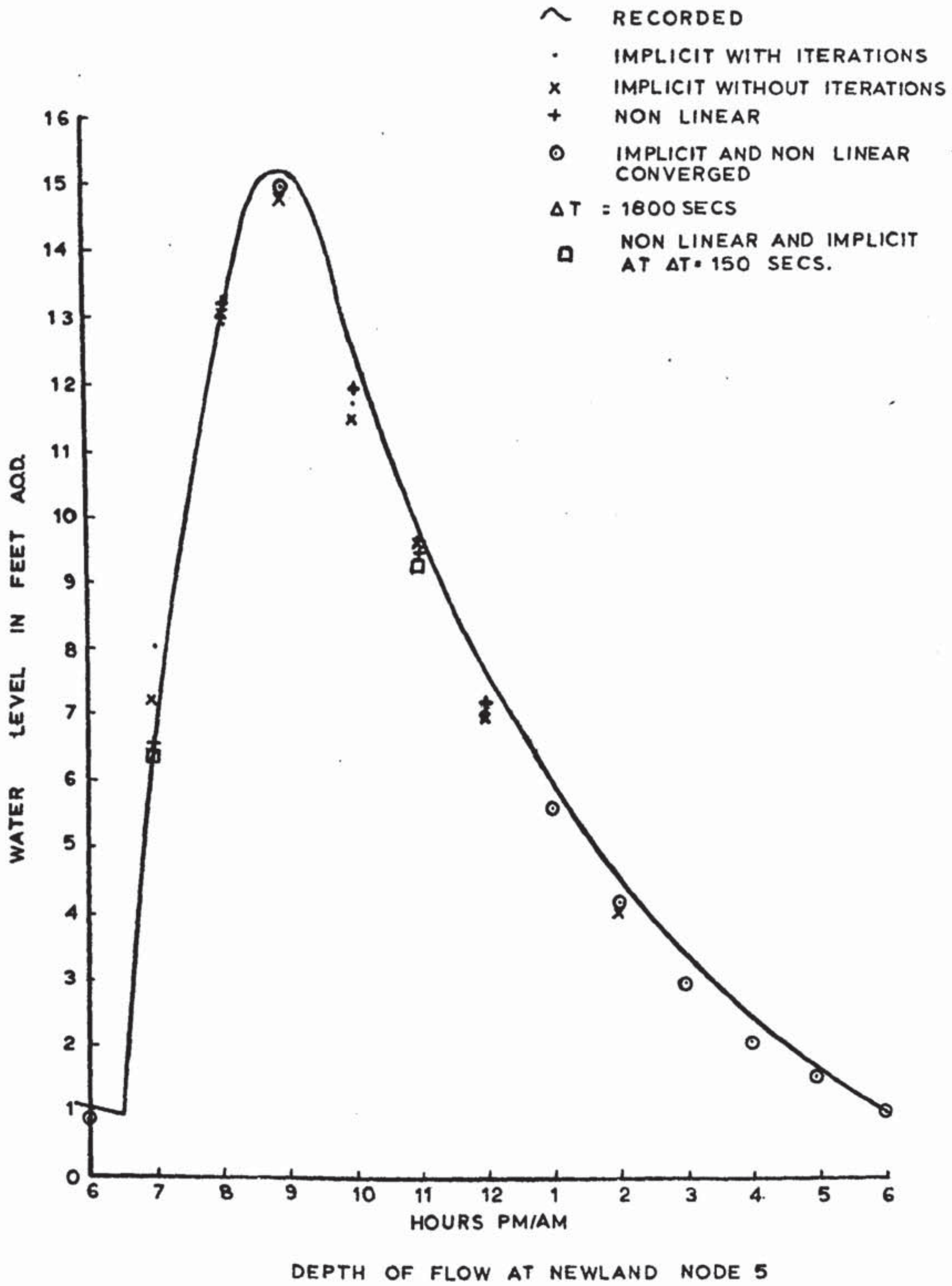
RIVER AIRE

FIGURE 8.12

900 seconds was used. It can be seen that on the whole there is little divergence between the calculated values and those recorded. However, there is a small amount of divergence between the results from individual methods and it will be noticed that the results from the Nonlinear programs follow most closely the recorded values. Further, it is evident that a small amount of oscillating is apparent in the results from the Implicit systems. As it is difficult to show individual points that coincide on the figures, the convention adopted was to use a circle with a dot in the centre where a Nonlinear result and either or both, of the Implicit results coincided.

The calculated depths and discharges for a time step of 1800 seconds are shown in Figures 8.13, 8.14 and 8.15. On examination of these figures it can be seen that the difference between individual methods has become greater during the periods of high rise and fall of depth. It is beginning to appear that the Nonlinear Method gives results that are consistently closer to those recorded than the others. Surprisingly, the Implicit Method with varying coefficients shows a greater amount of divergence from the recorded and Nonlinear results than those results from the Implicit system with constant coefficients, particularly with the discharge. Also shown on these figures are the calculated depths and discharges obtained from the Nonlinear and Implicit systems at the time step of 150 seconds. To avoid congestion these are shown only where they differ appreciably from the results obtained from the Nonlinear method at the ½ hour time step. When comparing the above results it must be remembered that the time step of ½ hour overshoots the sharp rise in boundary depth at node 1, occurring at 6¼ hours p.m. (see Fig.7.4). Because of this, some differences must be expected on the rising curves, particularly at node 5.

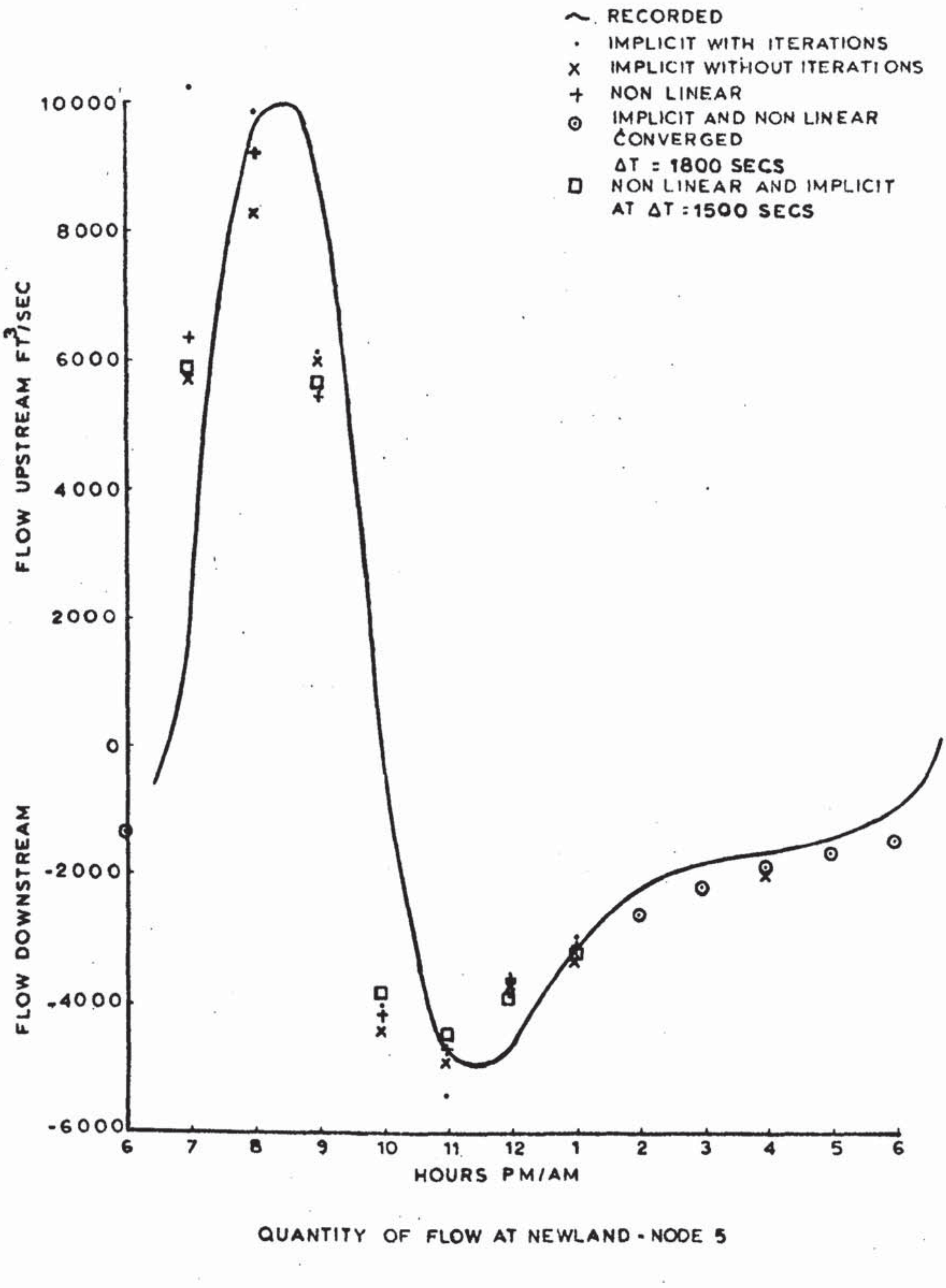
The calculated results for the time step of one hour are shown on Figures 8.16, 8.17 and 8.18, these as expected, show even greater divergence between individual methods and those recorded. The Nonlinear Method again gives consistently the closest results to those recorded. This, together



DEPTH OF FLOW AT NEWLAND NODE 5

RIVER AIRE

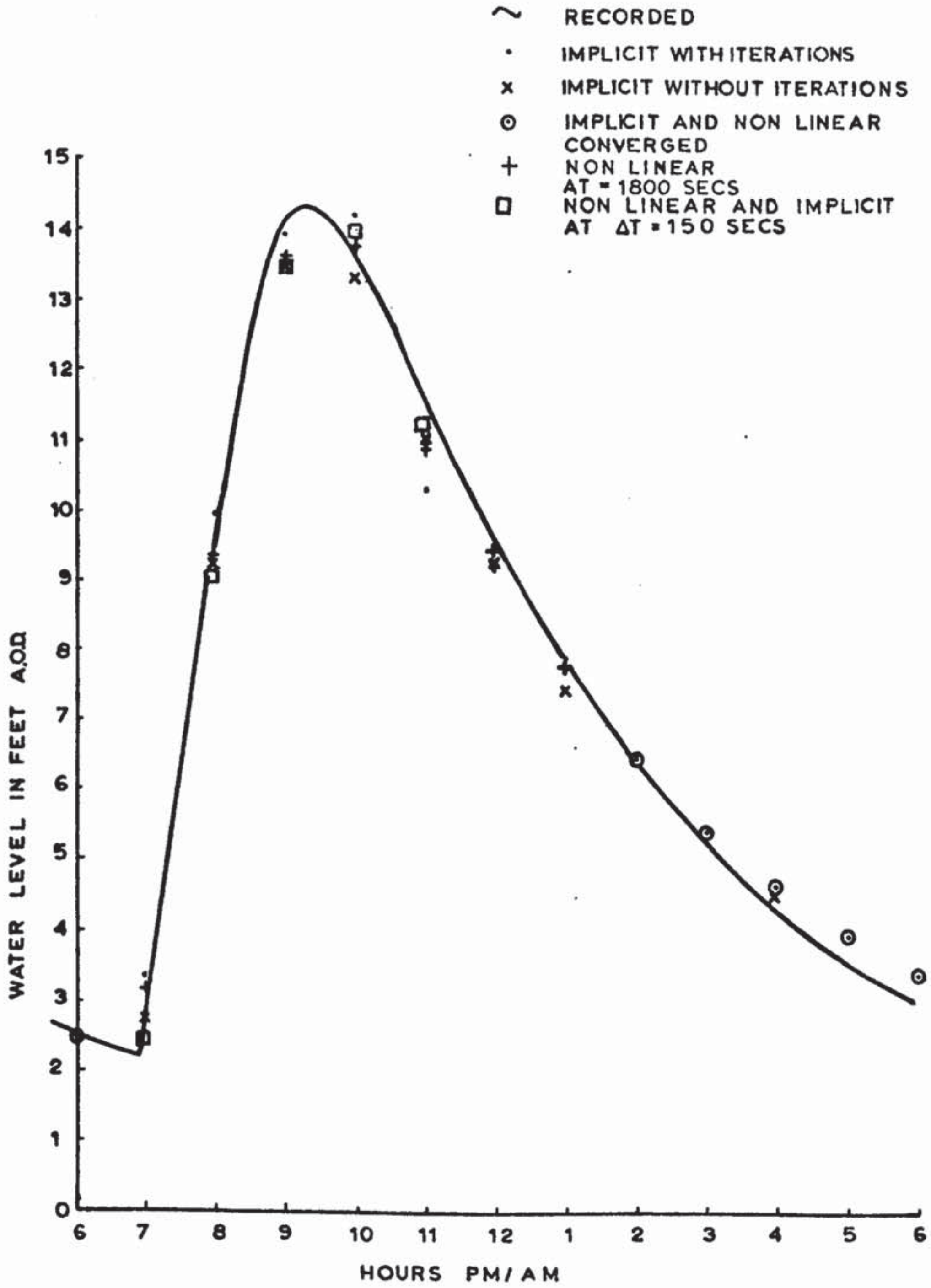
FIGURE 8.13



QUANTITY OF FLOW AT NEWLAND - NODE 5

RIVER AIRE

FIGURE 8.14



DEPTH OF FLOW AT CARLTON NODE 11

RIVER AIRE

FIGURE 8.15

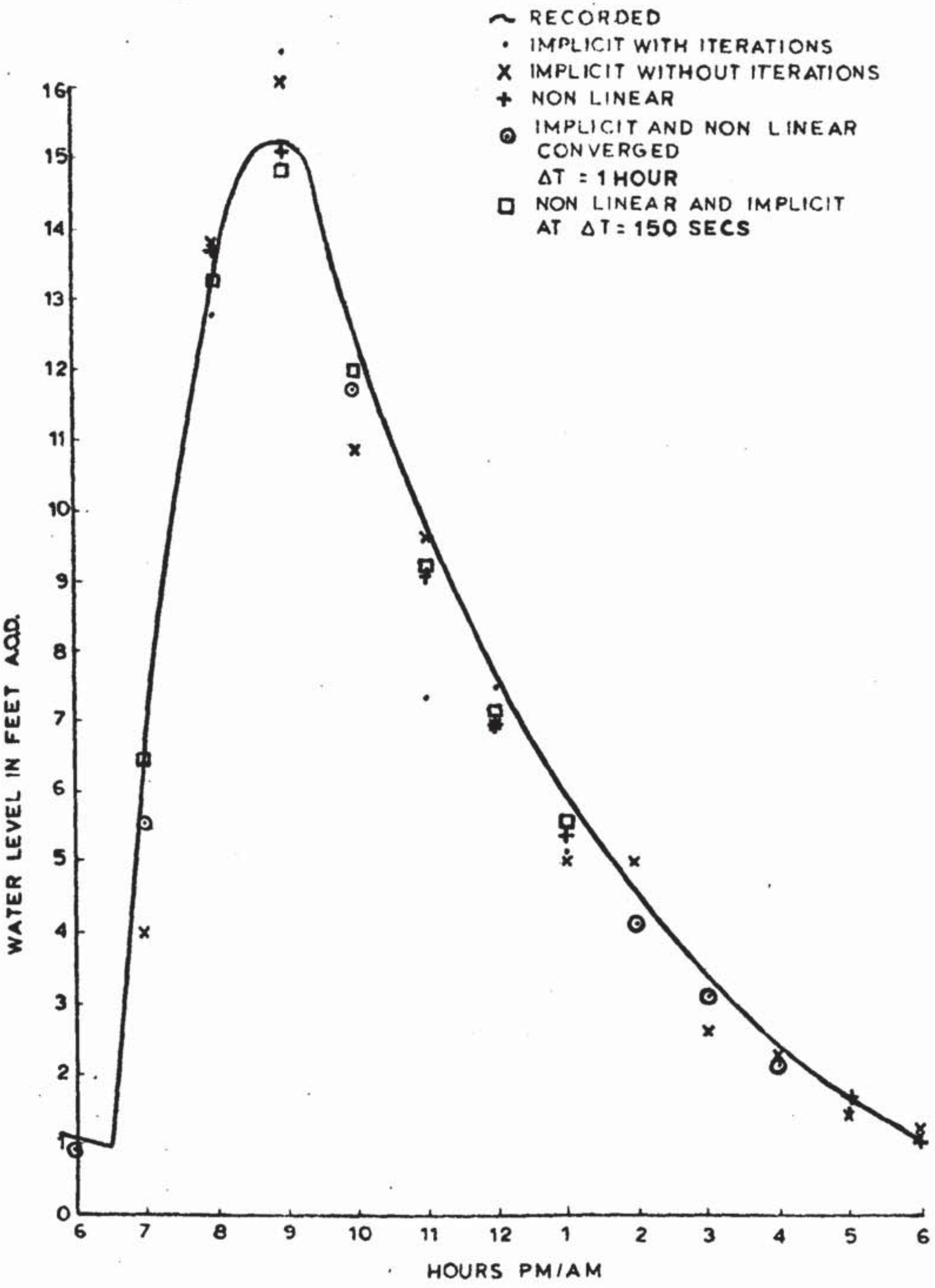
with what has been said previously, clearly shows the improvement obtained by representing the nonlinear friction and $\delta A/\delta x$ terms in the manner described in the section dealing with the formation of the nonlinear finite difference systems. Further, although there is greater divergence between the results obtained from the Implicit systems with those recorded, than the results from the Nonlinear systems, there is little to choose between the results obtained from the two variations of the Implicit Method as far as accuracy is concerned. Hence it is becoming increasingly apparent that there is little to be achieved in allowing the co-efficients of the Implicit systems to vary on an iterative basis.

TABLE 7 shows the computer running times of each method for the tidal cycle, together with the average number of iterations taken in reaching solutions for the time steps shown. For time steps less than or equal to ½ hour then the running times are for 12½ hours of tide, but for time steps greater than this then the running times are for 12 hours of tide only. Also shown in the table are the approximate time steps at which the systems either became unstable or did not converge to a solution. The degree of tolerance placed on the iterative procedures was as in the Hypothetical model, 10^{-3} and consequently 10^{-6} for the Nonlinear methods.

The performance of each of the numerical systems are now discussed in turn, both with reference to the previously presented results and with TABLE 7:

Explicit

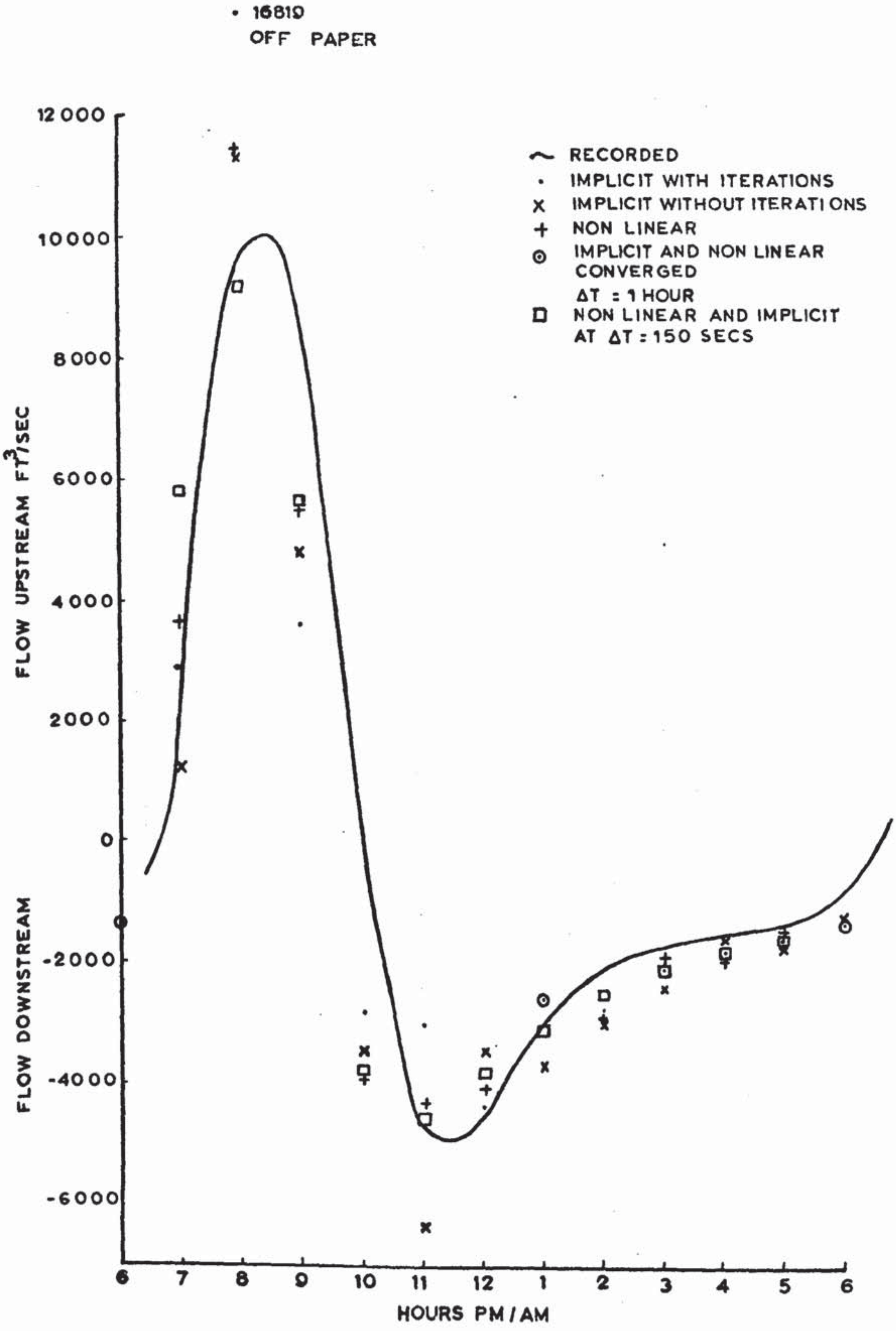
The First and Second Explicit systems were as stated previously, considered to be unstable in that oscillations were apparent in the deeper depths, at time steps of 75 and 150 seconds respectively. The computer running times for the two systems are 174 and 98 seconds and as can be seen from the table, this latter value is the smallest time when compared with the other systems for the time step of 150 seconds. It is not until a time step of ½ hour is used with the Double Sweep Implicit system is this running time bettered.



DEPTH OF FLOW AT NEWLAND - NODE 5

RIVER AIRE

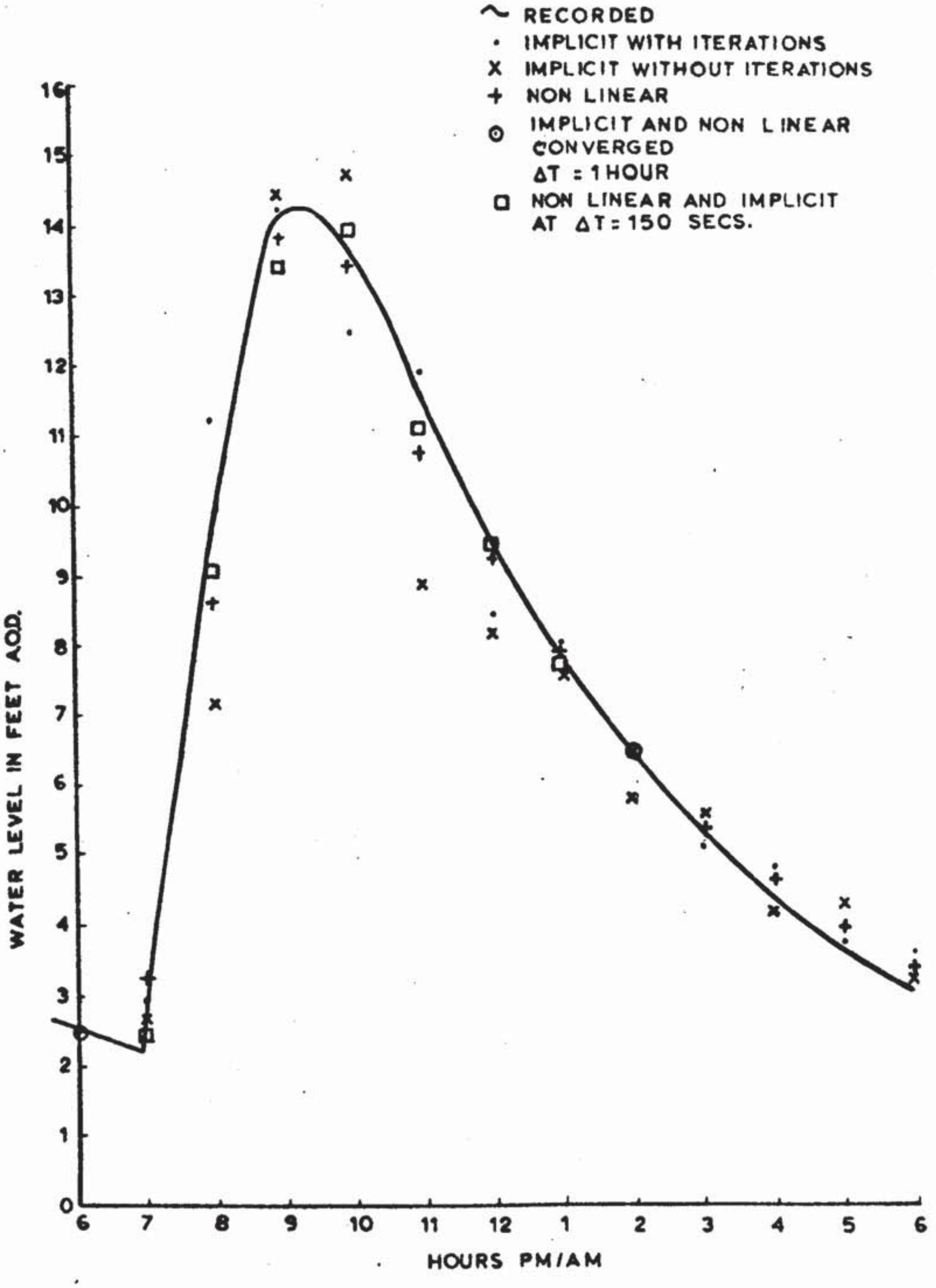
FIGURE 8.16



QUANTITY OF FLOW AT NEWLAND - NODE 5

RIVER AIRE

FIGURE 8.17



DEPTH OF FLOW AT CARLTON - NODE 11

RIVER AIRE

FIGURE 8.18

TABLE 7

NUMERICAL SYSTEM		TIME STEP (SECONDS)	COMPUTER RUNNING TIME (MILL SECONDS)	AVERAGE NUMBER OF ITERATIONS/ TIME STEP	APPROXIMATE FAILING TIME OF SYSTEM
EXPLICIT	FIRST	75	174	1	75
	SECOND	150	98	1	150
IMPLICIT (with varying co-efficients)	GAUSS-SEIDEL	150	442	9	600
	DOUBLE-SWEEP	150	330	4	NOT REACHED
		900	119	6	
		1800	91	7	
		3600	78	9	
		7200	67	11	
		21600	77	12	
		43200	74	13	
	SPARSE-SWEEP	150	1317	4	NOT REACHED
		900	360	6	
		1800	233	7	
		3600	166	9	
		7200	128	11	
		21600	94	12	
		43200	85	13	
IMPLICIT (with constant co-efficients)	SPARSE-SWEEP	150	387	1	NOT REACHED
		900	131	1	
		1800	106	1	
		3600	94	1	
		7200	84	1	
		21600	74	1	
		43200	73	1	

TABLE 7 continued

NONLINEAR	DOUBLE-SWEEP	150	422	3	NOT REACHED
		900	138	4	
		1800	110	5	
		3600	90	7	
		7200	96	14	
		21600	86	10	
		43200	78	6	
	SPARSE-SWEEP	150	1184	3	NOT REACHED
		900	323	4	
		1800	228	5	
		3600	174	7	
		7200	184	15	
		21600	113	10	
		43200	111	6	

Implicit, Gauss-Seidel

This system failed to converge to a solution at a time step of 600 seconds. At a time step of 150 seconds it can be seen that 442 seconds of computation time was required for the full tide with an average number of iterations of 9. These, when compared with the other methods, are by no means good.

Implicit, Double Sweep

This system surprisingly converged to solutions up to and including 12 hours with very few iterations. At a time step of 900 seconds where it is considered that results obtained from the Implicit systems are still sufficiently accurate then the running time is comparable with that of the Second Explicit system. On examination of TABLE 7 it can be seen that this system when compared with the other Implicit and Nonlinear methods is the most economical for this model. However, it must be remembered that a network is not modelled here and that this system is ideally suited to work on a single river in which the elements do not diverge from the basic tridiagonal structure. If this system had been operated with constant co-efficients then smaller running times than those shown could be expected with (as previously shown) little or no loss of accuracy. Such times would be more comparable with those of the Second Explicit system, but as a system was sought that would be capable of handling networks there was therefore, little point in pursuing this further.

It is believed that the slightly larger running times for the time steps of 6 hours and 12 hours were due to changes in the computer system.

Implicit, Sparse Sweep

Results were collected for two variations of this method; one in which the co-efficients of the linearised matrix varied and the other where they were held constant. The system where the co-efficients varied can, on examination of TABLE 7, be seen to have the largest running times up to the time step of 1 hour or over. This system behaves in exactly the same way, apart from running time, that is, as the above Double Sweep method for this model.

The second variation of this method produced savings in computer time over the above variation, but are not as economical as those of the Double Sweep procedure operating on an iterative basis. It will be noticed that this version also provides results for time steps up to and including 12 hours.

Nonlinear, Double-Sweep

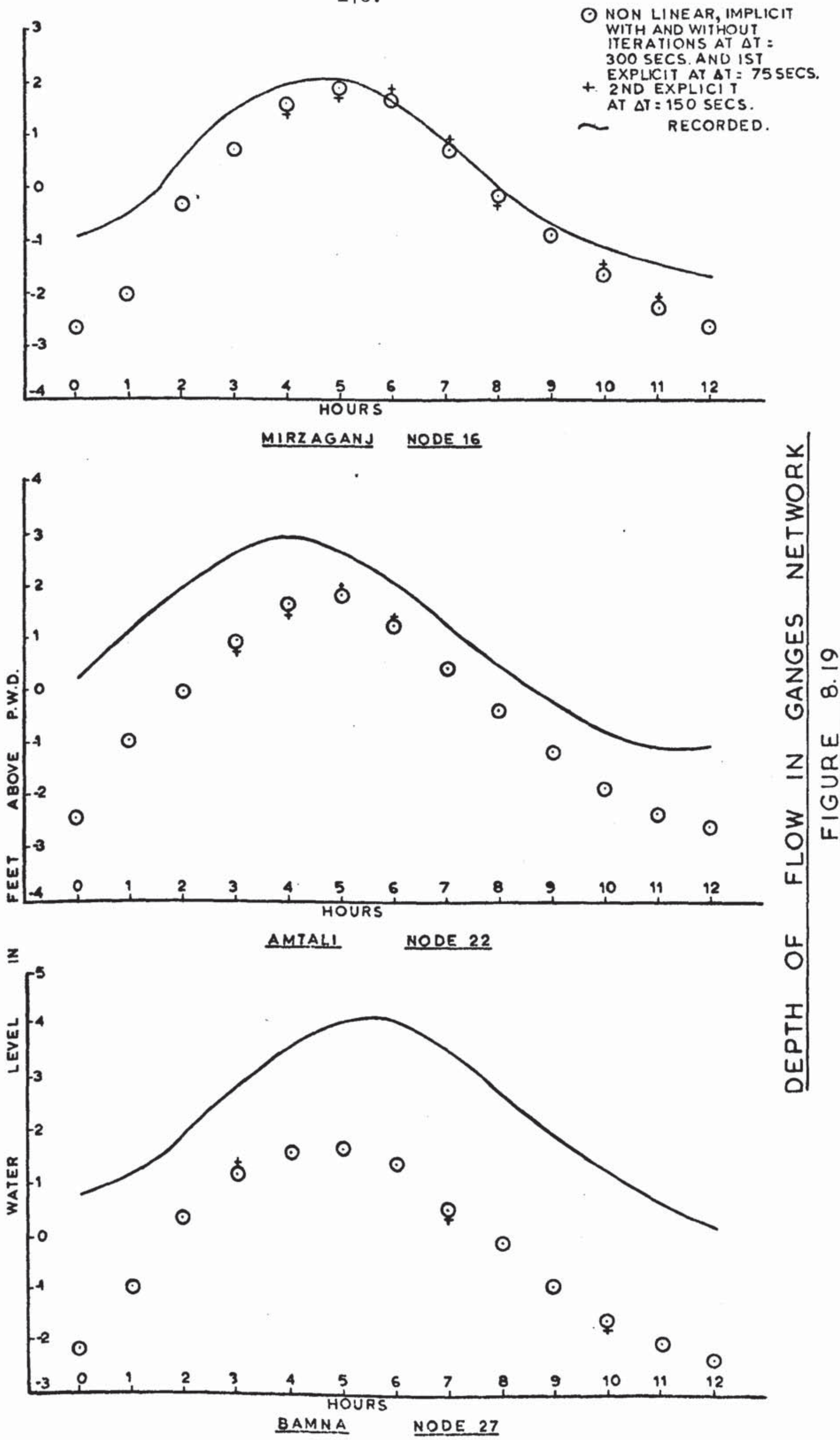
This system, which uses the Double Sweep method to invert its Jacobian, consequently has smaller running times that the following version of the Nonlinear method which uses the Sparse Sweep technique. Both this variation and the one below produced results with time steps up to and including 12 hours. At a time step of ½ hour where the results of the Nonlinear systems are still considered to be sufficiently accurate then this system has a running time comparable to that of the Second Explicit system at 150 seconds. The running times can be seen to "drop off" considerably as the time steps increase and at a value of 1 hour the method has a running time less than that of the Second Explicit system. It is noticeable that the method requires fewer iterations for the two largest time steps than at 2 hours. It is thought that this may be due to the fact that the boundary values for these larger time steps are nearer to the initial values than at 2 hours.

Nonlinear, Sparse-Sweep

This variation of the Nonlinear method which uses the Sparse Sweep technique, identical to that used in the Implicit scheme, to invert its Jacobian, has the largest but one, running time for time steps up to 2 hours. After this the system requires larger times, however, discrepancies appear to be evident both here and in the above version which could be attributed to changes in the computer system.

8.3 Ganges Delta

Figure 8.19 shows, for the three positions where recorded information was available i.e., Mirzaganj, Amtali and Bamna, the calculated depths of flow. On examination of this figure it can be seen that fairly good agreement is obtained between the calculated results and the recorded curve for Mirzaganj.

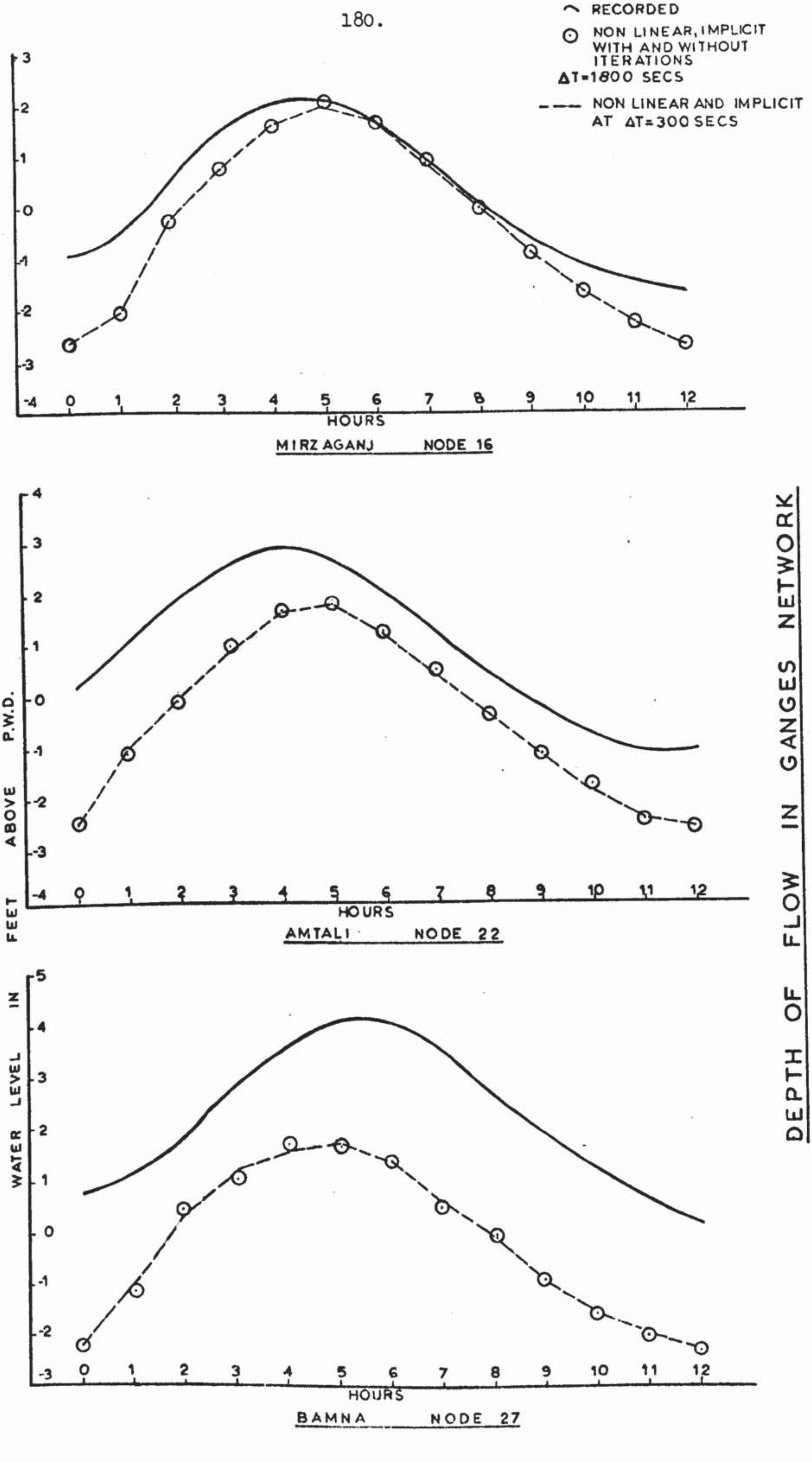


DEPTH OF FLOW IN GANGES NETWORK

FIGURE 8.19

However, the agreement is not so good for the other two station, although the calculated points do follow the general trend of the recorded curves. It is noticed, particularly on the Bamna figure, that differences are evident in the initial conditions and had these been determined more accurately then better agreement would almost certainly have been obtained. It was felt however, that little could be achieved by doing this. Further, although the differences appear large when drawn to the shown scales, it must be remembered that the actual depths of flow at zero hours were, for the three stations Mirzanganj, Amtali and Bamna, 24.56 ft., 32.55 ft., and 26.31 ft respectively. The average deviations between the recorded and calculated points then represent errors of 2.49%, 4.16% and 9.24% respectively.

Differences between the initial conditions and those recorded may be due to a number of things. Firstly, it does not necessarily follow that the recycling process described in Chapter 7 should produce the correct initial conditions for the tide tested. The flow conditions leading up to the beginning of the tide could be completely different altogether. Secondly, the region programmed is subjected to a large amount of sediment transport making the accurate determination of the bed levels very difficult. This, together with the fact that the widths of the rivers and the lengths between nodes were determined by scaling off maps and also that the Chezy 'C' factor was set from experience only to $80\text{ft}^{\frac{1}{2}}$/sec, could account not only for some of the differences in the initial conditions, but also in the subsequent computing of the tide in question. On the experience gained with the initial running of the River Aire model, in which both the friction and bed levels were altered to obtain good agreement between the calculated and recorded results, it was noticed that these altherations gave "fine" adjustment only and that larger effects were obtained by the correct representation of the boundary data. Thus on this basis it could be that the boundary curves themselves were in error. Bench mark errors in the setting of the zeros of the recording gauges would account, quite simply, for the differences present.



MIRZAGANJ NODE 16

AMTALI NODE 22

BAMNA NODE 27

DEPTH OF FLOW IN GANGES NETWORK

FIGURE 8.20

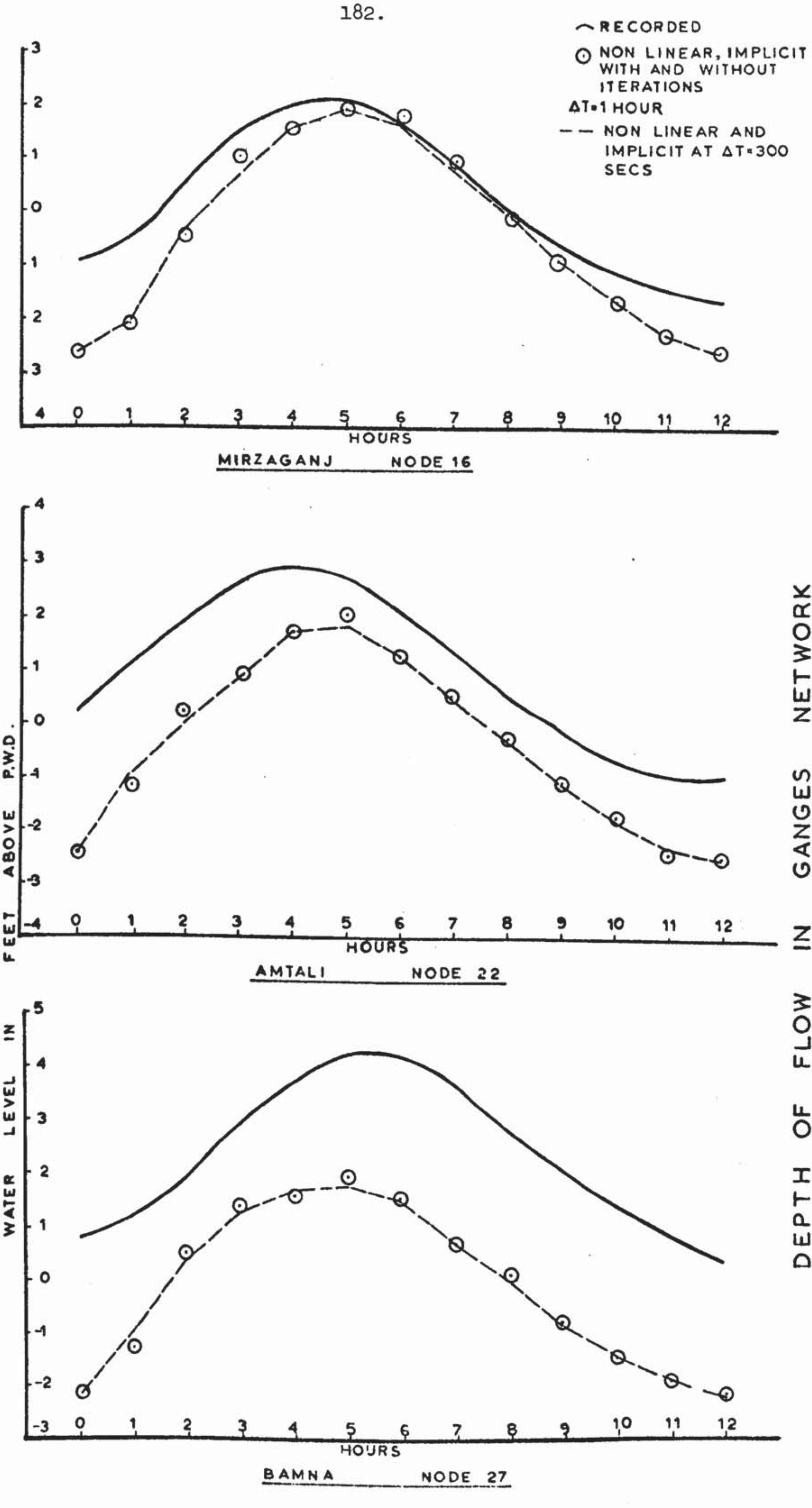
On examination of Fig. 8.19 it can be seen that the calculated values obtained from the Nonlinear and Implicit systems are plotted for a time step of 300 seconds. Results from the Explicit systems are however, plotted at time steps less than this, owing to instability occurring at greater values. It is noticed that there is good agreement between the results of the Nonlinear and Implicit systems with those of the Explicit systems, with only very small variations occurring, mainly at the peaks of the curves.

On Fig. 8.20 where the depths are plotted for the time step ½ hour it can be seen that identical results are obtained from the Nonlinear and Implicit systems and that these values show a similar agreement to those recorded as the previous Fig. 8.19. Also on Fig. 8.20 and on the following Figures 8.21 and 8.22 are plotted the results from the Nonlinear and Implicit systems for the time step of 300 seconds, and these are shown as dotted lines. On examination of the curves of Fig. 8.20 it can be seen that a small degree of oscillation, owing to the use of the relatively larger time step, is becoming evident in the calculated results.

Figure 8.21 shows the calculated results for the time step of 1 hour. Again, there is no difference between the Nonlinear and Implicit systems, at least they are too small to be shown on graphs of this scale and as such are insignificant. Also, it is noticed that the oscillations have become more pronounced but are by no means large.

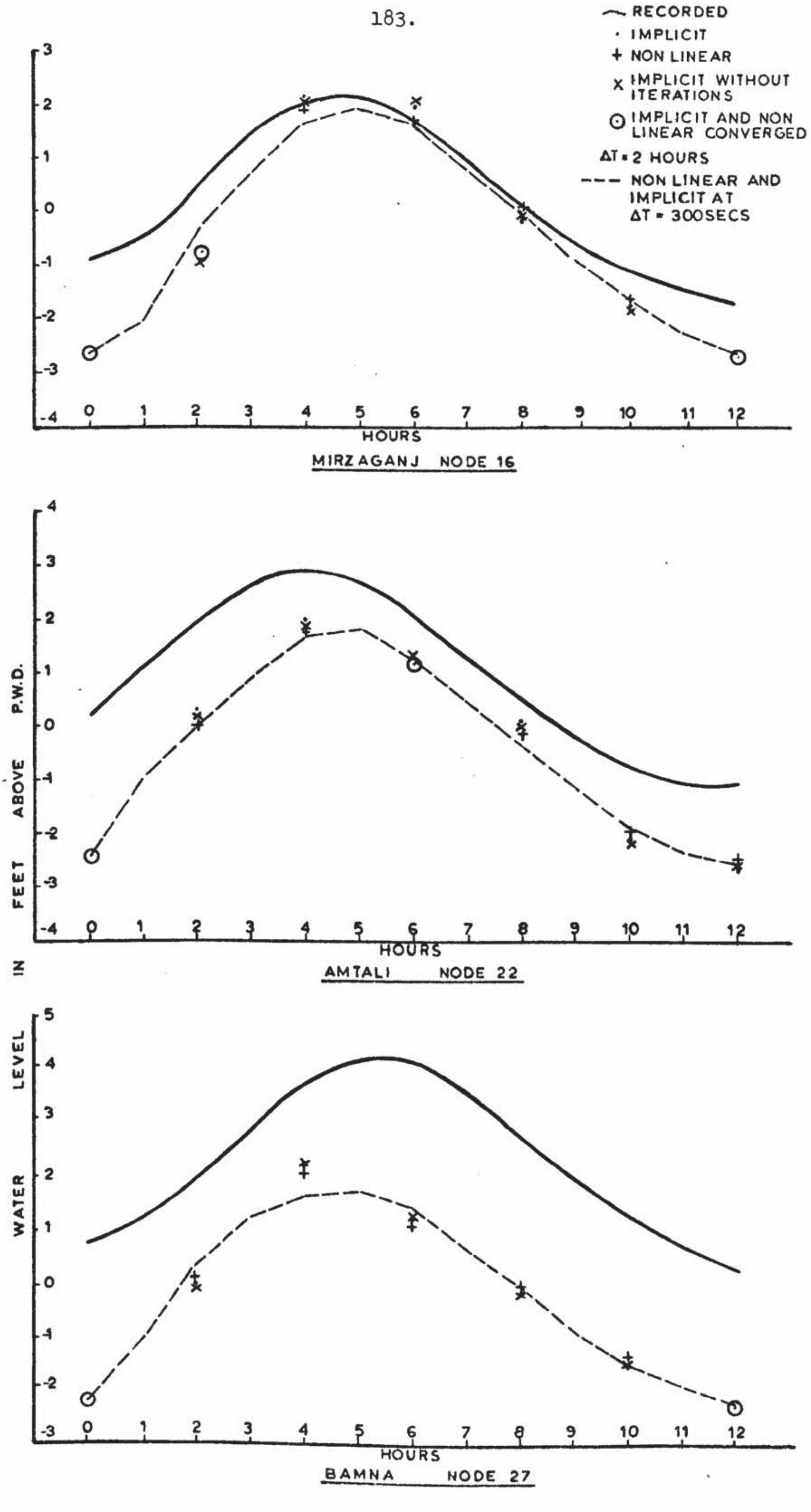
Results for a time step of 2 hours are shown on Fig. 8.22. It is clearly seen that the Nonlinear and Implicit systems no longer give the same values and that the results on the whole show greater divergence from the dotted curves. On examination of these curves it is again, apparent that the Nonlinear program gives results that compare most favourably with those at the time step of 300 seconds and that there appears to be little to choose between the two variations of the Implicit systems, as far as accuracy is concerned, at this time step.

TABLE 8 shows for the Ganges network, similar information as tables 6 and



DEPTH OF FLOW IN GANGES NETWORK

FIGURE 8.21



MIRZAGANJ NODE 16

AMTALI NODE 22

BAMNA NODE 27

DEPTH OF FLOW IN GANGES NETWORK

FIGURE 8.22

TABLE 8

NUMERICAL SYSTEMS		TIME STEP (SECONDS)	COMPUTER RUNNING TIME (MILL SECONDS)	AVERAGE NUMBER OF ITERATIONS/ TIME STEP	APPROXIMATE FAILING TIME OF SYSTEM
EXPLICIT	FIRST	150	205	1	150
	SECOND	300	117	1	300
IMPLICIT (with varying co-efficients)	GAUSS-SEIDEL	300	550	9	1000
	DOUBLE-SWEEP	300	599	8	8000
		1800	294	20	
		3600	257	34	
		7200	301	86	
	SPARSE-SWEEP	300	1709	4	NOT REACHED
		1800	448	6	
		3600	281	6	
		7200	186	7	
		21600	106	7	
		43200	114	8	
IMPLICIT (with constant co-efficients)	SPARSE-SWEEP	300	516	1	NOT REACHED
		1800	171	1	
		3600	139	1	
		7200	116	1	
		21600	100	1	
		43200	97	1	
NONLINEAR	DOUBLE-SWEEP	300	668	3	21600
		1800	307	4	
		3600	270	5	
		7200	402	6	
		300	1535	3	
		1800	404	4	

TABLE 8 continued

	SPARSE-SWEEP	3600	307	5	NOT REACHED
		7200	224	6	
		21600	155	8	
		43200	130	8	

7. The computer running times here though, are for 12 hours of tide. Again, the degree of tolerance placed on the iterative procedures was 10^{-3} for the Implicit systems and 10^{-6} for the norm in the Nonlinear systems.

As before, the performance of each of the numerical systems will be discussed, in turn, both with reference to the previously mentioned figures and with TABLE 8.

Explicit

Again, it was observed that good results were obtained from the Explicit systems when time steps were used that were below their numerical points of instability. These were approximately, 150 seconds for the first Explicit system and 300 seconds for the second. On examination of TABLE 8 it can again, be seen that the second Explicit system, at a time step of 300 seconds gives the most economical running time of 117 seconds. This was not bettered until a time step of 2 hours was computed with the Implicit, Sparse-Sweep system with constant co-efficients. Results at this large time step are not only widely spaced but are also inaccurate.

Implicit, Gauss-Seidel

It can be seen that this system failed at a time step of 1000 seconds and was due to the fact that it did not converge to a solution. At this time step the program completed 5 hours of computation with a maximum number of iterations of 81 and then failed to converge on the next iteration. For the time step of 300 seconds however, a running time of 550 seconds is shown which is good when compared with the other Implicit and Nonlinear systems, but is still by no means good when compared with the running time of the second Explicit system. The average number of iterations for the 300 second time step was 9 which does not indicate a strongly convergent system.

Implicit, Double-Sweep

The Double-Sweep Implicit system failed to converge to solutions at time steps of approximately 8000 seconds. This was due to the inversion process of the linearised finite difference system and is indicated by the large number of iterations at the time step of 2 hours. On examination of Figures 8.21 and 8.22 it can be seen however, that the use of time steps greater than 1 hour produces too much divergence of the calculated results from those with a time step of 300 seconds and that, within the limits of acceptable accuracy the system works quite adequately even though a large number of iterations are required.

At time steps of 1 hour or less then this system has the second most economical running times of the Implicit and Nonlinear systems. However, as the time step increases then, owing to the large number of iterations required then large amounts of computer time are necessary.

Implicit, Sparse-Sweep

As with the other models, then two variations of this method were also tested. One in which the co-efficients of the matrix $\bar{A}$ in eq. (4.4.10) were held constant and the other in which they were allowed to vary on an iterative basis. The performance of both of these variations can be seen on examination of TABLE 8. It is noticed that both converged to solutions up to and including 12 hours and so verifying that the reason why the two former Implicit methods (i.e., Gauss-Seidel and Double Sweep) failed because of their particular methods of inversion of the finite difference system and not because of the finite difference system itself.

The variation of this system with varying co-efficients can be seen to have very large running times initially but they then tailed off to become comparable to the other systems later. However, the results of the calculated depths show that little or no improvement in accuracy is obtained whatsoever by allowing the co-efficients to vary and that large savings in computer running time can be achieved by keeping them constant. In fact, this latter variation of the Implicit system using the sparse matrix inversion technique described in section 4.6, gave the most economical running times of all the systems (except the explicit) for no effective loss of accuracy.

Nonlinear, Double-Sweep

On examination of TABLE 8 it can be seen that this variation of the Nonlinear method failed to converge to solutions at time steps of 6 and 12 hours and is attributed to the technique of using the Double Sweep procedure to invert the Jacobian to obtain the vector $\bar{p}_i$, in eq. (4.8.1). This is illustrated by the fact that an average of 79 iterations were required to form this vector for the time step of 2 hours with a corresponding running time of 402 mill seconds. For the time step of 1 hour an average of only 24 iterations was required, which is by no means good, but had the correspondingly smaller computational time of 270 seconds. This system can be seen to produce savings over the following variation for time steps of 1 hour or less, but as stated previously the normal Implicit systems produce identical results to the Nonlinear systems and are more economical, at these time steps.

Nonlinear, Sparse-Sweep

This variation of the Nonlinear technique provided solutions for all the time steps tried but at no stage did it give economical computational times. However, the Nonlinear method does give more accurate results for large time steps and it is possible that if a model had been tested which had smooth boundary curves over a large period of time, such as in flood routing, then this system could provide economical running times for large time steps.

With the Ganges model, then at time steps of 6 hours and 12 hours the tolerance on the norm had to be increased from 10^{-6} to 10^{-5} as round off errors, owing to the computer's precision, were magnified by the dynamic function and so prevented convergence. This was attributed to the large time steps used and the large dimensions of the Ganges rivers.

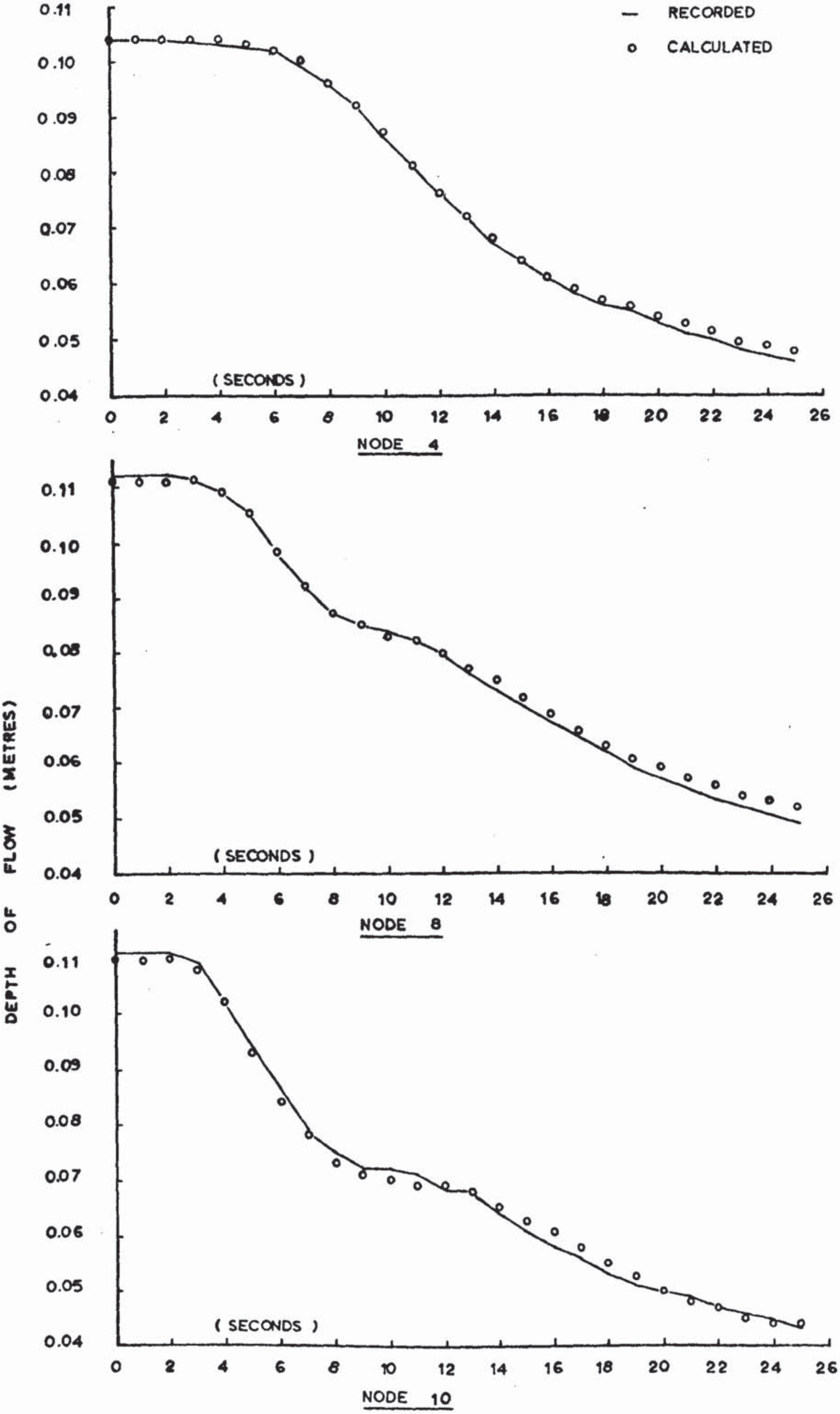
In both this, and the above variation, then solutions were obtained with the parameter $t_i = 1$ throughout, i.e., the full Neutonian step (see eq.(4.8.1)) except when a time step of 12 hours was used on the latter variation. Another noticeable characteristic of the high convergence of these Nonlinear methods are the very small number of iterations required at the smaller time steps.

8.4 Laboratory Model

In the foregoing tests it has been shown that the differences between the computed results from the different Finite Difference systems was negligible provided a relatively small time was used. With this in mind it was considered only necessary to run one of the Finite Difference systems on the laboratory tests and the one chosen was the Gauss-Seidel implicit system. The time step used for all three tests was 0.5 seconds which was slightly greater than the Leap Frog time step for the maximum depth and minimum section length.

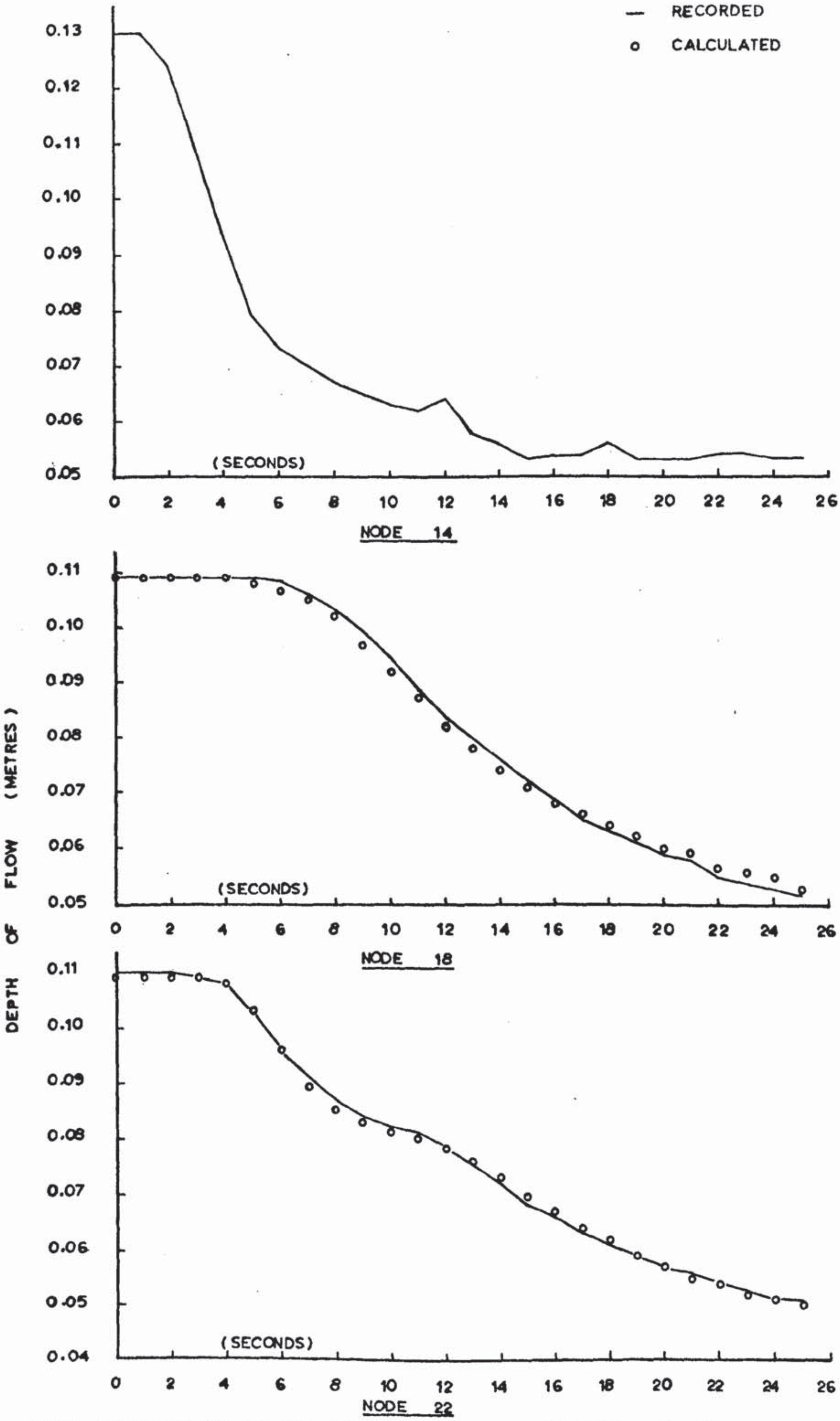
To obtain the initial conditions it was first thought that by feeding into the models the correct boundary data and arbitrary initial conditions the systems would converge to the correct initial conditions. However, this proved to be a difficult and lengthy process. A much more efficient method was adopted where the author wrote a program based on the interval halfing technique. The basic procedure was to compute backwater profiles around the network working from the downstream head controlled boundary. To do this an assumed value of the flow had to be made through the section spanning laterally between the two main lengths. Successive estimates of the estimated and computed flow at node 24, see Fig. 6.9, were compared and further estimates were made using the interval halfing technique until sufficient accuracy was obtained. The remaining two depths at nodes 2 and 16 were then calculated. To utilise this procedure it was important that the basic dynamic equation, eq. (2.5.4) was used, minus the time varying terms.

When first the calculated value of the linear friction factor, k_s of 0.055 cms, whose obtaining was discussed in section 6.6, was used on the first unsteady flow test, the calculated depths produced were much lower than those recorded. As no measures had been taken to account for the effect of turbulence and loss of momentum at the junctions it was decided to simulate this effect by increasing the friction parameters local to the junctions. The basic philosophy was to increase these values until good comparison was obtained with



1 ST. UNSTEADY FLOW TEST

FIGURE 8.23



I ST. UNSTEADY FLOW TEST FIGURE 8.24

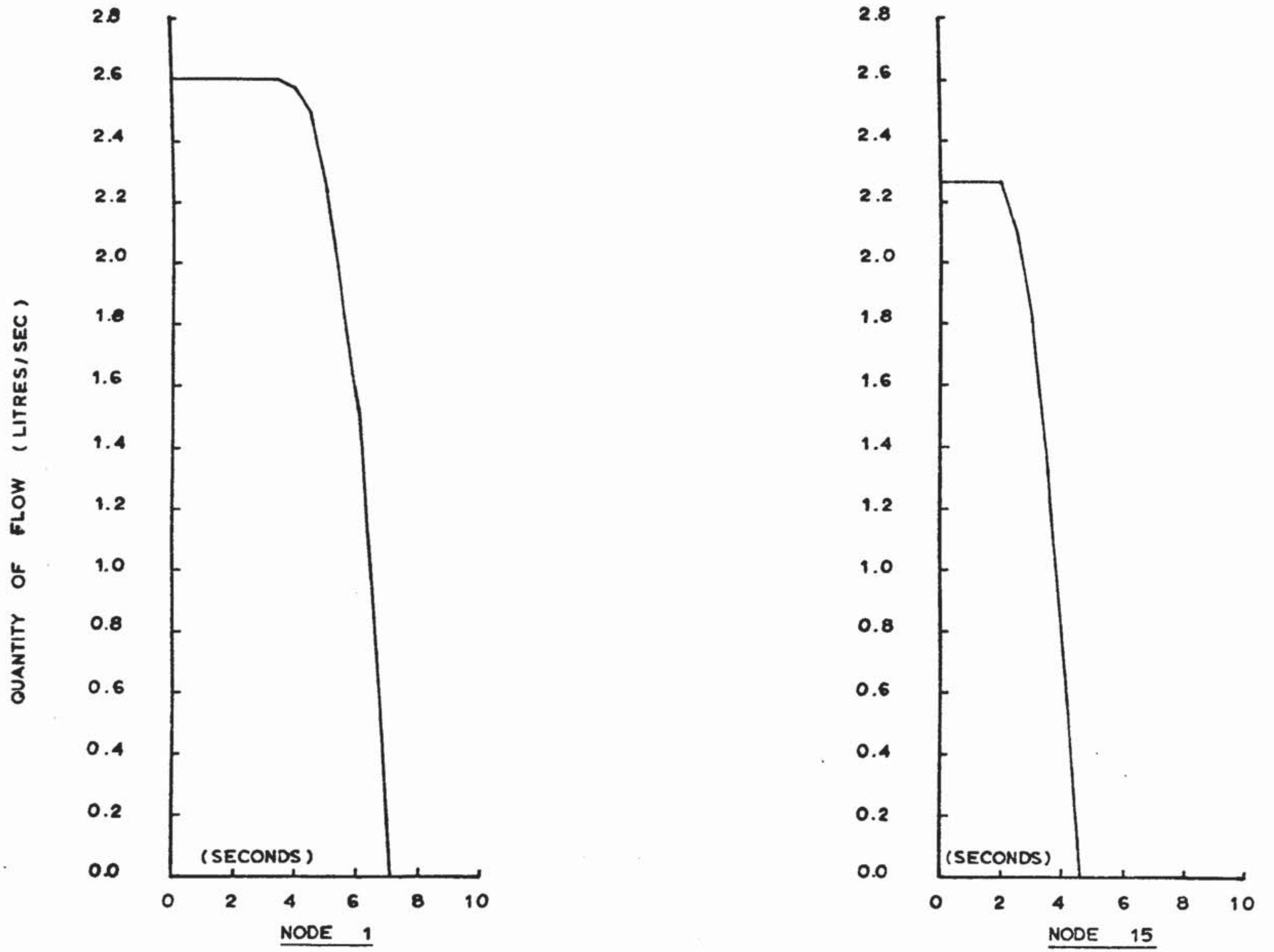
the first unsteady flow test. On obtaining these they would be held constant and used for the second and third unsteady flow tests. After several runs the final values of k_s were chosen to be :

1 cms	node 9
6 cms	node 11
0.5 cms	node 13
1.3 cms	node 23
0.055 cms	node rest

The computed and recorded results for the first unsteady flow test are shown on Figures 8.23 and 8.24. In this test the boundary flows were held constant at values of 0.611801×10^{-3} M^3/sec at node 1 and 1.25836×10^{-3} M^3/sec. for node 15. Fig. 8.24 also shows the head varying boundary, node 14, obtained when the downstream weir was lowered. On examination of these figures it can be seen that there is very good agreement between the calculated and recorded results. The initial conditions obtained using the previously mentioned backwater program and the above friction values, together with the relative heights of the head nodes, are shown in TABLE 9 for the full three tests.

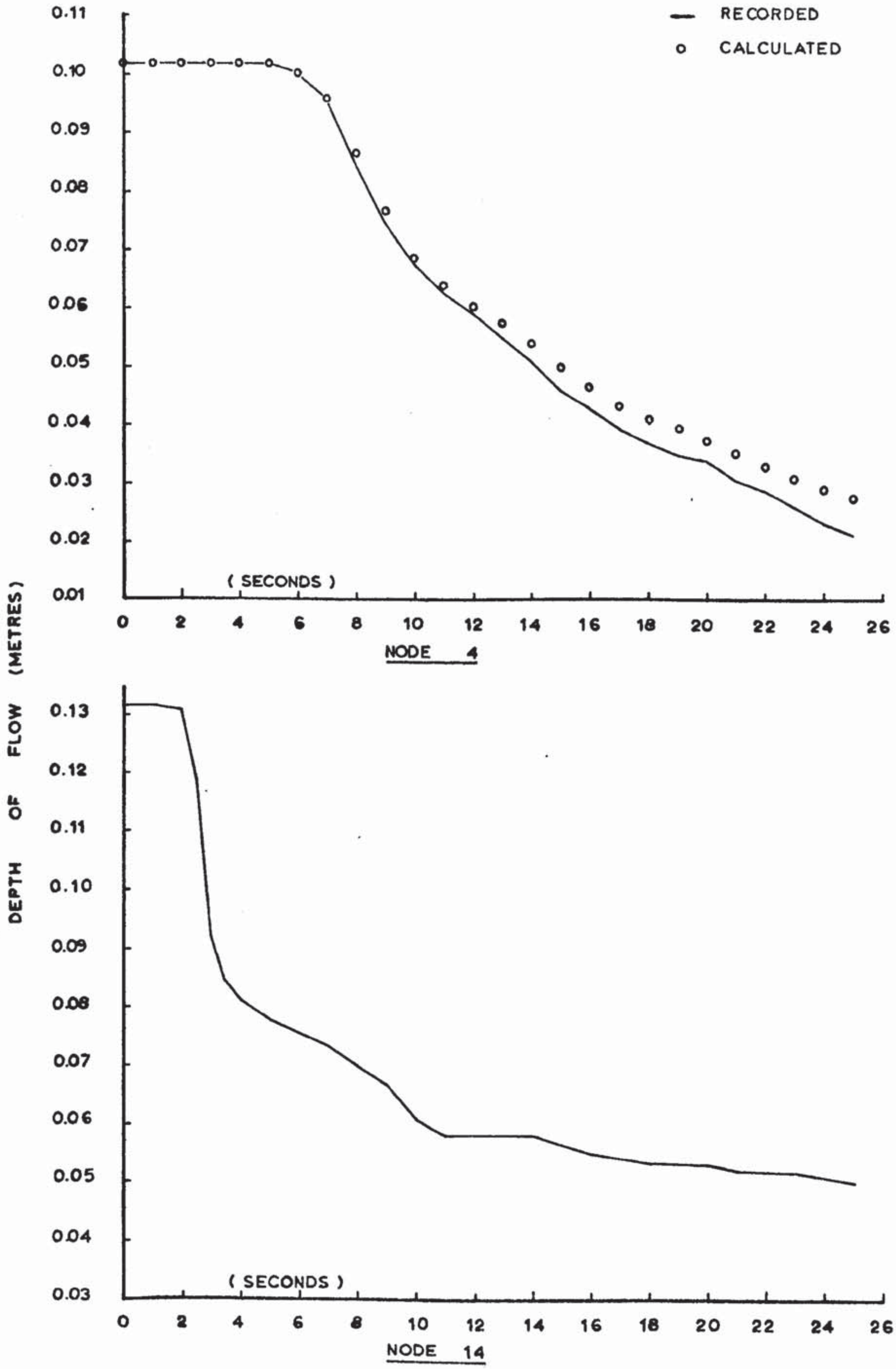
The second unsteady flow test, as previously described, had varying flow boundaries as well as a varying downstream head controlled boundary. The time varying flow curves for the nodes 1 and 15 are shown on Fig. 8.25, and the depth boundary curve is shown on Fig. 8.26. Results obtained from the running of this test are shown on Figures 8.26, 8.27 and 8.28. On examination of these curves it can be seen that good agreement was obtained over approximately half of their lengths, but then the calculated and recorded values began to diverge. In all the cases the computed results were higher than those recorded. This suggested that the increased friction values arrived at because of the turbulence were too large. The effect of such large values would be less for the deeper depths than the smaller ones.

Results from the running of the third unsteady flow test are shown on Figures 8.30 and 8.31. Again it can be seen that the adjusted friction values used were too high, particularly with the initial and end conditions, although

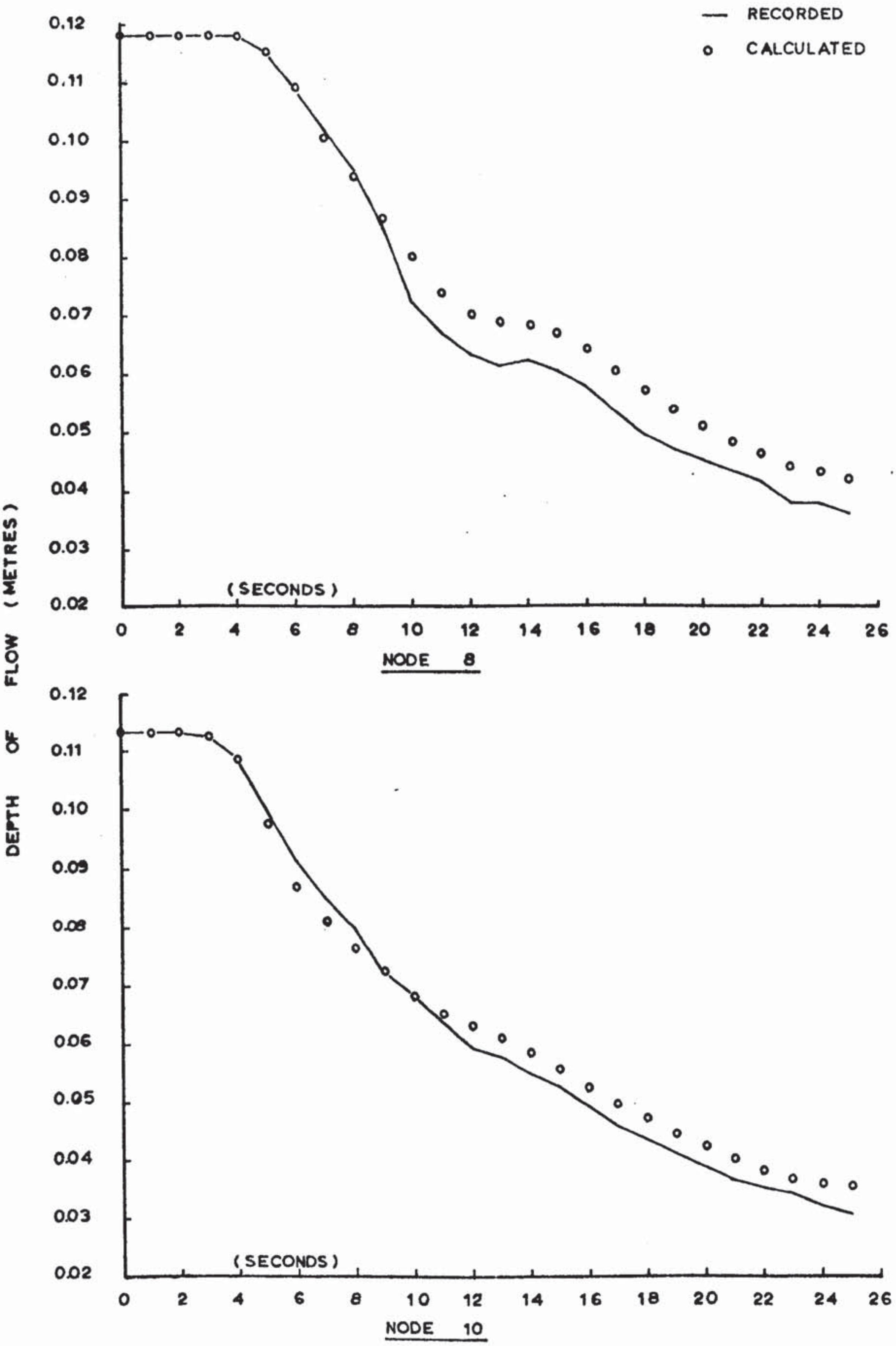


2 ND. UNSTEADY FLOW TEST

FIGURE 8. 25



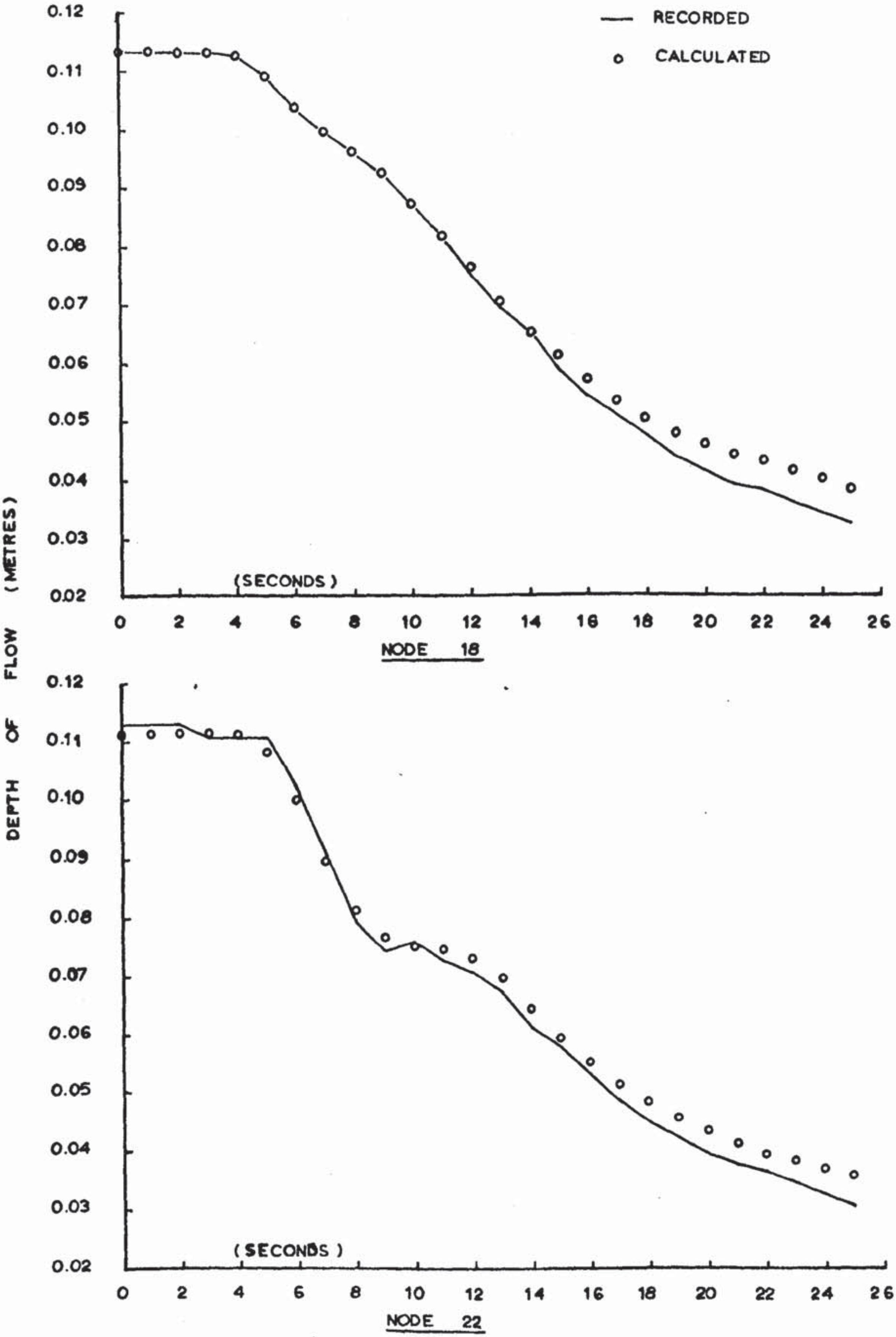
2 ND. UNSTEADY FLOW TEST FIGURE 8.26



NODE 8

NODE 10

2 ND. UNSTEADY FLOW TEST FIGURE 8. 27



2 ND. UNSTEADY FLOW TEST FIGURE 8.28

TABLE 9

NODE		UNSTEADY FLOW TESTS								
		1			2			3		
NUMBER	TYPE	DEPTH (M)	FLOW CUMECS	BED DATUM (M)	DEPTH (M)	FLOW	BED DATUM (M)	DEPTH (M)	FLOW	BED DATUM (M)
1	QB		0.612			2.603			0.604	
2	H	0.100		0.0315	0.094		0.0414	0.037		0.0414
3	Q		0.612			2.603			0.604	
4	H	0.104		0.0275	0.102		0.0328	0.045		0.0328
5	Q		0.957			2.645			1.020	
6	H	0.107		0.0235	0.111		0.0236	0.054		0.0236
7	Q		0.957			2.645			1.020	
8	H	0.111		0.0195	0.118		0.0164	0.061		0.0164
9	Q		0.957			2.645			1.020	
10	H	0.110		0.0205	0.113		0.0196	0.057		0.0196
11	Q		1.870			4.865			1.876	
12	H	0.120		0.01025	0.122		0.0098	0.065		0.0098
13	Q		1.870			4.865			1.876	
14	HB	0.130		0.0	0.132		0.0	0.075		0.0
15	Q		1.258			2.262			1.273	
16	H	0.113		0.0178	0.119		0.0173	0.062		0.0173
17	Q		1.258			2.262			1.273	
18	H	0.109		0.0225	0.114		0.0216	0.057		0.0216
19	Q		0.913			2.220			0.856	
20	H	0.108		0.0234	0.111		0.0234	0.054		0.0234
21	Q		0.913			2.220			0.856	
22	H	0.109		0.0219	0.112		0.0227	0.055		0.0227
23	Q		-0.913			-2.220			-0.856	
24	Q		0.345			0.042			0.417	
25	H	0.107		0.0245	0.108		0.0271	0.051		0.0271
26	Q		-0.345			-0.042			-0.417	

on the whole the agreement was good. The boundary curves for this test are shown on Figures 8.29 and also on 8.31.

The above three tests show that the agreement between the computed and recorded results was reasonably good. The degree of accuracy was however a function of the damping factors which were increased to take into account the turbulence at the junctions. Because of the size of the model these junctions had large effects on the rest of the system. In retrospect it is thought that a largernetwork should have been used with better designed junctions, so that their effect on the system as a whole would only be local.

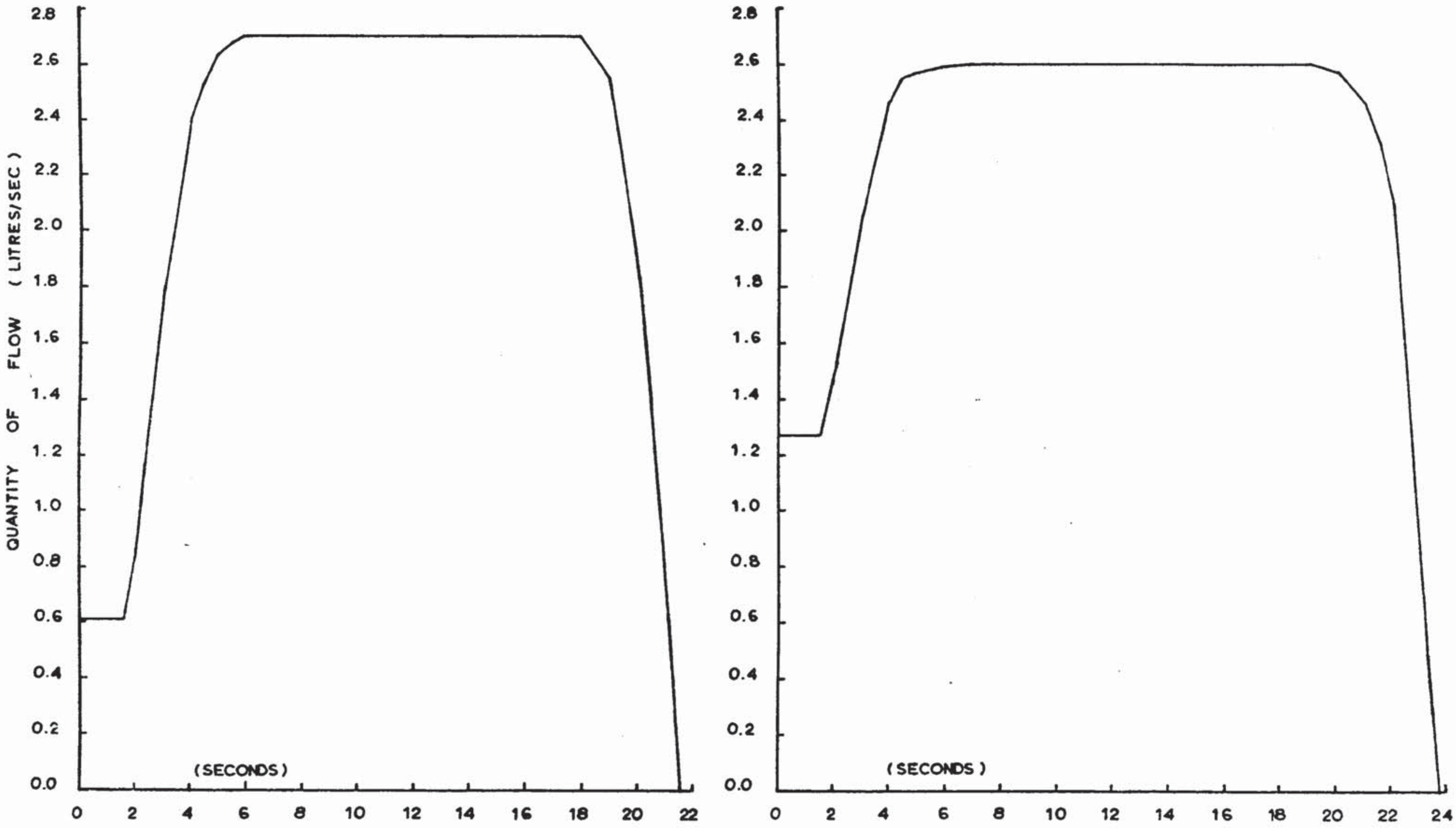
8.5 Numerical Stability and Convergence

In Chapter 5 it was stated that convergence of the Finite Difference equations was dependant upon the stability of their linearised forms and also upon their consistency, providing the equations possessed a certain degree of smoothness. The necessary theoretical criteria for the stability of the systems were in that Chapter, developed and it was also shown that all of the finite difference systems tested were consistent with their original partial differential equations. With this latter point shown then the theoretical aspect of both stability and convergence depends only upon stability. The first object of this section is then to compare the numerical points of instability of the Finite Difference systems in relation to the criteria developed in Chapter 5. As regards the consistency requirement, then it has been amply shown that smooth and reasonably accurate results were obtained from the Systems when small time steps were used, but as the time steps increased then divergence, in the form of oscillations, became evident.

The numerical stability of the four Finite Difference systems tested will now be considered under their two basic headings, i.e., explicit and implicit systems.

Explicit

In all the models tested then the Explicit methods became unstable at relatively small time steps. The theoretical criteria for stability of the

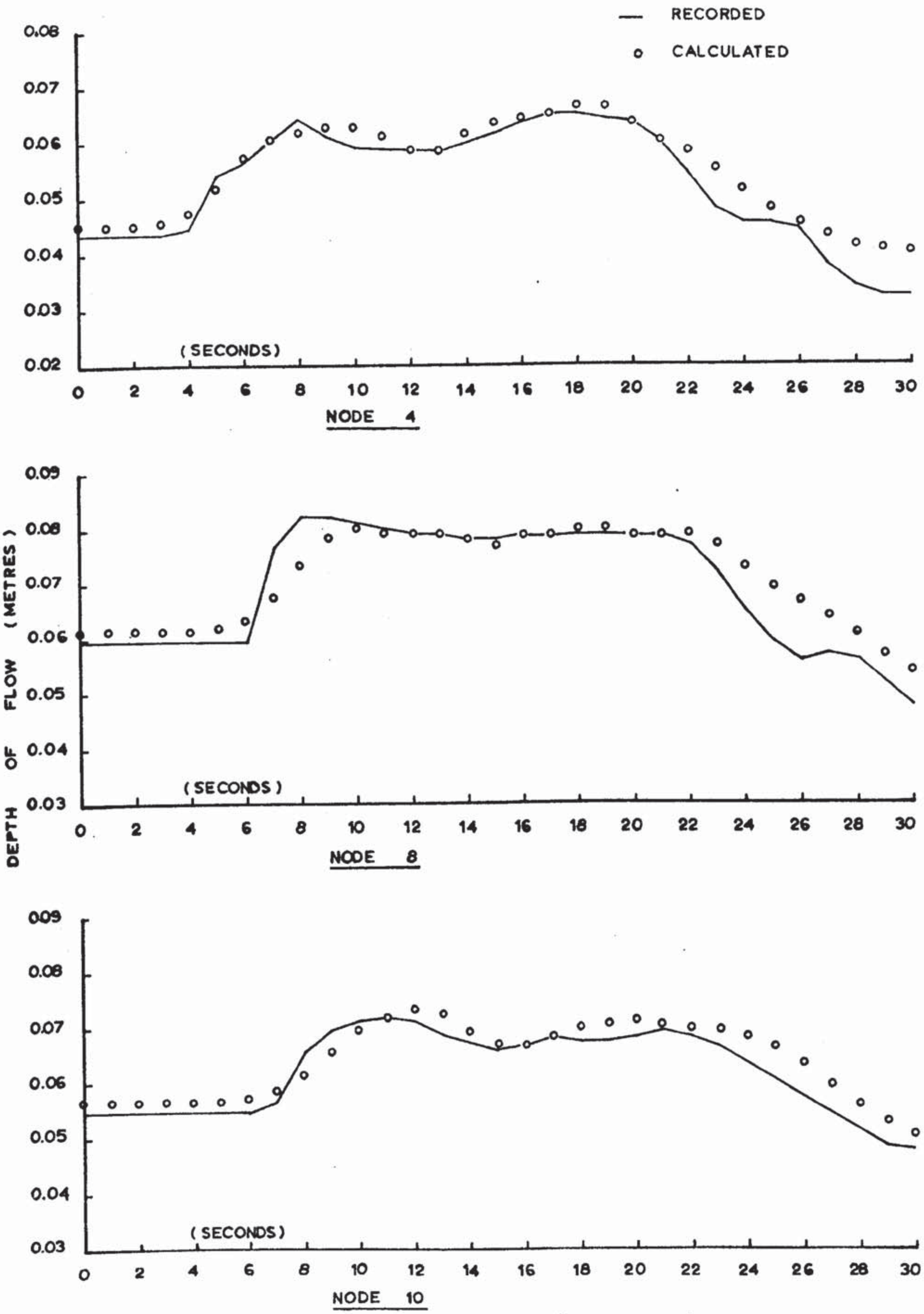


NODE 1

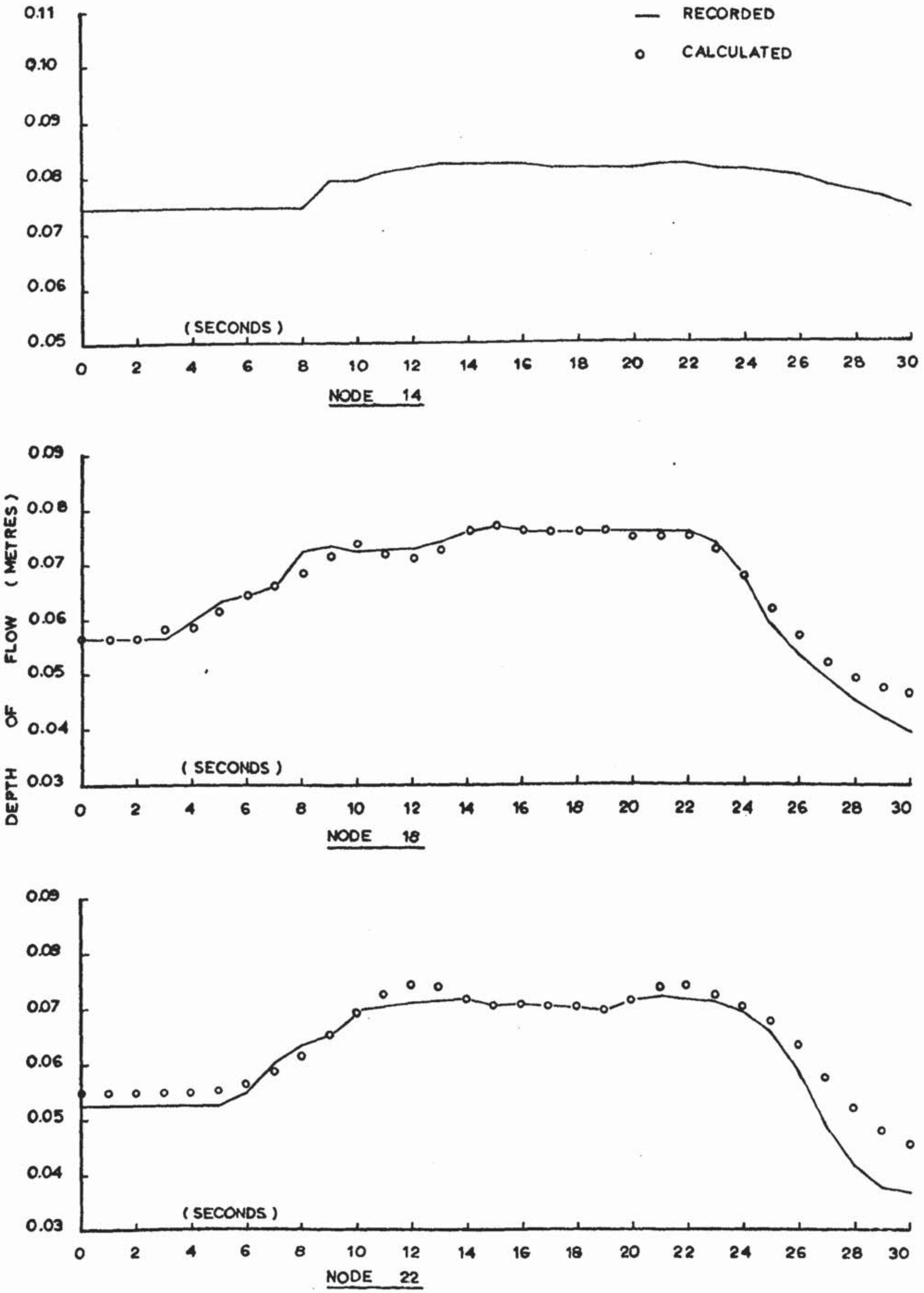
NODE 15

3 RD. UNSTEADY FLOW TEST

FIGURE 8.29



3 RD. UNSTEADY FLOW TEST   FIGURE 8.30



3 RD. UNSTEADY FLOW TEST FIGURE 8.31

two systems are given by equations (5.3.17) and (5.3.27). On examination of these equations it can be seen that they are virtually velocity dependant, which therefore means that if the numerical points of instability of the two Explicit systems comply with these equations, then the Finite Difference systems themselves are unsound. For, even at slackwater, the theoretical time step will be approximately zero.

The values calculated by the above mentioned equations are also related to the stability criteria of the more stable Leap Frog methods, given by eq. (3.2.12). These latter values also served the purpose of providing the basic time step upon which results were collected from the various models.

As stated previously, the numerical points of instability of the first and second Explicit methods for the Hypothetical model were, 55 and 111 seconds respectively. On application of equations (5.3.17) and (5.3.27) to the initial conditions of this model, the theoretical values were calculated to be 58.9 and 117.8 seconds, which gives very close agreement. The Leap Frog time step was calculated to be 31.9 seconds, at slackwater. This then set the initial value of the time step for this model to 25 seconds.

TABLE 10 shows the calculated theoretical points of instability for the first Explicit system for six positions along the River Aire model. These values were calculated for the three sets of conditions: initial, points of maximum discharge, and final, using values obtained from an implicit program. The value of the $\delta V/\delta x$ term was found to be of the order of 3 to 4% of the friction term and was therefore neglected in the calculations. It can be seen that the average theoretical value of the time step starts at 94.03 seconds and becomes 47.96 seconds at the maximum flow conditions, this value then returns to 122.31 seconds at the tide. The reduction at the deep depths is due to the effect of the increase in the hydraulic radius being much greater than the increase in velocity. This explains why instability is seen to be evident at the high points in the curves of the results. The disturbances would then be damped out as the depths dropped. Because of the large variation in depth

TABLE 10

NODE	LENGTH (FEET)	FLOW (FT^3/SEC)			DEPTH (FEET)			THEORETICAL TIME STEP (SECONDS)			LEAP FROG TIME STEP (SECONDS)		
		I	M	F	I	M	F	I	M	F	I	M	F
2	3960	-1515	11503	-1511	8.40	22.92	8.32	75.53	28.60	99.22	265.95	174.53	267.03
4	4280	-1432	9950	-1430	8.55	21.60	8.41	90.06	40.12	120.86	284.01	192.27	285.91
6	3960	-1332	8158	-1333	8.55	20.07	8.58	84.93	41.82	108.48	263.47	184.10	263.12
9	4700	-1171	5684	-1175	8.45	17.16	8.92	131.49	79.12	144.71	315.65	234.30	308.40
13	3440	-988	4333	-994	8.60	18.44	9.27	67.99	31.16	70.52	230.72	167.72	223.52
17	4970	-814	2583	-822	8.90	15.81	9.31	114.17	66.94	130.04	328.49	258.32	322.31
						AVERAGE		94.03	47.96	112.31			

the actual value at which instability occurred was difficult to determine however, as stated previously, the first Explicit system was considered to be unstable at a time step of 75 seconds. This value can be seen to compare quite favourably with the figures shown in TABLE 10. As the additional terms in the stability equations (5.3.17) and (5.3.27) have been neglected, then the theoretical points of instability of the second Explicit system are twice those shown for the first. Also, as the numerical point of instability was considered to be 150 seconds for this second system then again, this figure compares favourably with the theoretical values.

Also shown on TABLE 10 are the Leap Frog stability values for the three sets of conditions mentioned previously. These were calculated assuming zero velocity. It can be seen that these are much greater than those of the first Explicit system and that the minimum value was 167.7 seconds for the smallest length of 3440 ft. This then set 150 seconds as the basis of the time steps for the River Aire model.

The first and second Explicit systems of the Ganges model are shown to fail at time steps of 150 and 300 seconds respectively. At these values instability did not totally occur but oscillations of the order of 6 to 7 ft. were evident between the results from these programs when compared with those from the Implicit systems. To determine the actual theoretical values of the time steps at which instability occurred would be difficult as the flow conditions varied so much. However, to give some idea of their order, then four points were chosen as representative of the area, using the initial conditions. Although the flows varied greatly over the tidal period the depths, as shown on the previous figures, do not vary very much. The four positions, together with their section lengths, depths and flows are shown in TABLE 11. Also shown are the calculated points of instability of the first and second Explicit systems and the Leap Frog stability values at slackwater.

On examination of TABLE 11 it can be seen that the average theoretical values of the time steps were 63.7 and 132.8 seconds for the first and second

TABLE 11

NODE	LENGTH (FEET)	FLOW (FT^3/SEC)	DEPTH (FEET)	$\frac{g\|V\|}{C^2R} \times 10^{-2}$	$\frac{\delta V}{\delta x} \times 10^{-2}$	$-\frac{1}{h}(\frac{\delta H}{\delta t}+\frac{V\delta h}{\delta x}) \times 10^{-2}$	EXPLICIT THEORETICAL TIME STEPS (SECONDS)		LEAP FROG TIME STEP (SECONDS)
							1st	2nd	
34	9500	-94061	21.44	0.0452	-0.002	0.0009	56.5	119.4	361.6
27	20000	-85672	26.31	0.0188	-0.001	-0.0015	84.0	170.4	687.0
38	12600	72360	20.75	0.0341	-0.001	-0.0004	78.6	161.1	487.4
46	22100	-22795	25.09	0.0074	-0.0015	-0.0015	35.7	80.4	777.6

system respectively. These are then much smaller than the previously stated numerical values. However, it must be remembered that the flows are velocity dependant and that for instance the flow at node 46 becomes 89987 ft^3/sec. after only 1 hour with a corresponding increase in depth of only 1.5 ft. This represents an increase in the time step, for that point, by a factor of 4. It would seem reasonable that these periods of high flow could lend stability to periods of low flow with a subsequent continuing of the computation. Another consideration is that had the computations been continued for periods greater than 12 hours then instability may quite easily have occurred at time steps smaller than those estimated.

It can be seen from TABLE 11 that the minimum value of the Leap Frog stability time step is given as 361.6 seconds for node 34 which had the smallest section length. This then set the basic time step for the Ganges model to 300 seconds.

In order to appreciate the magnitude of size of the theoretical values of the time steps given by the relevant stability criteria, for the laboratory data, then a number of tests were conducted with data from the first unsteady flow test only. Both the first and second Explicit systems ran with this data which had initial conditions of 0.1 M and zero flow, and constant boundary conditions set at their initial values. Calculation of theoretical values was again complicated by the varying flow conditions produced, however, values obtained were of the order of 0.03 seconds for the first Explicit system and approximately twice this for the second Explicit system. Both of these values compared well with those obtained numerically from the programs. In comparison the Leap Frog stability value for this test, for maximum depth and minimum length, was approximately 0.37 seconds at zero flow.

Implicit

On examination of TABLES 7 and 8, it can be seen that, both for the River Aire and Ganges Delta models, one form or another of both the Implicit and Nonlinear systems converged to solutions for time steps up to and including

12 hours. This, incidently, represented maximum time steps of 288 and 144 times the explicit Leap Frog steps for the River Aire and Ganges models respectively. Because of the foregoing, where certain Implicit or Nonlinear programs did fail, then their failure can be attributed to non-convergence of the particular numerical procedures in solving the nonlinear simultaneous equations and not the Finite Difference systems themselves. Furthermore it was stated in the previous section dealing with the stability of the explicit systems that for the River Aire model the $\delta V/\delta x$ term was approximately 3 to 4% of the friction term in the stability eq. (5.3.17), also for the Ganges model, TABLE 11 shows the values of the friction term to greatly exceed the values of the other terms in the stability equations. The implication is that, as the terms in eq. (5.3.34) may be reduced to those mentioned above, then the inequality regarding the theoretical stability of the Implicit systems, i.e., eq. (5.3.34) was held, at least for the small time steps from which the results were collected. Further, this inequality was considered to be held for the complete tides in question.

A slightly different picture is evident with the Hypothetical model. Here, only the Nonlinear programs converged to results for atime step of 2500 seconds, i.e., 100 times the Leap Frog value. All the Implicit systems failed at time steps less than this. The Double Sweep and Sparse Sweep programs which had varying co-efficients failed, as stated previously, at a time step of 375 seconds, and these programs with constant co-efficients failed at a time step of 225 seconds. In order to investigate why these systems failed then two things were done; the first was to program a nonlinear system with the same finite difference equations as the Implicit systems, i.e., using linearised friction and $\delta A/\delta x$ terms, and allowing the co-efficients to vary. The second was to do exactly as the first but this time keep the co-efficients constant. In other words solve the two variations of the Implicit Finite Difference systems upon which the Double Sweep and Sparse Sweep methods worked by the Nonlinear method.

The results of this exercise were, that the system with the varying co-efficients converged for time steps up to and including 2500 seconds, and thus showed that the reason why the Double Sweep and Sparse Sweep methods with varying co-efficients failed, at the time steps shown, was due to the particular solution processes and not the Finite Difference system. However, the variation where the co-efficients were held constant failed again, by the system becoming unstable, at the smaller time step. This indicates that it was the Finite Difference system that caused failure by instability for this time step and had nothing to do with the inversion technique. This is further illustrated by the fact that the Implicit system using the two inversion methods in question, with varying co-efficients, failed by not converging to a solution, whereas the Implicit system using the same inversion techniques, but with constant co-efficients, failed by the solutions gradually becoming unstable.

The results of the Double Sweep and Sparse Sweep Implicit methods with constant co-efficients, at $\Delta t = 225$ seconds, were compared with the results of the Implicit method with varying co-efficients, with $\Delta t = 2500$ seconds. This showed that the former produced the larger oscillations initially. The results of the latter did not vary greatly from those of the original Nonlinear Finite Difference systems which as stated previously, converged very slowly to the steady state. The reason why instability occurred in the former case may therefore be due to these large oscillations in depth and discharge, and that in the second case, although the oscillations were not large enough to produce instability, they were large enough to produce very slow convergence to the steady state. Possibly, if a larger time step than 2500 seconds had been used on the original Nonlinear system, or the Nonlinear system that had been modified to work on the Implicit system with varying co-effients, then instability may have occurred.

TABLE 12 shows, for nodes 3 and 5, the values of the variables in eq. (5.3.34) both for the Implicit system with constant co-efficients at

TABLE 12

<table>
<tr><td colspan="7">IMPLICIT DOUBLE SWEEP SYSTEM WITH CONSTANT CO-EFFICIENTS,
ΔT = 225 SECONDS</td></tr>
<tr><td rowspan="2">TIME</td><td colspan="3">NODE 3</td><td colspan="3">NODE 5</td></tr>
<tr><td>$\frac{g|V|}{C^2R}$</td><td>$\frac{2}{h}\ \frac{\delta H}{\delta t}$</td><td>$\frac{V}{h}\ \frac{\delta h}{\delta x}$</td><td>$\frac{g|V|}{C^2R}$</td><td>$\frac{2}{h}\ \frac{\delta H}{\delta t}$</td><td>$\frac{V}{h}\ \frac{\delta h}{\delta x}$</td></tr>
<tr><td>0</td><td>0.0576</td><td>0.0025</td><td>0</td><td>0.0576</td><td>0.0005</td><td>0</td></tr>
<tr><td>225</td><td>0.0215</td><td>0.0013</td><td>-0.0012</td><td>0.0255</td><td>0.0027</td><td>-0.0003</td></tr>
<tr><td>450</td><td>0.0410</td><td>-0.0006</td><td>0.0006</td><td>0.0363</td><td>0.0012</td><td>-0.0016</td></tr>
<tr><td>675</td><td>0.0246</td><td>-0.0007</td><td>0.0013</td><td>0.0263</td><td>-0.0011</td><td>0.0003</td></tr>
<tr><td colspan="7">IMPLICIT NONLINEAR SYSTEM WITH VARYING CO-EFFICIENTS,
ΔT = 2500 SECONDS</td></tr>
<tr><td>0</td><td>0.0354</td><td>0.0004</td><td>-0.0001</td><td>0.0367</td><td>0.0004</td><td>-0.0001</td></tr>
<tr><td>2500</td><td>0.0307</td><td>0.0003</td><td>0.0001</td><td>0.0289</td><td>-0.0002</td><td>0</td></tr>
<tr><td>5000</td><td>0.0387</td><td>0.0002</td><td>-0.0002</td><td>0.0431</td><td>0.0001</td><td>-0.0002</td></tr>
<tr><td>7500</td><td>0.0291</td><td>-0.0002</td><td>-0.0001</td><td>0.0263</td><td>-0.0001</td><td>-0.0001</td></tr>
</table>

Δt = 225 seconds and for the same system with varying co-efficients with Δt = 2500 seconds. It will be noticed that the expressions in eq. (5.3.34) have been divided throughout by Δt and have been presented in terms of depth and velocity only. Results were calculated for four time steps only as further calculation was unnecessary owing to the fact that the system with a time step of 225 seconds gradually became unstable and the other with a time step of 2500 seconds converged to the steady state. Also, we are concerned only with the conditions that produce either instability or convergence.

Two things are apparent from the examination of the table, the first is that the inequality in eq. (5.3.34) is fulfilled in all cases, and secondly, it can be seen that there is a larger fluctuation in the nonlinear terms of the upper table than in those below. The first comparison means that the stability condition for the Implicit systems was complied with in both cases. However, it must be remembered that this criteria was developed for linearised systems with gradually varying co-efficients and when this is not true then the subsequent criteria may not yield reliable information.

As stated in the section dealing with the results from the laboratory tests, it was felt that only one of the implicit methods need be used. Consequently no detailed observations were made on the behaviour of the Implicit systems as a whole, from a stability point of view. However, one of the observations that was made was that when the programs were running to obtain the initial conditions (using the method by which arbitrary initial and the correct constant boundary values were used) then convergence to the steady state was very slow indeed. The friction parameter used for these exercises was the one obtained from direct measurement of the water slopes, as previously described, and no increase had been made, at that time, for the junctions. Computation of the elements of the stability criteria, eq. (5.3.34), yielded that the friction term was in fact less than the others and consequently the inequality was not true.

Once the correct initial values had been obtained, the Gauss-Seidel method then computed the unsteady profiles and no problems were experienced. The method took on average 5 to 6 iterations per 0.5 second time step.

Convergence of the Numerical Systems

In all of the foregoing results, each of the Implicit and Nonlinear systems failed at least once by non-convergence of the numerical procedures used to solve the resulting set of nonlinear simultaneous equations. The reasons why these systems failed together with their general performance with respect to convergence are now discussed.

Gauss-Seidel

This system has been shown to become unstable before all the other Implicit and Nonlinear systems. The discussion in Chapter 5 showed that the convergence of this method is dependant upon the spectral radius of the resulting Gauss-Seidel iteration matrix. If this is greater than unity then errors could propagate in the iteration process and so prevent convergence, whilst if it is less than unity then errors will decrease and convergence will take place. In Chapter 5 inequalities were developed, relating the elements of the linearised system, to see if convergence was guaranteed. It was shown that guaranteed convergence was only conditional; depending upon the relative values of the elements and in particular, upon the time steps.

In order to investigate the above further, a comparison was made of how the spectral radius varied with the number of iterations required to reach a solution for a test on the Hypothetical model. A time step was chosen that was relatively high but not too large to prevent convergence occurring at all. Owing to the oscillatory nature of the results produced by the Hypothetical model when a large time step was used then the discharges produced were either very large or very small. When they were large then it would be expected that eq. (5.4.9) is more fulfilled because of the friction term, and this, as expected gave smaller eigenvalues, a correspondingly smaller spectral radius and few iterations. However, when the flows were small then the reverse, as to be expected, occurred. Figure 8.32 shows the spectral radius for the

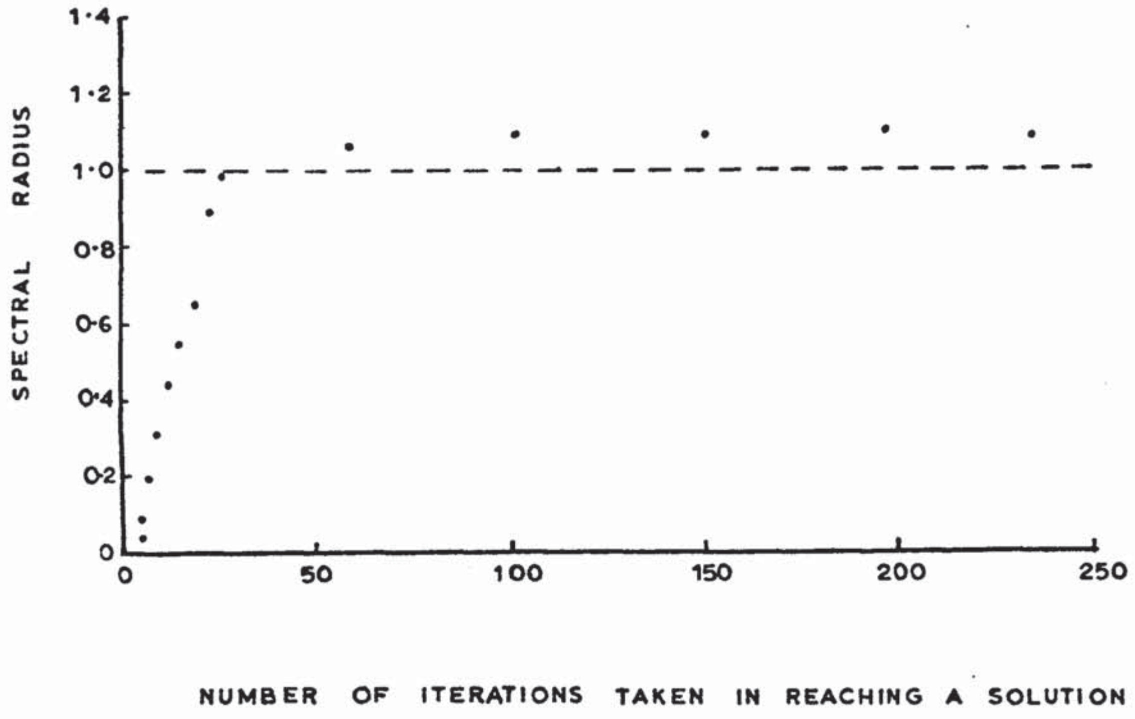

1·4
1·2
1·0
0·8
0·6
0·4
0·2
0
SPECTRAL RADIUS
0
50
100
150
200
250
NUMBER OF ITERATIONS TAKEN IN REACHING A SOLUTION

FIGURE 8.32

Gauss-Seidel matrices constructed in determining the last iteration of a time step, against the number of iterations taken to reach the solution. Although the spectral radius may vary with each iteration for one time step it is expected to converge to one value if the iteration process is itself convergent. The spectral radii themselves were calculated using a subroutine written by the author, using the "Power Series" method, ref. [21].

On examination of Fig. 8.32 it can clearly be seen that as the spectral radius (ρ)increased from 0 to 1 then so did the number of iterations increase slowly. However, as ρ exceeded 1 then the number of iterations became very large indeed, although convergence still occurred. Further, it is interesting to note that ρ had a value of 2.59 for the matrix constructed in determining the first iteration of the second time step, of the computed model. Also, tests showed that when a time step was used that was too large for a particular model, then non-convergence was preceded by extremely large eigenvalues.

Double-Sweep

The convergence of this system must be considered for the two cases, one in which all the elements of the linearised system lie on the tridiagonal band and two, when they do not. The non-convergence of the first case, when the system is working in an iterative manner, can be attributed to the equations being too nonlinear,and that the initial values of the variables were too much in error. Here, the variables are assigned values on the first back substitution which are too far removed from their correct value, and after feeding back into the matrix to update the co-efficients, the variables are then assigned values that are even more removed from their true value, and so on. Another consideration is that round-off errors due to the elimination process may become so large with the larger time steps that again the variables may not be assigned values nearer their correct ones and the above may again, occur. For the systems that work with constant co-efficients then round-off error is the only consideration. Its effect here is simply to produce inaccurate results, depending on its magnitude.

The above discussions also apply to the second case where off diagonal elements exist, but in addition there is the effect of the way in which these were treated. For small time steps in which it is expected that the variables do not vary much over the time considered then this method works reasonably well. However, as the time step increases, as shown by the results of the Ganges model, then the system requires a large number of iterations prior to failure. Further, it would seem likely that the method would be less convergent with complex networks than with simple ones.

Sparse-Sweep

As this method is for the direct solution of sparse matrices then the factors governing its convergence on an iterative basis are the same as those for the first case of the Double Sweep method. These are again, that convergence will depend upon how close the initial values of the variables are to their final value and secondly, round-off error may become large.

When the method does not work iteratively then, as before, round-off error is the criteria.

Nonlinear Methods

Until the Jacobian was estimated in the analytical manner described in Chapter 4 then a large amount of trouble was experienced with the convergence of these methods. However, after adopting the measures described then the method proved to be strongly convergent for a wide range of conditions. It can be seen though that these analytical estimates are still functions of the unknown variables and consequently if estimates of the latter are too much in error then problems could be expected. This is likely to be the cause of the relatively large number of iterations required for the time steps of 2 hours and 4 hours in the River Aire model, which for these would give large variations in depth at the boundary. This is further supported by the fact that the procedure which chose steps less than the full Newtonian step had to be entered several times in order to reduce the norm. Whereas for all the Hypothetical model tests and for all the Ganges model tests, except one using the 12 hour time step, the full Neutonian step was used.

However, non-convergence was experienced with the version of this method which used the Double Sweep procedure to invert the estimated Jacobian on an iterative basis. This, as to be expected, was for the Ganges model and has previously been discussed.

The variation which used the Sparse Sweep method to invert the Jacobian proved to be very convergent for the Ganges network. Although, as stated previously, problems arose owing to the round-off errors which were introduced into the arithmetic by the computer being magnified to affect the norm.

CHAPTER 9

CONCLUSIONS

The basic aim of the research, as stated in the introduction, was to investigate the applicability of a set of finite difference schemes for the solution of the unsteady flow equations, in open channel networks. In this respect, conclusions are now drawn from the results presented in the previous chapter. On the basis of these conclusions recommendations are made, both for the subsequent use of the systems developed and also for further work. Finally, conclusions are made on the general application of the procedures described for the determination of the theoretical stability and convergence criteria.

The Finite Difference systems tested gave reasonably accurate results, on the whole. The results from the laboratory work, although limited in nature, did show that the systems were basically accurate and the river models showed, that depending upon the time step used, there was little divergence between individual schemes. In the River Aire model a small degree of "fitting" of the results was used, which is perfectly acceptable with mathematical modelling This then produced excellent correl ation between calculated and recorded values. Where the recorded information was incomplete, as in the Ganges network then differences between calculated and recorded were obtained. Even though the differences were not large it is expected that better data would produce more accurate results.

Of the numerical systems themselves then it was shown that the Explicit methods gave accurate results within their stability limits. However, as these limits were primarily a function of the velocity of flow then the systems are basically unsound and should not be used. This is contrary to what was originally thought about these schemes from the initial running on the Hypothetical Model.

The Implicit systems, i.e., not including the Nonlinear ones, provided accurate results at time steps approximately six to twelve times greater, depending upon the rate of change of conditions, than the Explicit systems (calculated using the Leap Frog stability criteria). Both of the two river models clearly showed that there was no point in updating the co-efficients of the linearised systems and that savings in computer time could be achieved because of this.

The Nonlinear methods provided the most accurate results for the larger time steps. Reasonably accurate results were still obtained at time steps approximately twice those mentioned above for the Implicit systems. This, as stated previously, shows the advantage of treating the friction and $\delta A/\delta x$ terms in the manner described. In all the systems programmed no tests were conducted on the effect of large differences in section lengths. However, on considering the section on Consistency, it is suggested that differences should not be large.

As far as economy was concerned, then the most important point was that the Second Explicit system gave consistently the most economical running times. The Implicit and Nonlinear methods only gave running times that were comparable to these for the single River Aire and did not even give this for the Ganges network. However, these considerations depend upon what one considers is accurate and what is not, and if greater divergence had been acceptable then the Implicit and Nonlinear methods would become more economical. A point worth mentioning is that it was apparent from the River Aire results, that had a double sweep system been tried with constant co-efficients, then more economical running times than the Second Explicit system would probably have been obtained with acceptable accuracy. This supports the method's popularity with unbranched channels.

Of the Implicit and Nonlinear systems that worked on the Ganges network then the Sparse Sweep Implicit Method, with constant co-efficients was the most economical. This system gave a better running time than the Second Explicit system at a time step of two hours, but the results were considered to be unacceptable from an accuracy point of view. It is felt that improvements could undoubtedly be made to this method to improve its efficiency.

When considering the performance of the individual numerical methods in solving the Implicit and Nonlinear systems then it is clear that the Gauss-Seidel method did not prove useful. The method not only gave uneconomical running times but failed at time steps less than those from which accurate results were still to be obtained. It was hoped that this method would prove more useful as it is ideally suited to the sparse nature of the equations in open channel networks. The reasons why the system failed have been clearly stated.

The Double Sweep technique was shown to work well for the single River Aire. However, when it worked on the Ganges network, on an iterative basis, it gave uneconomical running times. An important point worth mentioning though, is that the method did not fail by non-convergence when working on the network, until a time step was used that was much greater than that from which accurate results were still just to be obtained. This itself supported the testing of this variation of the basic method, although it must be remembered that a more complex network may alter this statement.

The Sparse Sweep technique which worked with constant co-efficients proved to be the most efficient method of the Implicit and Nonlinear systems when working with the Ganges network. Also, it was shown to converge for time steps of twelve hours in length. The method is, itself, ideally suited for dealing with complex networks. The variation which updated the co-efficients gave grossly uneconomical running times.

The Nonlinear Methods were shown to be quite successful whilst working on the Ganges model, in that they gave very accurate results and, on the whole, converged well. However, the Double Sweep version did not give the better convergence characteristics of the Sparse Sweep version. It was also shown that this latter system did not give economical running times even for very large time steps. However, it is apparent that if a more efficient method was used to invert the Jacobian, then the Nonlinear Method, as a whole may prove to provide economical running times and accurate results for large time steps in mildly varying flows. Further, it was shown that, although the norm reducing step was rarely used with a value less than one, it did, at times prove useful to include the procedure that chose a value less than one to search for convergence. A final point against the use of these methods is their complex nature and the subsequent complexity in programming them.

In all, it is suggested that an explicit system would be the most efficient method to model channel networks. Not only are these methods easy to program, they have been shown to be economical with computer time. Also, by virtue of the fact that they work with small time steps they not only record rapid changes in the boundary data but also provide accurate results. Explicit methods are suggested that revolve around the Leap Frog technique. An excellent one is the two-step Lax-Wendroff method as it offers the advantages of speed and accuracy. Further, it has the ability to work in supercritical flow and can handle discontinuities in the form of bores etc. However, the literature concerning this method only deals with the case where velocity and depth of flow are determined at the same node, and it is therefore suggested, that modifications are made to deal with the staggered net in the space direction. It is likely however, that this method and similar, may give slightly larger running times than the Second Explicit system upon which the above recommendations are made. Although such increases are thought to be only small.

If an implicit method is to be preferred then a method which obtains solutions by a direct means is to be recommended on the grounds of economy and convergence. The application of the Sparse Sweep technique used here proved to be very encouraging and it is thought that improvements could be made to make the application more efficient. However, to make the running times of this system comparable to those of an explicit method, then larger time steps must be used which could provide inaccurate results in rapidly varying flow situations. Such time steps would also, not record rapid changes at the boundaries. Again, it is repeated that there is no point in updating the co-efficients of the linearised systems from an accuracy point of view. A direct method would then obtain solutions after one inversion only.

It is concluded that the Nonlinear Methods developed here would only be useful where very large time steps could be used, i.e., two hours or more, in slowly varying flow situations, for instance in river floods. The basic method does however, depend upon an efficient sparse matrix inversion technique and again improvements could be made in this respect. Further, comparisons could be done to compare this basic method with one that formed the Jacobian directly in each iteration (as in Newton's method), but still retained the facility of using a multiple of these steps less than one. The Nonlinear Method itself however, may prove to be useful in other aspects of hydraulics or civil engineering in general, here one thinks particularly of the pipe network problem.

The theoretical stability criteria developed in Chapter 5 are seen to be complied with on the whole, reasonably well by the Explicit systems. However, instability was experienced with the Implicit systems in certain cases where the relevant criteria predicted that they should be stable. The reason for this was thought to be due to the relatively rapid varying flow situations in which the criteria do not necessarily apply. It is hoped that the treatment given here may help to present a clearer and more complete picture of the problem.

APPENDIX A

Computer Program of the Implicit Gauss-Seidel Method.

```
    *****
    'BEGIN' 'COMMENT' CLPSF149  UNSTEADY FLOW NETWORKS,IMPLICIT SYSTEM;
            'COMMENT' THE FOLLOWING BLOCK READS IN THE LAYOUT OF THE SYSTEM.
             N IS THE NODE ARRAY, NCN IS THE NUMBER OF CONNECTING NODES, NNN
             ARE THE NUMBERS OF THE CONNECTING NODES, NNC ARE THE NUMBERS OF
             THE CONNECTING CHANNELS, TB IS THE TYPE OF BOUNDARY,TNN  THE
             TOTAL NUMBER OF NODES,TNC THE TOTAL NUMBER OF CHANNELS,NT IS
             TYPE OF NODE,IF=1 THEN HEAD,IF=2 THEN VELOCITY,TB IS =1 IF
             HBOUND AND =2 IF QBOUND;
            'INTEGER' J,I,K,T,TNN,TNC,XA,XB,XJ,XI,CA,CJ1,CJ2,CI1,CI2,KC,SN,
BLOCK      1
            TOTALTIME,DT,SS,WS,F1,F2,TOL,COUNT,M,AS,BS,TU,TX,TY,TZ,JUNCT,WN;
            'REAL' G,TWOG,SIDEF,FLOW,SURA,QS,SA,    Q,VOB,VBC,VOA,VAC,VJI,
             QEAVE,ENLOST,A,DX2,AC,CV,R,VJSB,AH,DZ,DH,DV,V,C,AC1,R1,R2,BF,
             BFE,BF1,BF2,DVZH,FRT,SID,VSUM1,VSUM2,DVDX,DXV,DXSUM1,DXSUM2,
             VAVE1,VAVE2,DAVE1,DAVE2,AC2,QDX,SAREAT,QS1,QS2,NEWH,NEWQ;
             TNN:=READ:
             TNC:=TNN-1;
             JUNCT:=READ;
    'BEGIN' 'INTEGER' 'ARRAY' N,NCN,TB,NT[1:TNN],NNN,NNC,           NCA,NNJ,
BLOCK      2
               DIR[1:TNN,1:5],XT[1:14],YT[1:14],               TIME[0:100]:
            'REAL' 'ARRAY' Z,HB,HA,QUE,KI,QB,QA,VB,VA,MUA,NUA,BETTAA,SUMQB.
             SUR,                         BO,MS[1:TNN],DX[1:TNN,1:5],
             XQ[1:14],VH[1:14,1:4];
            'COMMENT' THE FOLLOWING PROCEDURE(S) ARE INSERTED FOR
             THE MODEL ONLY;
            'PROCEDURE' ACON(NJ,H,AC);
BLOCK      3
            'INTEGER' NJ:
            'REAL' H,AC;
             AC:=(BO[NJ] +MS[NJ]*H/2)*H;
            'PROCEDURE' AREA(NJ,H,A);
BLOCK      4
            'INTEGER' NJ:
            'REAL' H,A:
                A:=(BO[NJ]+MS[NJ]*H/2)*H;
               'PROCEDURE' SAREA(NJ,DX,H,SA);
BLOCK        5
               'INTEGER' NJ;
               'REAL' H,SA,DX:
                SA:=(BO[NJ]+MS[NJ]*H)*DX;
               'PROCEDURE' LENGTH(KC,DX):
BLOCK        6
               'INTEGER' KC;
               'REAL' DX:
     'BEGIN'   DX:=200;  'GOTO' STOP:
               'BEGIN' 'SWITCH' SS:=L1,L2,L3,L4,L5,L6,L7:
BLOCK        7

                         'GOTO' SS[KC]:
                          L1:DX:=250.0: 'GOTO' LE;
                          L2:DX:=370.0: 'GOTO' LE;
                          L3:DX:=680.0: 'GOTO' LE;
                          L4:DX:=190.0: 'GOTO' LE;
                          L5:DX:=460.0: 'GOTO' LE;
                          L6:DX:=850.0: 'GOTO' LE;
                          L7:DX:=230.0: 'GOTO' LE;
           LE:'END';
     STOP:'END';
               'PROCEDURE' SIDEQ(NNC,T,QS);
BLOCK        8
               'INTEGER' NNC,T;
               'REAL' QS:
                QS:=0.0;
               'PROCEDURE' CHEZY(NNC,H,C);
BLOCK        9
               'INTEGER'NNC:
               'REAL' H,C:
                C:= 80;
               'PROCEDURE' HRAD(NJ,H,R);
BLOCK        10
               'INTEGER' NJ;
               'REAL' H,R:
                R:=(BO[NJ]+MS[NJ]*H/2)*H/(BO[NJ]+2*H*SQRT(1+MS[NJ]*MS[NJ]/4));
               'PROCEDURE' HBOUND(J,N,T,H);
BLOCK        11
               'INTEGER' N,T,J;
               'REAL' H:
               'BEGIN'
               'INTEGER' IB,JB;
               'BEGIN' 'SWITCH' SW:=L101,L102;
BLOCK        12
                         'GOTO' SW[BS]:
                    L101:'FOR' JB:=1 'STEP' 1 'UNTIL' 14 'DO'
                         'BEGIN' YT[JB]:=READ*3600;
                                 'FOR' IB:=1'STEP'1'UNTIL'4'DO' YH[JB,IB]:=READ;
                         'END';
                          BS:=2:
                    L102:IB:='IF'N=1'OR'N=24'THEN'1'ELSE''IF'N=20'THEN'2'ELSE'
                              'IF'N=11'THEN'3'ELSE'4:
                         'FOR'JB:=1 'STEP' 1 'UNTIL' 14 'DO'
                         'IF' YT[JB+1] 'GE' T 'THEN'
                         'BEGIN' H:=(YH[JB+1,IB]-YH[JB,IB])*(T-YT[JB])/
                                     (YT[JB+1]-YT[JB])+YH[JB,IB]-Z[J]:
                                 'GOTO' STH;
                         'END';
               'END';
         STH:'END';
```

```
             'PROCEDURE' QBOUND(N,T,BF);
BLOCK     13
             'INTEGER' N,T:
             'REAL' BF:
              BF:=20 ;
             'PROCEDURE' BRFADTHC(NJ,H,B);
BLOCK     14
             'INTEGER'NJ;
             'REAL' H,B:
              B:=BU[NJ]+MS[NJ]*H;
             'PROCEDURF' BRFADTHT(NJ,H,B);

BLOCK     15
             'INTEGER' NJ;
             'REAL' H,B:
              B:=BO[NJ]+MS[NJ]*H;
             'PROCEDURE' PARAMQ(J,NEWQ);
BLOCK     16
             'REAL' NEWQ;
             'INTEGER' J;
             'BEGIN'
             'REAL' DH1.DH2,MU,NU,ALPHA,BETTA,QU,LAM,GAM,DHT,
              SIG,RHO,BC,BT.DELTAA,QS1,QS2;
             'COMMENT' XB IS ASSOCIATED WITH H[I+1] AND XA WITH H[I-1];
             'BEGIN' 'SWITCH' S:=L1.L2;
BLOCK     17
                       'GOTO' S[SS];
                   L1:MUA[J]:=ABS(QB[J])*G*DT;
                       NUA[J]:=QB[J]*DT/KI[J];
                       BFTTAA[J]:=G*DT/(2.0*KI[J]):
                       SIDEQ(N[J],T,QS1);
                       SIDEQ(N[J],T+DT,QS2);
                       QUE[J]:=(QS1+QS2)*DT/2.0;
                  L2:'IF' NNN[J,1]>NNN[J,2] 'THEN' 'BEGIN' XB:=NNJ[J,1];
                                                           XA:=NNJ[J,2];
                                                   'END'
                                           'ELSE' 'BEGIN' XA:=NNJ[J,1];
                                                           XB:=NNJ[J,2];
                                                   'END';
                       AH:=(HB[ XA]+HB[ XB]+HA[ XA]+HA[ XB])/4.0;
                       DH1:=(HA[ XB]+HB[ XB])/2.0;
                       DH2:=(HA[ XA]+HB[ XA])/2.0;
                       CHEZY(J    ,AH,C);
                       HRAD(J    ,AH,R);
                       BREADTHC(J    ,AH,BC);
                       BREADTHT(J    ,AH,BT);
                       ACON(J    ,DH1,AC1);
                       ACON(J    ,DH2,AC2);
                       ACON(J    ,AH,AC);
                       DELTAA:=AC1-AC2;
                       MU:=MUA[J]/(C*C*AC*R);
                       NU:=NUA[J]*DFLTAA/(AC*AC);
                       ALPHA:=(QB[J]+QA[J])*(BC+BT)/(4.0*AC);
                       BFTTA:=BETTAA[J]*AC;
                       QU:=QUE[J]/AC;
                       LAM:=1+MU+QU-NU:
                       GAM:=ALPHA+BFTTA;
                       SIG:=ALPHA-BFTTA;
                       RHO:=1-QU:
                       NFWQ:=(SIG*((HA[XB]+Z[XB])-(HB[XA]+Z[XA]))+GAM*((
                       HA[XA]+Z[XA])-(HB[XB]+Z[XB]))+RHO*QB[J])/LAM:
             'END';
             'END';
```

```
               'PROCEDURE' PARAMH(J,NEWH);
   BLOCK     18
               'REAL' NEWH;
               'INTEGER' J;
               'BEGIN'
               'REAL' QDX,SAREAT,SUMQ;
               'INTEGER' A;
               'BEGIN' 'SWITCH' SW:=W1,W2;
   BLOCK     19
                        'GOTO' SW[WS];
                     W1:QUE[J]:=SUMQB[J]:=SUR[J]:=0.0;
                        'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
                        'BEGIN' A:=NNJ[J,I];
                                SIDEQ(N[A],T,QS1);
                                SIDEQ(N[A],T+DT,QS2);
                                QUE[J]:=QUE[J]+(QS1+QS2)*DX[J,I]/2.0;
                                SUMQB[J]:=SUMQB[J]+QB[A]*DIR[J,I];
                                SAREA(NNJ[J,I],DX[J,I],HB[J],SA);
                                SUR[J]:=SUR[J]+SA;
                        'END';
                     W2:SAREAT:=SUR[J];
                         SUMQ:=SUMQB[J];
                        'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
                        'BEGIN' A:=NNJ[J,I];
                                SAREA(NNJ[J,I],DX[J,I],HA[J],SA);
                                SAREAT:=SAREAT+SA;
                                SUMQ:=SUMQ+QA[A]*DIR[J,I];
                        'END';
                         NEWH:=(QUE[J]-SUMQ)*DT/SAREAT+HB[J];
               'END';
               'END';
               'COMMENT' THE NODAL AND CHANNEL SYSTEM IS NOW READ IN INCLUDING
                THE LENGTHS OF THE CHANNELS,KI IS THE LENGTH OF REACH BETWEEN
                TWO HEAD NODES;
               'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
               'BEGIN' N[J]:=READ;
                       NT[J]:=READ;
                       NCN[J]:=READ;
                      'IF' NCN[J]=1 'THEN' TB[J]:=READ;
                      'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
                      'BEGIN' NNN[J,I]:=READ;
                              NNC[J,I]:=READ;
                              DX[J,I]:=READ;
                      'END';
                      'IF' NCN[J]=2 'THEN' KI[J]:=DX[J,1]+DX[J,2];
               'END';
               'COMMENT' THE CHANNEL SYSTEM IS NOW CROSS REFERENCED.
                NNJ[J,I] CONTAINS THE VALUE OF J OF THE NODE NNN[J,I];
               'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
               'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
               'BEGIN' XA:=NNN[J,I];
                      'FOR' M:=1 'STEP' 1 'UNTIL' TNN 'DO'
                      'IF' N[M]=XA 'THEN' 'BEGIN' NNJ[J,I]:=M;
                                                  'GOTO' M1;
                                             'END';
            M1:'END';
               'COMMENT' THE CHANNEL SYSTEM IS NOW OUTPUTED;
                WRITETEXT('('('20S')'NODE'('47S')'CONNECTING')');
                NEWLINE(1);
                WRITETEXT('('('60S')'NODE(S)'('5S')'CHANNEL(S)'('5S')'LENGTH
                ')');
                NEWLINE(2);
```

```
'FOR' J:=1 'STFP' 1 'UNTIL' TNN 'DO'
'BEGIN' SPACE(17);
        PRINT(N[J],3,0);
       'IF' NCN[J]=1 'THEN' 'BEGIN' 'IF' NT[J]=1 'THEN'
                                         WRITETEXT('('HB')')
                                    'ELSE' WRITETEXT('('QB')');
                                     SPACE(34);
                            'END'
                     'ELSE' 'BEGIN' 'IF' NT[J]=1 'THEN'
                                         WRITETEXT('('H')')
                                    'ELSE' WRITETEXT('('Q')');
                                     SPACE(35);
                            'END';
       'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
       'BEGIN' PRINT(NNN[J,I],3,0);
               SPACE(9);
               PRINT(NNC[J,I],3,0);
               SPACE(8);
               PRINT(DX[J,I],3,0);
               NEWLINE(1);
               SPACE(59);
       'END';
        NEWLINE(1);
'END';
'COMMENT' THE INITIAL CONDITIONS ARE NOW READ IN.
 VELOCITIES ARE POSITIVE LEAVING THE NODE WHICH HAS THE
 SMALLEST REF. NUMBER IE.N[J];
'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
'IF' NT[J]=1 'THEN'
'BEGIN' Z[J]:=READ;
        HB[J]:=READ-Z[J];
        HA[J]:=HB[J];
'END'
'ELSE'
'BEGIN' QB[J]:=READ;
        QA[J]:=QB[J];
'END';
'COMMENT' THE IDEALISED CROSS-SECTIONAL CHANNEL
 DATA IS NOW READ IN,IN TERMS OF A BOTTOM WIDTH
 BO AND AN AVERAGE SIDE SLOPE MS;
'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
'BEGIN' BO[J]:=READ;
        MS[J]:=READ;
'END';
'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
'IF' NT[J]=2 'THEN'
'BEGIN' XA:=NNJ[J,1];
       'IF' NCN[J]=1 'THEN' AH:=HB[XA]
       'ELSE'
       'BEGIN' XB:=NNJ[J,2];
               AH:=(HB[XA]+HB[XB])/2.0;
       'END';
        ACON( J ,AH,AC);
        VB[J]:=QB[J]/AC;
'END';
'COMMENT' THE FOLLOWING BLOCK ASSIGNS DIRECTION MULTIPLIERS
 TO QUANTITY NODES AROUND A PARTICULAR HEAD NODE, SO THAT
 FLOW FROM THAT NODE IS POSITIVE AND TOWARDS IT IS NEGATIVE,
 I.E. FLOW FROM NODE N[J] IS QB[J+N]*DIR[J,I];
'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
'IF' NT[J]=1 'THEN'
'BEGIN' 'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
```

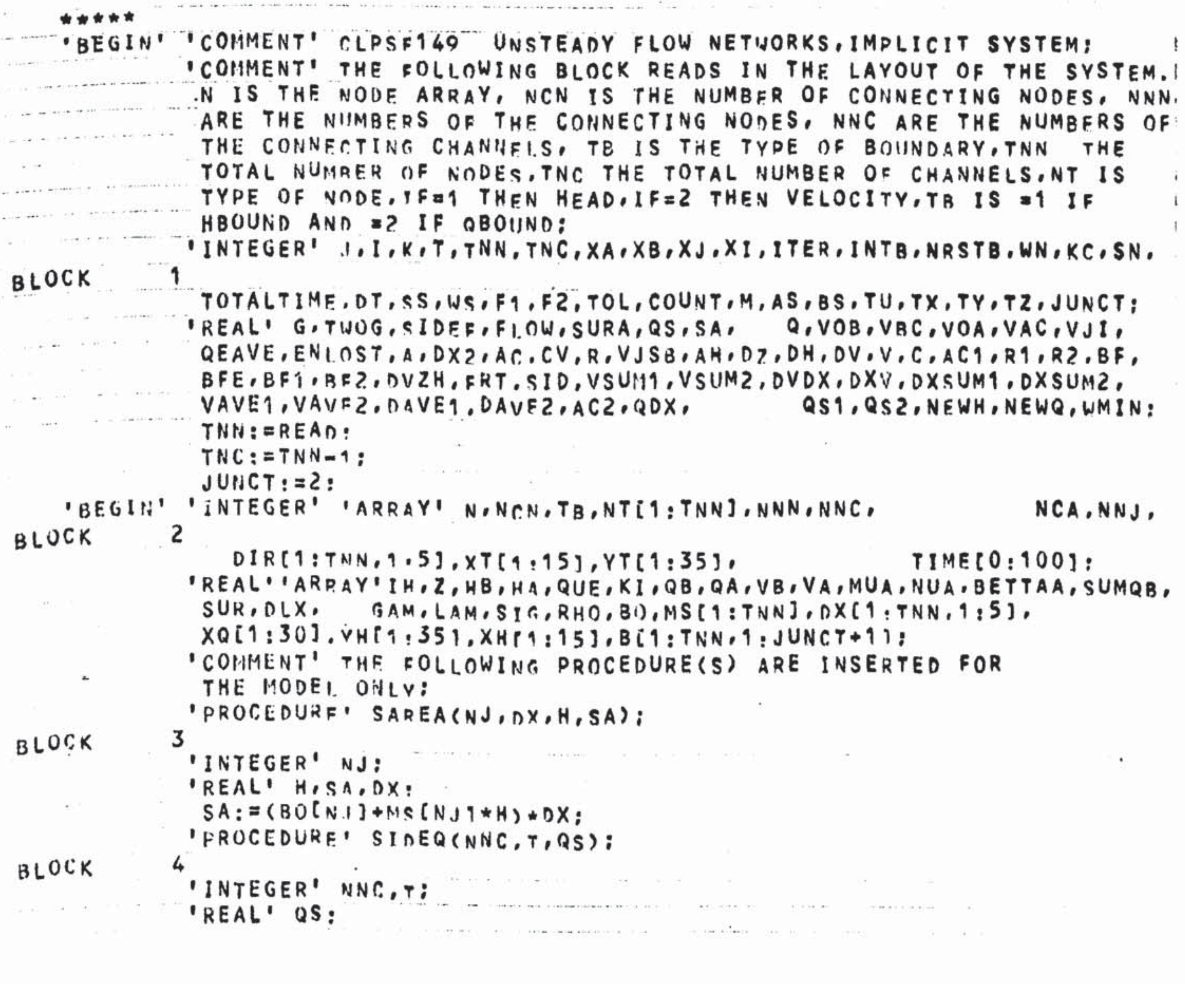
```
           'BEGIN' A:=NNJ[J,I];
                'IF' NCN[A]=1 'THEN'
                'BEGIN' DIR[J,I]:= 'IF' N[J]<N[A] 'THEN' 1
                                                       'ELSE'-1;
                                                       'GOTO' BTM;

                'END';
                'IF' N[J]=NNN[A,1]
                'THEN'
                'BEGIN' DIR[J,I]:='IF' N[J] < NNN[A,2]
                                    'THEN' 1
                                    'ELSE'-1;
                'END'
                'ELSE'
                'BEGIN' DIR[J,I]:='IF' N[J] < NNN[A,1]
                                    'THEN' 1
                                    'ELSE'-1;
                'END';
        BTM:
          'END';
      'END';
       DT:=1000;
       TOTALTIME:= 43200;
       G:=32.175;
       TWOG:=2.0*G;
       AS:=BS:=1;
RECYCLE:
       COUNT:=0;
       TU:=TZ:=0;TY:=0;
      'COMMENT' THE FOLLOWING IS THE TIME VARYING
       COMPUTATIONAL AND OUTPUT BLOCK;
      'FOR' T:=0 'STEP' DT 'UNTIL' TOTALTIME 'DO'
      'BEGIN'
'IF' ENTIER(T/3600)=(T/3600) 'OR' T=44550 'THEN'
'BEGIN'
           PAPERTHROW;
           NEWLINE(4);
           WRITETEXT('('TIME[T]=')');
           PRINT(T,3,0);
           SPACE(10);
           TX:=T/60-TZ*60;
          'IF' TX=60 'THEN' 'BEGIN' TY:=TY+1; TZ:=TZ+1;TX:=0;'END';
          'IF' TY>12 'THEN' 'BEGIN' TY:=TY-12; TU:=1; 'END';
           PRINT(TY,2,0);
           WRITETEXT('('.')');
           PRINT(TX,1,0);
           SPACE(10);
           PRINT(COUNT,3,0);
           NEWLINE(4);
                WRITETEXT('(''('18S')'NODE'('16S')'QUANTITY'('
                14S')'DEPTH'('16S')'BED%A.O.D.'('12S')'TOTAL%
                A.O.D.')');
           NEWLINE(1);
          'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
          'BEGIN'
                NEWLINE(1);
                SPACE(17);
                PRINT(N[J],3,0);
               'IF' NT[J]=1 'THEN' 'BEGIN' SPACE(35);
                                           PRINT(HA[J],2,3);
                                           SPACE(14);
                                           PRINT(Z[J],2,3);
                                           SPACE(14);
```

```
                                                   PRINT(HA[J]+Z[J],2,3);
                                           'END'
                                    'ELSE' 'BEGIN' SPACE(12);
                                                   PRINT(QA[J],6,3);
                                           'END';
               'END';
'END';
               'IF' T=TOTALTIME 'THEN' 'GOTO' EN3;
                SS:=WS:=F1:=F2:=1;
                COUNT:=0;
               'IF' F1=1 'THEN'
               'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
               'IF' NT[J]=1 'THEN' HB[J]:=HA[J]
                             'ELSE' QB[J]:=QA[J];
'IF' T=44550   'THEN' 'GOTO' RECYCLE;
'IF' T=44400   'THEN' DT:=150;
           EN1:TOL:=1;
               COUNT:=COUNT+1;
               'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
               'BEGIN'
                        'IF' NCN[J]=1
                        'THEN' 'BEGIN' 'IF' F1=2 'THEN' 'GOTO' EN2;
                                       'IF' TB[J]=1
                                       'THEN' HBOUND(J,N[J],T+DT,HA[J])
                                       'ELSE' QBOUND(N[J],T+DT,QA[J]);
                                       'GOTO' EN2;
                               'END';
                        'IF' NT[J]=1
                        'THEN' 'BEGIN' PARAMH(J,NEWH);
                                      'IF' ABS(HA[J]- NEWH)>0.001
                                      'THEN' TOL:=2;
                                       HA[J]:=NEWH;
                               'END'
                        'ELSE'
                        'BEGIN' PARAMQ(J,NEWQ);
                               'IF' ABS(QA[J]- NEWQ)>0.001
                               'THEN' TOL:=2;
                                QA[J]:=NEWQ;
                        'END';
           EN2:'END';
                WS:=SS:=F1:=2;
               'IF' TOL=2 'THEN' 'GOTO' EN1;
    EN3:'END';
'END';
'END';
```

APPENDIX B

Computer Program of the Nonlinear Sparse Sweep Method.



```
     *****
    'BEGIN' 'COMMENT' CLPSF149  UNSTEADY FLOW NETWORKS,IMPLICIT SYSTEM;
            'COMMENT' THE FOLLOWING BLOCK READS IN THE LAYOUT OF THE SYSTEM.
            N IS THE NODE ARRAY, NCN IS THE NUMBER OF CONNECTING NODES, NNN
            ARE THE NUMBERS OF THE CONNECTING NODES, NNC ARE THE NUMBERS OF
            THE CONNECTING CHANNELS, TB IS THE TYPE OF BOUNDARY,TNN  THE
            TOTAL NUMBER OF NODES,TNC THE TOTAL NUMBER OF CHANNELS,NT IS
            TYPE OF NODE,IF=1 THEN HEAD,IF=2 THEN VELOCITY,TB IS =1 IF
            HBOUND AND =2 IF QBOUND;
            'INTEGER' J,I,K,T,TNN,TNC,XA,XB,XJ,XI,ITER,INTB,NRSTB,WN,KC,SN,
BLOCK     1
            TOTALTIME,DT,SS,WS,F1,F2,TOL,COUNT,M,AS,BS,TU,TX,TY,TZ,JUNCT;
            'REAL' G,TWOG,SIDEF,FLOW,SURA,QS,SA,    Q,VOB,VBC,VOA,VAC,VJI,
            QEAVE,ENLOST,A,DX2,AC,CV,R,VJSB,AH,DZ,DH,DV,V,C,AC1,R1,R2,BF,
            BFE,BF1,BF2,DVZH,FRT,SID,VSUM1,VSUM2,DVDX,DXV,DXSUM1,DXSUM2,
            VAVE1,VAVE2,DAVE1,DAVE2,AC2,QDX,        QS1,QS2,NEWH,NEWQ,WMIN;
            TNN:=READ;
            TNC:=TNN-1;
            JUNCT:=2;
    'BEGIN' 'INTEGER' 'ARRAY' N,NCN,TB,NT[1:TNN],NNN,NNC,           NCA,NNJ,
BLOCK     2
              DIR[1:TNN,1:5],XT[1:15],YT[1:35],               TIME[0:100];
            'REAL''ARRAY'IH,Z,HB,HA,QUE,KI,QB,QA,VB,VA,MUA,NUA,BETTAA,SUMQB,
            SUR,DLX,   GAM,LAM,SIG,RHO,BO,MS[1:TNN],DX[1:TNN,1:5],
            XQ[1:30],VH[1:35],XH[1:15],B[1:TNN,1:JUNCT+1];
            'COMMENT' THE FOLLOWING PROCEDURE(S) ARE INSERTED FOR
            THE MODEL ONLY;
            'PROCEDURE' SAREA(NJ,DX,H,SA);
BLOCK     3
            'INTEGER' NJ;
            'REAL' H,SA,DX;
            SA:=(BO[NJ]+MS[NJ]*H)*DX;
            'PROCEDURE' SIDEQ(NNC,T,QS);
BLOCK     4
            'INTEGER' NNC,T;
            'REAL' QS;
```

```
              QS:=0.0;
              'PROCEDURE' QBOUND(N,T,BF);
BLOCK      5
              'INTEGER' N,T;
              'REAL' BF;
              'BEGIN'
              'INTEGER' JA;
              'BEGIN' 'SWITCH' SSS:=L100,L200;
BLOCK      6
                     'GOTO' SSS[AS]:
                L100:'FOR' JA:=1 'STEP' 1 'UNTIL' 13 'DO'
                     'BEGIN' XT[JA]:=READ;
                             XQ[JA]:=READ;

                             AS:=2;
                     'END';
                L200:'FOR' JA:=1 'STEP' 1 'UNTIL' 13 'DO'
                     'IF' XT[JA+1] 'GE' T 'THEN'
                     'BEGIN' BF:=(XQ[JA+1]-XQ[JA])*(T-XT[JA])/5400+XQ[JA];
                             'GOTO' STQ;
                     'END';
              'END';
          STQ:'END';
              'PROCEDURE' HBOUND(N,T,H);
BLOCK      7
              'INTEGER' N,T;
              'REAL' H;
              'BEGIN'
              'INTEGER' JB;
              'BEGIN' 'SWITCH' SW:=L101,L102;
BLOCK      8
                     'GOTO' SW[BS]:
                L101:'FOR' JB:=1 'STEP' 1 'UNTIL' 35 'DO'
                     'BEGIN' YT[JB]:=READ;
                             YH[JB]:=READ;
                     'END';
                     'FOR' JB:=1 'STEP' 1 'UNTIL' 15 'DO'
                     'BEGIN' XT[JB]:=READ*3600;
                             XH[JB]:=READ;
                     'END':
                             BS:=2:
                L102:'IF' N=1 'THEN'
                     'BEGIN'
                     'FOR' JB:=1 'STEP' 1 'UNTIL' 35 'DO'
                     'IF' YT[JB+1]'GE' T 'THEN'
                     'BEGIN'H:=(YH[JB+1]-YH[JB])*(T-YT[JB])/(YT[JB+1]-YT[JB])
                             +YH[JB]-Z[N];
                             'GOTO' STH;
                     'END';
                     'END' 'ELSE'
                     'BEGIN'
                     'FOR' JB:=1 'STEP' 1 'UNTIL' 15 'DO'
                     'IF' XT[JB+1]'GE' T 'THEN'
                     'BEGIN'H:=(XH[JB+1]-XH[JB])*(T-XT[JB])/(XT[JB+1]-XT[JB])
                             +XH[JB]-Z[N];
                             'GOTO' STH;
                     'END';
                     'END';
              'END';
          STH:'END';
              'PROCEDURE' HRAD(NJ,H,R);
BLOCK      9
              'INTEGER' NJ;
```

```
                'REAL' H,R:
                 R:=(BO[NJ]+MS[NJ]*H/?)*H/(BO[NJ]+2*H*SQRT(1+MS[NJ]*MS[NJ]/4));
                'PROCEDURE' ACON(NJ,H,AC);
BLOCK      10
                'INTEGER' NJ;
                'REAL' H,AC;
                 AC:=(BO[NJ] +MS[NJ]*H/2)*H;
                'PROCEDURF' ARFA(NJ,H,A);
BLOCK      11
                'INTEGER' NJ;
                'REAL' H,A:

                 A:=(BO[NJ]+MS[NJ]*H/?)*H;
                'PROCEDURE' CHFZY(NNC,H,C);
BLOCK      12
                'INTEGER'NNC;
                'REAL' H,C:
                 C:='IF' T'GE'14400 'THEN' 75 'ELSE' 85;
                'PROCEDURF' BRFADTHC(NJ,H,B);
BLOCK      13
                'INTEGER'NJ;
                'REAL' H,B:
                 B:=BO[NJ]+MS[NJ]*H;
                'PROCEDURF' BRFADTHT(NJ,H,B);
BLOCK      14
                'INTEGER' NJ;
                'REAL' H,B:
                 B:=BO[NJ]+MS[NJ]*H;
                'PROCEDURF' PARAMQ(J,F);
BLOCK      15
                'REAL' F;
                'INTEGER' J;
                'BEGIN'
                'REAL' DH1,DH2,DQDT,DHDT,DADX,DHDX,FRT,SDF,
                          BC,BT.DELTAA,QS1,QS2,XB1,XB2;
                'INTEGER' XA,XB,XB3;
                'BEGIN' 'SWITCH' S:=L1,L2;
BLOCK      16
                    'GOTO' S[SS];
                 L1:SIDEQ(N[J],T,QS1);
                    SIDEQ(N[J],T+DT,QS2);
                    QUE[J]:=(QS1+QS2)/2.0;
               L2:'IF' NNN[J,1]>NNN[J,2] 'THEN' 'BEGIN' XB:=NNJ[J,1]:
                                                         XA:=NNJ[J,2];
                                                  'END'
                                           'ELSE' 'BEGIN' XA:=NNJ[J,1];
                                                         XB:=NNJ[J,2];
                                                  'END';
                    AH:=(HB[ XA]+HB[ XB]+HA[ XA]+HA[ XB])/4.0;
                    DH1:=(HA[ XB]+HB[ XB])/2.0;
                    DH2:=(HA[ XA]+HB[ XA])/2.0;
                    CHEZY(J    ,AH,C);
                    HRAD(J    ,AH,R);
                    BREADTHC(J    ,AH,BC);
                    BREADTHT(J    ,AH,BT);
                    ACON(J    ,DH1.AC1);
                    ACON(J    ,DH?.AC2);
                    ACON(J    ,AH.AC);
                    DFLTAA:=AC1-AC2;
                    DQDT:=(QA[J]-QB[J])/(G*AC*DT);
                    DHDT:=(BC+BT)*(QB[J]+QA[J])*((HA[XB]+HA[XA])
                          -(HB[XB]+HB[XA]))/(4*AC*AC*G*DT);
                    DADX:=(QA[J]+QB[J])*(QA[J]+QB[J])*DELTAA/
                 (4*AC↑3*G*KI[J]);
              DHDX:=((HB[XB]+HA[XB]+2*Z[XB])-(HB[XA]+HA[XA]
                  +2*Z[XA]))/(2*KI[J]);
              FRT:=(QB[J]+QA[J])*ABS(QB[J]+QA[J])/(4*C*C*AC*AC*R);
              SDF:=QUE[J]*(QB[J]+QA[J])/(AC*AC*G);
              F:=(DQDT-DHDT-DADX+DHDX+FRT+SDF)*G*AC*DT:
             'IF'INTB=0'THEN''BEGIN'
                          B[J,1]:=1+(-DHDT-2*DADX+2*FRT+SDF)
                          *G*AC*DT/(QB[J]+QA[J]);

              XB1:=-(BC+BT)*(QB[J]+QA[J])*(1-BC/(4*AC))/(4*AC)+
              (DHDX-2*FRT+2*DADX-SDF)*(G*DT*BC/4);
                              XB2:= G*AC*DT/(2*KI[J]);
                              XB3:='IF' NNN[J,1]<NNN[J,2]
                              'THEN'1 'ELSE'-1;
                              B[J,2]:=XB1-XB3*XB2;
                              B[J,3]:=XB1+XB3*XB2;
                          'END';
        'END';
        'END';
        'PROCEDURE' PARAMH(J,F);
BLOCK  17
        'REAL' F;
        'INTEGER' J;
        'BEGIN'
        'REAL' QDX,SAREAT,SUMQ,    BB,BA;
        'INTEGER' A,I;
        'BEGIN' 'SWITCH' SW:=W1,W2;
BLOCK  18
              'GOTO' SW[WS];
            W1:QUE[J]:=SUMQB[J]:=SUR[J]:=0.0;
              'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
              'BEGIN' A:=NNJ[J,I];
                     SIDEQ(N[A],T,QS1);
                     SIDEQ(N[A],T+DT,QS2);
                     QUE[J]:=QUE[J]+(QS1+QS2)*DX[J,I];
                     SUMQB[J]:=SUMQB[J]+QB[A]*DIR[J,I];
                     SAREA(NNJ[J,I],DX[J,I],HB[J],SA);
                     SUR[J]:=SUR[J]+SA;
              'END';
            W2:SAREAT:=SUR[J];
               SUMQ:=SUMQB[J];
              'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
              'BEGIN' A:=NNJ[J,I];
                     SAREA(NNJ[J,I],DX[J,I],HA[J],SA):
                     SAREAT:=SAREAT+SA;
                     SUMQ:=SUMQ+QA[A]*DIR[J,I];
              'END';
               F:=(HA[J]-HB[J])        -(QUE[J]-SUMQ)*DT/SAREAT;
              'IF' INTB=0 'THEN'
              'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
               B[J,I+1]:=DIR[J,I]*DT/SAREAT;
        'END';
        'END';
        'PROCEDURE' BROYDEN(NN,TOL,ITER);
BLOCK  19
        'REAL' TOL:
        'INTEGER'NN,ITER;
        'COMMENT' THIS PROCEDURE SOLVES A SYSTEM OF SPARSE NON-LINEAR
         EQUATIONS BY BROYDENS FULL-STEP METHOD;
        'BEGIN'
        'REAL' TE  ,NORM,PJTPJ,LASTT;
        'INTEGER' I,J,A;
```

```
            'ARRAY'Y,F.P[1.NN]:
            'PROCEDURE' LOWERNORM(NORM);
BLOCK      20
            'REAL'NORM:
            'BEGIN' 'INTEGER' COUNT;
                     'REAL' NEWNORM:
                     'PROCEDURE' NEWT(COUNT);

BLOCK      21
                     'INTEGER' COUNT;
                     'BEGIN' 'REAL' THETA;
                             'INTEGER' INT;
                             'SWITCH' SWH:=L2,L3;
                              INT:=COUNT-1;
                             'IF' INT>2 'THEN' 'GOTO' L3;
                             'GOTO' SWH[INT];
                           L2:THETA:=NEWNORM/NORM:
                              TE :=(SQRT(1+6*THETA)-1)/(3*THETA); 'GOTO' FIN;
                           L3:TE :=10/(10↑INT); 'GOTO' FIN:
                          FIN:
                     'END' OF NEWT,
                      START OF LOWERNORM;
                   LO:TE:=1.0;
                      LASTT:=0.0;
                      COUNT:=1;
                    L:COMPUTE XE(TE);
                     'IF' COUNT=11 'THEN'
                     'BEGIN''COMMENT' FAILED TO FIND A LOWER NORM SO THE  B
                             MATRIX IS RESTIMATED ANALYTICALLY:
                            'IF' NRSTB>1 'OR' ITER =0 'THEN'
                            'BEGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' NN'DO'
                                   P[J]:=-P[J]; 'GOTO' LO;
                            'END' 'ELSE'
                             RESETB:
                     'END';
                      EUCLID(F,NN,NEWNORM);
                     'IF' NEWNORM<NORM 'THEN' NORM:=NEWNORM
                                      'ELSE''BEGIN' COUNT:=COUNT+1;
                                                    LASTT:=TE:
                                                    NEWT(COUNT);
                                                   'GOTO' L;
                                             'END';
           'END' OF LOWERNORM;
            'PROCEDURE' RESETB:
BLOCK      22
            'BEGIN'                INTB:=0;
                                  'FOR' J:=1 'STEP' 1 'UNTIL' NN 'DO'
                                  'IF' NT[J]=1 'OR' NCN[J]=1 'THEN'
                                  'BEGIN' B[J,1]:=1.0:
                                          'IF' NCN[J]=1 'THEN' B[J,2]:=0.0
                                          'ELSE' PARAMH(J,F[J]):
                                  'END' 'ELSE' PARAMQ(J,F[J]);
                                  'GOTO' AGAIN;
            'END';
           'PROCEDURE' EUCLID(F,NN,NORM);
BLOCK      23
           'INTEGER'NN:
           'REAL' NORM:
           'ARRAY' F:
           'COMMENT' THIS PROCEDURE EVALUATES THE SQUARE OF
            THE EUCLIDEAN NORM OF F;
           'BEGIN' 'INTEGER' I:
                    NORM:=0.0:
                         'FOR' I:=1 'STEP'1'UNTIL' NN'DO' NORM:=NORM+F[I]*F[I];
               'END' OF EUCLID:
               'PROCEDURE' GAUSSELIM(P,B,F);
BLOCK     24
               'ARRAY' P,B,F;

               'BEGIN'
               'INTEGER' WM,X.K,M,XK,M1;
                WM:=UN*2;
               'BEGIN'
               'ARRAY' W[0:WM];
BLOCK     25
               'INTEGER' 'ARRAY' WS[0:3],XI[1:JUNCT+1];
               'PROCEDURE' WREAD(B);
BLOCK     26
               'REAL' B;
               'BEGIN' X:=X+1:
                       W[X]:=B;
               'END';
               'INTEGER' 'PROCEDURE' INT(E);
BLOCK     27
               'REAL' E:
                INT:=ENTIER(E+0.5);
               'PROCEDURE' WDIV(H);
BLOCK     28
               'VALUE' H:
               'INTEGER' H;
               'BEGIN' 'REAL' X,Q.RX,S,Y ,D,FRACPT;
                       'INTEGER' CX,I.J,K,L,M,N,P,TX,V,U,K1,L1,A;
                        A:=H; CX:=WS[A+1]; RX:=S:=1;
                        Y:=Q:=V:=U:=0:
                        WS[0]:=A: A:=WS[A]; K:=-2; I:=J:=1:
                        M:=A-3; TX:=A+1;
                      -'IF' W[TX-1]'NE' -1 'THEN' 'GOTO' L9;
                        W[CX]:=-1:
                        W[WM]:= 1:
                     L8:N:=INT    (W[TX] );
                       'IF' M=A-3 'THEN' 'GOTO' LL1;
                       'FOR' I:=A 'STEP' 3 'UNTIL' M 'DO'
                       'BEGIN' D:=W[I+1];
                               'IF' 0<D 'AND' D<N 'THEN' N:=INT(D);
                               'IF' N'LE'0 'AND' 0<D 'THEN' N:=INT(D);
                       'END';
                   LL1:'IF' N 'LE' 0 'THEN' 'GOTO' L11;
                         X:=0:
                        'IF' N 'NE' INT(W[TX])'THEN' 'GOTO' L2;
                         X:=W[TX+1];
                         TX:=TX+2:
                        'IF' M=A-3  'THEN' 'GOTO' L3;
                     L2:'FOR' L:=A 'STEP' 3 'UNTIL' M 'DO'
                        'BEGIN' 'IF' N 'NE' INT(W[L+1])  'THEN' 'GOTO'L4;
                         K1:=I+2;
                         P:=INT(W[K1]);
                         X:=-W[L]*W[CX+P+1]+X;
                         P:=P+2;
                         W[K1]:=P:
                         W[L+1]:=W[CX+P];
                     L4:'END':
                     L3:'IF'K> -2 'THEN' 'GOTO' L18;
                        'IF'N 'GE' I 'THEN' 'GOTO' L6;
                         M:=M+3;
                         W[M]:=X:
                         P:=INT(W[WM-N+1]);
               W[M+2]:=P;
               W[M+1]:=W[CX+P];
              'GOTO' L8:

          L6:'IF' N 'NE' I 'THEN' 'GOTO'L7;
              'IF' X=0 'THEN' 'GOTO' L9;
               Q:=X:
               RX:=RX*Q:
               D:=ABS(Q/W[TX-1]);
              'IF' D<S 'THEN' S:=D;
              'IF' I>1.1 'THEN' 'GOTO' L1500;
       L1500:
    FRACPT:='IF'(0.2*I)>0 'THEN'(0.2*I)-ENTIER(0.2*I)
                             'ELSE'-((0.2*I)-(ENTIER(0.2*I)+1));
              'IF'RX<0 'THEN' RX:=-1;
              'IF'RX'GE'0 'THEN' RX:=1;
              'IF' Y/X <0 'THEN' V:=V+1;
               Y:=X:
              'GOTO' L8:
          L7:'IF' CX+J+3<WM-I 'THEN' 'GOTO' L10;
         L12:L:=TX-A-3*I;
             'IF' L 'LE' 0 'THEN' 'GOTO' L14;
              L1:=TX-L;
              K1:=CX+J-1;
             'FOR' P:=TX 'STEP' 1 'UNTIL' K1 'DO'
             'BEGIN' W[L1]:=W[P];
                     L1:=L1+1;
             'END';
              CX:=CX-L;
              TX:=TX-L;
             'IF'W[TX]=0 'THEN' 'GOTO' L13;
       L10:L1:=CX+J;
            W[L1]:=N;
            W[L1+1]:=X/Q;
            J:=J+2:
           'GOTO' L8;
       L9:NEWLINE(1);
           PRINT(Q,3,6);
           PRINT(X,3,6);
           PRINT(W[TX-1],3,6);
           PRINT(W[TX],3,6);
           WRITETEXT('('%%DIV%%FAILS%%0%%PIVOT%ON%ROW%%')');
           PRINT(I,3,0);
           NEWLINE(2):
           WRITETEXT('('RUN%%FAILURE%%DUE%%TO%%ABOVE%%CONDITION
           %%IGNORE%%FOLLOWING%%OUTPUT%%%%')');
           NEWLINE(12):
K:=1/0;
        L14:WRITETEXT('('%%MORE%%STORE%%REQUIRED%%')');
             NEWLINE(2):
       L11:'IF'K'NE'-2 'THEN' 'GOTO' L20:
             W[CX+J]:=-I-1:
             J:=J+1;
             W[WM-I]:=J;
            'IF'W[TX]=0 'THEN' 'GOTO' L12:
             I:=I+1;
            'IF' INT(-W[TX]) 'NE' I 'OR' Q=0 'THEN' 'GOTO'L9;
             Q:=0;
             TX:=TX+1;
             M:=A-3;
            'GOTO' L8:
       L13: K:=I-1;
             L1:=WM-K;
        K1:=INT(W[L1]);

        CX:=CX+K1:
        J:=J-K1;
        W[L1]:=W[WM-I]-W[L1];
        W[WM-I]:=0;
  L15:'IF' K=0 'THEN' 'GOTO' L21;
        M:=A-3;
        TX:=INT(W[WM-K+1])+3*I+1+A;
  L17:  D:=W[TX];
       'IF' D>I 'OR' D<0 'THEN' 'GOTO' L8;
        M:=M+3;
        W[M]:=W[TX+1]:
        P:=INT(W[WM-INT(D)]);
        W[M+2]:=P:
        W[M+1]:=W[CX+P];
        TX:=TX+2:
       'GOTO' L17:
  L18:'IF'CX+J+3<WM-I 'THEN''GOTO' L19;
        L:=INT(CX-A-3*I-W[WM-K]);
        WRITETEXT('('%B:%')');
        PRINT(L,2,0):
       'IF' L'LE' 0 'THEN' 'GOTO' L14;
        K1:=CX+J-1;
       'FOR' P:=CX 'STEP' 1 'UNTIL' K1 'DO'
        W[P-L]:=W[P];
        CX:=CX-L:
  L19:  K1:=CX+J:
        W[K1]:=N;
        W[K1 +1]:=X;
        J:=J+2;
       'GOTO' L8:
  L20  :W[CX+J]:=-K+1:
        J:=J+1;
        W[WM-K+1]:=J:
        K:=K-1;
       'GOTO' L15:
  L21:  J:=A;
        K:=1:
  L26:  Q:=0;
      TX:=INT(W[WM-K]);
  L23:  K1:=TX+CX:
       'IF' W[K1] 'LE'0 'THEN' 'GOTO' L22:
       'IF' W[K1+1]=0 'THEN' 'GOTO' L27;
       'IF' Q'NE'0 'THEN' 'GOTO' L24:
        W[J]:= -K:
        J:=J+1;
        Q:=1:
  L24:  W[J]:=W[K1]-I:
        W[J+1]:=W[K1+1];
        J:=J+2;
  L27:  TX:=TX+2;
       'IF' J+4<CX'THEN' 'GOTO' L23;
        L:=INT(WM-I-CX-W[WM-K+1]-1.0);
        WRITETEXT('('%C:%')');
        PRINT(L,2,0):
       'FOR' P:=0 'STEP' 1 'UNTIL' INT(W[WM-K]) 'DO'
        W[P+CX+L]:=W[P+CX];
        CX:=CX+L;
       'GOTO' L23:
  L22:'IF'K=I 'THEN' 'GOTO' L25;
        K:=K+1;
```

```
                    'GOTO' L26:
               L25: W[J]:=0;
                    WS[WS[0]+1]:=J+1;
                    W[WM-3]:=S;
                    W[WM-2]:=V;
                    W[WM-1]:=RX;
                    W[WM]:=U;
           'END';
            X:=-1;
            WS[1]:=0;
            WS[2]:=WN+1;
           'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
           'BEGIN' WREAD(-J);
                   'FOR' K:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
                                   'BEGIN' XI[K]:=1;
                           M:='IF' K=1 'THEN' 0 'ELSE' NNJ[J,XI[K-1]];
                                            M1:=0:
                          'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
                                           'IF' NNJ[J,I]>M 'THEN'
                                           'BEGIN''IF' NNJ[J,I]<NNJ[J,XI[K]]
                                                  'THEN' XI[K]:=I;
                                                   M1:=1;
                                           'END' 'ELSE' 'IF' M1=0
                                           'THEN' XI[K]:=I+1;
                   'END';
                    M:=0;
                   'FOR' K:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
                   'BEGIN' XK:=NNJ[J,XI[K]];
                          'IF'XK>J 'THEN' 'BEGIN' 'IF' M=0 'THEN'
                                                  'BEGIN' WREAD(J);
                                                          WREAD(B[J,1]);
                                                          M:=1;
                                                  'END';
                                           'END';
                           WREAD(XK);
                           WREAD(B[J,XI[K]+1]);
                          'IF'(XK<J'AND'NCN[J]=1)'THEN''BEGIN'WREAD(J);
                                                             WREAD(B[J,1]):
                                                        'END':
                   'END';
            WREAD(TNN+1);
            WREAD(-F[J]);
           'END';
            WREAD(0):
            WDIV(1);
            I:=0;
           'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
           'BEGIN' 'IF' NCN[J]=1 'THEN' 'BEGIN' P[J]:=0.0; I:=I+1;
                                         'END' 'ELSE' P[J]:=W[(3*(J-I))-1]; ;
           'END';
           'END';
           'END';
           'PROCEDURE' COMPUTE XF(TE);
BLOCK    29
           'REAL' TE:
           'COMMENT' THIS PROCEDURE UPDATES THE UNKNOWN VALUES
            HA[J] AND QA[J] AND ASLO CALCULATES THE VECTOR OF
            RESIDUALS F:
           'BEGIN' 'INTEGER' J;
           'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'

           'BEGIN' 'IF' NCN[J]=1
                   'THEN' 'BEGIN''IF' F1=2 'THEN' 'GOTO' EN2;
```

```
                                'IF' TB[J]=1
                                'THEN' HBOUND(N[J],T+DT,HA[J])
                                'ELSE' QBOUND(N[J],T+DT,QA[J]);
                                'GOTO' EN2;
                        'END';
                'IF'TE<1.5 'THEN''BEGIN' 'IF' NT[J]=1
                                       'THEN'HA[J]:=HA[J]+(TE-LASTT)*P[J]
                                       'ELSE'QA[J]:=QA[J]+(TE-LASTT)*P[J];
                                   'END';
         EN2:
        'END' OF LOOP TO SET UP QA AND VA;
        'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
        'IF' NCN[J]=1 'THEN' F[J]:=0.0
                       'ELSE' 'IF' NT[J]=1 'THEN' PARAMH(J,F[J])
                                             'ELSE' PARAMQ(J,F[J]);
         WS:=SS:=F1:=2;
        'END' OF COMPUTE XF.
         START OF BROYDEN;
         NRSTB:=ITER:=INTB:=0;
        'FOR' J:=1 'STEP' 1 'UNTIL' NN 'DO'
        'IF' NT[J]=1 'OR' NCN[J]=1 'THEN'
        'BEGIN' B[J,1]:=1.0;
               'FOR' I:=2 'STEP' 1 'UNTIL' NCN[J]+1 'DO' B[J,I]:=0.0;
        'END' OF LOOP TO SET UP B;
         COMPUTE XF(2);
         EUCLID(F,NN,NORM);
        'IF' NORM<TOL 'THEN' 'GOTO' BTM;
AGAIN:'FOR' J:=1 'STEP' 1 'UNTIL' NN'DO'
        'BEGIN' P[J]:=0.0;
                Y[J]:=F[J];
        'END' OF LOOP TO SET UP P AND Y;
         GAUSSELIM(P,B,F);
         NRSTB:=NRSTB+1;
         LOWERNORM(NORM);
         NRSTB:=0;
         ITER:=ITER+1;
         INTB:=ITER;
        'IF' NORM<TOL 'THEN' 'GOTO' BTM;
        'FOR' J:=1 'STEP' 1 'UNTIL' NN'DO'
        'BEGIN' 'IF' NCN[J]=1 'THEN' 'GOTO' STOP;
                 PJTPJ:=0.0;
                'FOR' I:=1'STEP'1'UNTIL' NCN[J]'DO'
                'BEGIN' A:=NNJ[J,I];
                         PJTPJ:=PJTPJ+P[A]*P[A];
                'END';
                 PJTPJ:=PJTPJ+P[J]*P[J];
                'FOR' I:=1'STEP'1'UNTIL' NCN[J]+1 'DO'
                'BEGIN' A:='IF' I=1 'THEN' J 'ELSE' NNJ[J,I-1];
                         B[J,I]:=B[J,I]+(Y[J]+(F[J]-Y[J])/TE)*P[A]/PJTPJ;
                'END';
 STOP:'END';
        'GOTO'AGAIN;
         BTM:
        'END' OF BROYDEN';
        'COMMENT' THE NODAL AND CHANNEL SYSTEM IS NOW READ IN INCLUDING
         THE LENGTHS OF THE CHANNELS,KI IS THE LENGTH OF REACH BETWEEN
         TWO HEAD NODES;
        'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'

        'BEGIN' N[J]:=READ;
                NT[J]:=READ;
                NCN[J]:=READ;
               'IF' NCN[J]=1 'THEN' TB[J]:=READ;
```

```
           'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
           'BEGIN' NNN[J,I]:=READ;
                   NNC[J,I]:=READ;
                   DX[J,I]:=READ;
           'END';
           'IF' NCN[J]=2 'THEN' KI[J]:=DX[J,1]+DX[J,2];
      'END';
      'COMMENT' THE CHANNEL SYSTEM IS NOW CROSS REFERENCED.
       NNJ[J,I] CONTAINS THE VALUE OF J OF THE NODE NNN[J,I];
      'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
      'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
      'BEGIN' XA:=NNN[J,I];
           'FOR' M:=1 'STEP' 1 'UNTIL' TNN 'DO'
           'IF' N[M]=XA 'THEN' 'BEGIN' NNJ[J,I]:=M;
                                      'GOTO' M1;
                                 'END';
   M1:'END';
      'COMMENT' THE CHANNEL SYSTEM IS NOW OUTPUTED;
       WRITETEXT('('''('20S')'NODE'('47S')'CONNECTING')');
       NEWLINE(1);
       WRITETEXT('('''('60S')'NODE(S)'('5S')'CHANNEL(S)'('5S')'LENGTH
       ')');
       NEWLINE(2);
      'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
      'BEGIN' SPACE(17);
            PRINT(N[J],3,0);
           'IF' NCN[J]=1 'THEN' 'BEGIN' 'IF' NT[J]=1 'THEN'
                                              WRITETEXT('('HB')')
                                         'ELSE' WRITETEXT('('QB')');
                                          SPACE(34);
                                 'END'
                        'ELSE' 'BEGIN' 'IF' NT[J]=1 'THEN'
                                              WRITETEXT('('H')')
                                         'ELSE' WRITETEXT('('Q')');
                                          SPACE(35);
                                 'END';
           'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
           'BEGIN' PRINT(NNN[J,I],3,0);
                   SPACE(9);
                   PRINT(NNC[J,I],3,0);
                   SPACE(8);
                   PRINT(DX[J,I],3,0);
                   NEWLINE(1);
                   SPACE(59);
           'END';
            NEWLINE(1);
      'END';
      'COMMENT' THE INITIAL CONDITIONS ARE NOW READ IN.
       VELOCITIES ARE POSITIVE LEAVING THE NODE WHICH HAS THE
       SMALLEST REF. NUMBER IE.N[J];
      'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
      'IF' NT[J]=1 'THEN'
      'BEGIN' Z[J]:=READ;
            HB[J]:=READ;
HB[J]:=HB[J]-Z[J];
IH[J]:=HB[J];

            HA[J]:=HB[J];
      'END'
      'ELSE'
      'BEGIN' QB[J]:=READ;
            Z[J]:=READ;
            QA[J]:=QB[J];
```

```
         'END';
         'COMMENT' THE IDEALISED CROSS-SECTIONAL CHANNEL
          DATA IS NOW READ IN,IN TERMS OF A BOTTOM WIDTH
          BO AND AN AVERAGE SIDE SLOPE MS;
NEWLINE(6);
         'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
         'BEGIN' WMIN :=READ;
                 MS[J]:=READ;
                 BO[J]:=WMIN+MS[J]*Z[J];
PRINT(J,2,0);
PRINT(MS[J],2,4);
PRINT(BO[J],4,4);
NEWLINE(1);
         'END';
         'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
         'IF' NT[J]=2 'THEN'
         'BEGIN' XA:=NNJ[J,1];
                 'IF' NCN[J]=1 'THEN' AH:=HB[XA]
                 'ELSE'
                 'BEGIN' XB:=NNJ[J,2];
                         AH:=(HB[XA]+HB[XB])/2.0;
                 'END';
                  ACON(N[J],AH,AC);
                  VB[J]:=QB[J]/AC;
                  VA[J]:=VB[J];
         'END';
         'COMMENT' THE FOLLOWING BLOCK ASSIGNS DIRECTION MULTIPLIERS
          TO QUANTITY NODES AROUND A PARTICULAR HEAD NODE, SO THAT
          FLOW FROM THAT NODE IS POSITIVE AND TOWARDS IT IS NEGATIVE,
          I.E. FLOW FROM NODE N[J] IS QB[J+N]*DIR[J,I];
         'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
         'IF' NT[J]=1 'THEN'
         'BEGIN' 'FOR' I:=1 'STEP' 1 'UNTIL' NCN[J] 'DO'
                 'BEGIN' A:=NNJ[J,I];
                          'IF' NCN[A]=1 'THEN'
                          'BEGIN' DIR[J,I]:= 'IF' N[J]<N[A] 'THEN' 1
                                                                'ELSE'-1;
                                                                'GOTO' BTM;
                          'END';
                          'IF' N[J]=NNN[A,1]
                          'THEN'
                          'BEGIN' DIR[J,I]:='IF' N[J] < NNN[A,2]
                                                 'THEN' 1
                                                 'ELSE'-1;
                          'END'
                          'ELSE'
                          'BEGIN' DIR[J,I]:='IF' N[J] < NNN[A,1]
                                                 'THEN' 1
                                                 'ELSE'-1;
                          'END';
              BTM:
                    'END';
         'END';
          WN:=0;

          'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO' WN:=WN+5+2*NCN[J];
          DT:=1800;
          TOTALTIME.= 45000;
          G:=32.175;
          TWOG:=2.0*G;
          AS:=BS:=1;
RECYCLE:
          COUNT:=0;
          TU:=TZ:=0;TY:=6;
         'COMMENT' THE FOLLOWING IS THE TIME VARYING
          COMPUTATIONAL AND OUTPUT BLOCK;
         'FOR' T:=0 'STEP' DT 'UNTIL' TOTALTIME 'DO'
          'BEGIN'
'IF' ENTIER(T/3600)=(T/3600) 'OR' T=45000 'THEN'
'BEGIN'
                'COMMENT' OUTPUT BLOCK;
                 PAPERTHROW;
                 NEWLINE(8);
                 WRITETEXT('('TIME[T]=')');
                 PRINT(T,3,0);
                 SPACE(10);
                 TX:=T/60-TZ*60;
                'IF' TX=60 'THEN' 'BEGIN' TY:=TY+1; TZ:=TZ+1;TX:=0;'END';
                'IF' TY>12 'THEN' 'BEGIN' TY:=TY-12; TU:=1; 'END';
                 PRINT(TY,2,0);
                 WRITETEXT('('.')');
                 PRINT(TX,1,0);
                'IF'TU=0 'THEN' WRITETEXT('('PM')')
                          'ELSE' WRITETEXT('('AM')');
                 SPACE(10);
                 WRITETEXT('('ITERATIONS')');
                 PRINT(COUNT,3,0);
                 NEWLINE(4);
                        WRITETEXT('(''('18S')'NODE'('16S')'QUANTITY'('
                        14S')'DEPTH'('16S')'BED%A.O.D.'('12S')'TOTAL%
                        A.O.D.')');
                 NEWLINE(1);
                'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
                'BEGIN'
                        NEWLINE(1);
                        SPACE(17);
                        PRINT(N[J],3,0);
                       'IF' NT[J]=1 'THEN' 'BEGIN' SPACE(35);
                                                   PRINT(HA[J],2,3);
                                                   SPACE(14);
                                                   PRINT(Z[J],2,3);
                                                   SPACE(14);
                                                   PRINT(HA[J]+Z[J],2,3)
                                                   NEWLINE(1);
                                           'END'
                                    'ELSE' 'BEGIN' SPACE(12);
                                                   PRINT(QA[J],5,3);
                                                   SPACE(34);
                                                   PRINT(Z[J],2,3);
                                                   NEWLINE(1);
                                           'END';
                'END';
'END';
                'IF' T=TOTALTIME 'THEN' 'GOTO' EN3;
                 SS:=WS:=F1:=F2:=1;

                 COUNT:=0;
                'IF' F1=1 'THEN'
                'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
                'IF' NT[J]=1 'THEN' HB[J]:=HA[J]
                             'ELSE' 'BEGIN' QB[J]:=QA[J];
                                            VB[J]:=VA[J];
                                    'END';
'IF' T=45000 'THEN' 'BEGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' TNN 'DO'
                            'IF' NT[J]=1 'THEN' HA[J]:=HB[J]:=IH[J];
                            'GOTO' RECYCLE;
```

```
                    'END';
           EN1:TOL:=1;
                BROYDEN(TNN,1&- 6,ITER);
                COUNT:=ITER;
    EN3:'END';
'END';
'END';
```

REFERENCES

1. ABBOTT, M. R. "Solution of the Unsteady One-Dimensional Equations of Non-Linear Shallow Water Theory by the Lax-Wendroff Method, with applications to Hydraulics", Royal Aircraft Establishment, Technical Report 69179, Aug.1969.

2. AMEIN, M. and FANG, C.S. "Implicit Flood Routing in Natural Channels", Journal of the Hydraulics Division, ASCE, December, 1970, pp.2481-2500.

3. BALLOFFET, A. "One-Dimensional Analysis of Floods and Tides in Open Channels", Journal of the Hydraulics Division, ASCE, Vol.95, No.HY4, Proc. Paper 6695, July, 1969, pp.1429-1451.

4. BALTZER, R. A. and LAI, C., "Computer Simulation of Unsteady Flows in Waterways", Journal of the Hydraulics Division, ASCE, Vol.94, No.HY4, Proc. Paper 6048, July, 1968, pp.1083-1117.

5. BROYDEN, C. G. "A Class of Methods for Solving Nonlinear Simultaneous Equations", Mathematics of Computation, 1965, Vol.19, pp.577-593.

6. BROYDEN, C. G. "The Convergence of an Algorithm for Solving Sparse Nonlinear Systems", Mathematics of Computation, April, 1971, Vol.25, No.114.

7. BRUTSAERT, W. "De Saint-Venant Equations Experimentally Verified", Journal of the Hydraulics Division, ASCE, Vol.97, No.HY9, Proc. Paper 8378, September 1971, pp.1387-1401.

8. DRONKERS, J. J. "Tidal Computations for Rivers, Coastal Areas and Seas", Journal of the Hydraulics Division, ASCE, Vol.95, No.HY1, Proc. Paper 6341, January, 1969, pp.29-77.

9. HENDERSON, F. M. Open Channel Flow, Macmillan Company, N.Y. 1966.

10. JENNINGS, A. "A Sparse Matrix Scheme for the Computer Analysis of Structures", Int.Journal of Computer Mathematics, 1968, Vol.2, pp.1-21.

11. KAMPHUIS, W. J. "Mathematical Tidal Study of St. Lawrence River", Journal of the Hydraulics Division, ASCE, Vol.96, No.HY3, Proc. Paper 7141, March, 1970, pp.643-664.

12. LIGGETT, J. A. and WOOLHISER, D. A. "Difference Solutions of the Shallow-Water Equation", Journal of the Engineering Mechanics Division, ASCE, Vol.93, No.EM2, April, 1967, pp.39-71.

13. LIGGETT, J. A. and WOOLHISER, D. A. "Difference Solutions of the Shallow-Water Equation - Closure", Journal of the Engineering Mechanics Division, ASCE, Vol.95, No.EM1, February 1969, pp.303-311.

14. Modern Computing Methods, H.M.Stationary Office, Notes on Applied Science No.16, National Physical Laboratory.

15. RICHTMYER, R. D. "A Survey of Difference Methods for Non-Steady Fluid Dynamics", NCAR Technical Notes 63-2, Boulder, Colo., 1962.

16. SHUBINSKI, A. M., McCARTY, J. C. and LINDORF, M. R. "Computer Simulation of Estuarial Networks", Journal of the Hydraulics Division, ASCE, Vol.91, No.HY5, September, 1965, pp.33-49.

17. SMITH, G. D. Numerical Solution of Partial Differential Equations, Oxford University Press, 1971.

18. STRANG, G. "Accurate Partial Difference Methods II. Non-Linear Problems". Numerische Mathematik 6, 1964, pp.37-46.

19. STRELKOFF, T. "Numerical Solution of Saint-Venant Equations", Journal of the Hydraulics Division, ASCE, Vol.96, No.HY1, Proc. Paper 7043, January, 1970, pp.223-252.

20. VARGA, R. S. Matrix Iterative Analysis, Prentice Hall, 1962, p.73.

21. WALSH, J. An Introduction to Numerical Analysis, Academic Press, 1966.

22. YEN, B. C. and WENZEL, H. G. "Dynamic Equations for Steady Spatially Varied Flow", Journal of the Hydraulics Division, ASCE, Vol.96, No.HY3, Proc. Paper 7179, March, 1970, pp.801-814.