ICIWRM – 2000, Proceedings of International Conference on Integrated Water Resources Management for Sustainable Development, 19 – 21 December, 2000, New Delhi, India

An efficient solution algorithm for dendritic channel networks using FEM or FDM

DHRUBAJYOTI SEN
Assistant Professor, Department of Civil Engineering, Indian Institute of Technology, Kharagpur – 721 302, India

Abstract
This paper presents an efficient algorithm for solving one-dimensional flow equations through a dendritic channel network system. The equations generated through the finite element or the finite difference formulations can easily solved by applying the proposed algorithm without the requirement of substantial computer memory, even for a large network. An algorithm proposed earlier for linear finite elements has been extended here to cover its applicability towards using higher order finite elements and implicit finite difference schemes. The maximum active memory required in either case is only 2N x 2N where N is the number of branches of the network. The main advantage, perhaps, lies in the fact that the computational nodes of the branches can be numbered independently for each branch. The algorithm is suitable for programming on computers using parallel processing technology.

INTRODUCTION

Steady or unsteady flow routing through a dendritic network of channels with sub-critical flow requires the solution of the relevant equations (i.e., the de St. Venant equations) in all the branches simultaneously (Akan and Yen, 1981; Choi and Molinas, 1993). Mostly, numerical simulations of the phenomena have been attempted through implicit finite difference schemes (Cunge et al. 1980, Chaudhry 1993). Investigators like Cooley and Moin (1976), Keuning (1976), King (1976), Nwaogazie and Tyagi (1984), Addeff and Wang (1985), Szymkiewicz (1991), etc. have also attempted the method of finite elements. Whichever method is adopted, it is well known that when the governing equations together with the mass conservation and energy conservation equations at the junctions are employed to describe the flow in a network of channels, the resulting coefficient matrix involved in the solution process is not banded. Different researches like Cunge and Wegner(1964), Wylie(1972), Fread(1973), Kao(1980) , Schaffranek et al.(1981), Akan and Yen(1981), Joliffe(1984), Ball(1985), Chaudhry and Schulte(1986), Schulte and Chaudhry (1987), Choi and Molinas(1990,1993), Nguyen and Kawano(1995), Naidu et al. (1997) and others have presented different methods for minimizing the computational efforts involved in solving the flow problem in channel networks. Sen and Garg (1998) have presented an algorithm which was shown to efficiently store and solve the system of equations for a dendritic channel network obtained by applying the finite element technique. The algorithm reduces the nodal equations to one node per branch in a forward elimination phase, and then, a system of equations containing the variables of the single node of each branch is solved simultaneously. A back-substitution phase is then used to obtain the remaining variables, independently for each branch.

Sen and Garg (1998) had demonstrated the applicability of their concept for solving the dendritic channel network problems using linear finite elements. The present paper extends the concept and demonstrates its applicability towards solving the equations using higher order finite elements or with any implicit finite difference scheme. As with the original concept of Sen and Garg (1998), a major advantage of the algorithm is that it can easily be programmed on a computer with parallel processing capability. Even on a serial machine, relatively less memory is required than for other algorithms since the reduced coefficients are stored branch-wise. Further, the proposed algorithm does not require any special node numbering scheme.

THE GOVERNING EQUATIONS AND BOUNDARY CONDITIONS

The governing equations for simulating gradually varied unsteady flow are described by the well-known de Saint Venant equations. These equations may be derived from the laws of conservation and momentum and may be expressed as follows:

$$\frac{\delta Q}{\delta t} + \frac{\delta}{\delta x}\left(\frac{Q^2}{A}\right) + g A \frac{\delta h}{\delta x} + g A S_f - g A S_0 = 0 \qquad (1)$$

$$\frac{\delta h}{\delta t} + \frac{1}{B}\frac{\delta Q}{\delta x} - \frac{q}{B} = 0 \qquad (2)$$

where t is time ; x is the longitudinal distance; Q is the discharge ; h is the water depth ; A is the cross sectional area ; B is the free surface width ; S_0 is the bed slope ; S_f is the friction slope ; q is the rate of lateral inflow, and g is the acceleration due to gravity . For channel flow, S_f may be estimated by any of the flow resistance equations, for example the Manning's equation

$$S_f = \frac{n^2 Q|Q|}{A^2 R^{4/3}} \qquad (3)$$

where n is Manning's roughness coefficient and R is the hydraulic radius.

It is to be noted that the equations of continuity and momentum are applied only within the channel branch lengths and different relations are used to link the flow variables at the junctions (i.e., confluence in a dendritic network). The hydraulic conditions at the confluence may be described by the equations of mass and energy conservation. Assuming no change in storage volume within the confluence, the continuity equation can be written as

$$\Sigma Q_i = Q_o \qquad (4)$$

where 'i' stands for the inflow branches and 'o' for the outflow branch . In this study two inflow and one outflow branches have been considered meeting at a confluence, although any number of inflow branches may be taken into account. When the flows in all the

branches meeting at a confluence are subcritical, the equation for energy conservation can be approximated by a kinematic compatibility condition as (Akan and Yen 1981)

$$h_i + Z_i = h_o + Z_o \quad (5)$$

where Z denotes the elevation of the channel bed. If the bed elevation at the confluence of the channels is assumed to be the same for all the channels, then (5) simplifies to:

$$h_i = h_o \quad (6)$$

Equations (4) and (6) constitute the interior boundary conditions. The total number of interior boundary conditions at a confluence is equal to the number of channels meeting there. The possible applications of these boundary conditions have been demonstrated in Fig. 1. It may be observed that a dendritic network of channels will have a single downstream end where an exterior boundary condition in the form of a relation between depth as a function of discharge or depth as a function of time is specified (Cunge et al. 1980). The network may have one or several upstream ends where the boundary conditions prescribed is in the form of specified discharge values as function of time. Other interior boundary conditions may occur within a single branch, for example due to a weir, constrictions like bridges, etc. However, these conditions are omitted from the present study, as the application of this is very straightforward and no special modification of the algorithm is required.

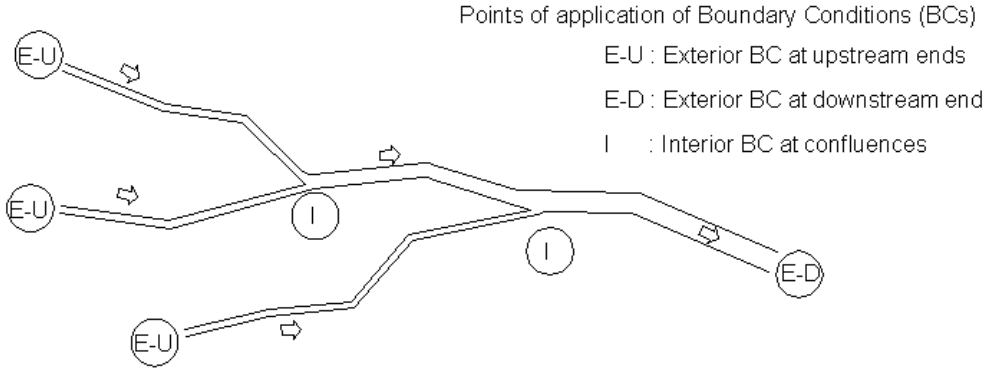

Figure 1. A typical dendritic network of channels showing the various boundary conditions

DISCRETIZATION OF THE FLOW DOMAIN

An essential step involved in solving the flow equations (1) and (2) simultaneously in order to evaluate the unknown variables Q and h along the various branches of any network is the discretization of the solution domain into a finite number of computational nodes. The variables are calculated at these nodes on applying the finite element or the finite difference techniques to the governing equations and solving along with the boundary conditions. For demonstration purposes, a simple three-branched dendritic network is considered in Fig. 2(a), the idealized form of which is discretized as shown in Fig. 2(b).

The procedure of node numbering may be noted: node b·j represents the j^{th} node of the b^{th} branch. This implies that the nodes of one branch are serially numbered either in the upstream or in the downstream direction (as it is done here) but are independently numbered for the different branches. In the finite element method, one may employ linear, quadratic, or higher order elements. Sen and Garg (1998) demonstrated the proposed algorithm for linear elements and it is extended here to quadratic elements. Thus, the elements may be assumed to span between alternate nodes of each branch, as shown in Fig. 2(c). For deriving the finite difference equations, a computational reach is assumed to span between successive nodes as shown in Fig. 2(d).

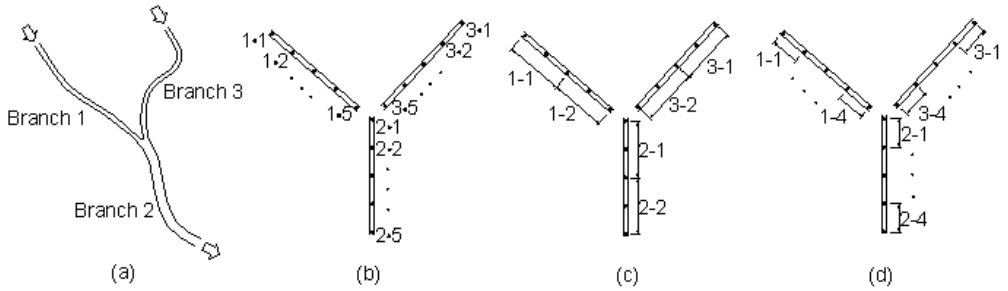

Figure 2. (a) A three branched dendritic network; (b) Computational nodes; (c) Quadratic finite elements; (d) Computational reaches for finite difference calculations

ALGORITHM FOR SOLVING FLOW EQUATIONS WITH FINITE ELEMENTS

The finite element method, using quadratic elements (Fig. 2 c), is used with the Galerkin weighted residual principle to solve equations (1) and (2). We thus obtain element matrices of the form:

$$\left[A \right]^e \left\{ \frac{\partial \Phi}{\partial t} \right\}^e + \left\{ B \right\}^e \tag{7}$$

where $\left\{ \partial \Phi / \partial t \right\}^e$ is the temporal derivative of the variable $\left\{ \Phi \right\}^e = \{Q_1, h_1, Q_2, h_2, Q_3, h_3\}^T$ in which Q_i and h_i are the discharges and depths at the local nodes 'i' of the element 'e'. It may be noted that for quadratic elements, there are three local nodes within each element. Hence, $\left[A \right]^e$ is a square matrix of size 6x6 and $\left\{ B \right\}^e$ is a column matrix of size 6x1.

Since the governing equations are applied branch wise, the element matrices pertaining to each branch are assembled separately, giving the branch equation

$$\left[A \right]^b \left\{ \frac{\partial \Phi}{\partial t} \right\}^b + \left\{ B \right\}^b = 0 \tag{8}$$

where $\{\Phi\}^b = \{ Q_{b\cdot 1}, h_{b\cdot 1}, Q_{b\cdot 2}, h_{b\cdot 2}, \ldots\ldots, Q_{b\cdot N}, h_{b\cdot N} \}^T$ in which Q_j and h_j (j varying from b · 1 to b · N) are the discharges and depths at the j^{th} node of the b^{th} branch. Further assembly of the branch matrices into a global matrix in terms of all the variables of the network system yields an equation of the form

$$[A]\left\{\frac{\partial \Phi}{\partial t}\right\} + \{B\} = 0 \qquad (9)$$

where $\{\Phi\}$ contains the variables of the complete network system.

Equation (9) when applied with the internal and external boundary conditions produce the final system of equation which is required to be solved for the unknown variables. On replacing the temporal derivatives with a difference equation in time (assuming known initial values of the time dependent variables Q and h) we obtain an equation of the form

$$[C]\{\Phi\} + \{D\} = 0 \qquad (10)$$

where $[C]$ is a global coefficient matrix; $\{D\}$ is a global column matrix and $\{\Phi\}$ contains the unknown system variables at time *(t + Δt)*. For the simple three-branched network of Fig. 2, the arrangement of the global equation (10) would be as shown in Fig. 3. Please note that due to space constraint, the subscript showing the node numbers had to be suffixed to the variable name in the figure. The general non-zero coefficients have been represented by dots.

The coefficients of equation (10) are non-linear, i.e., they are functions of the system variables themselves. Equation (10), therefore, cannot be solved directly and the method of Newton-Raphson is used. Denoting equation (10) as

$$f(\Phi) = 0 \qquad (11)$$

we may obtain a Jacobian matrix [W] and a column vector { F } such that

$$[W] = \left[\frac{\partial f}{\partial \Phi}\right] \qquad (12)$$

And

$$\{F\} = f(\Phi) \qquad (13)$$

corresponding to an initial guess of the variable $\{\Phi\}$ as $\{\Phi\}_I$.

The error vector $\{\Delta\Phi\}$ is then obtained from the following equation

$$[W]\{\Delta\Phi\} = \{F\} \qquad (14)$$

and the improved values of the variables $\{\Phi\}$ are worked out from the relationship

$$\{\Phi\}_{I+1} = \{\Phi\}_I - \{\Delta\Phi\} \qquad (15)$$

$\{\Phi\}_I$ is now replaced by $\{\Phi\}_{I+1}$, and equations (12) to (15) are repeatedly solved till $\{\Delta\Phi\}$ decreases below a specified tolerance.

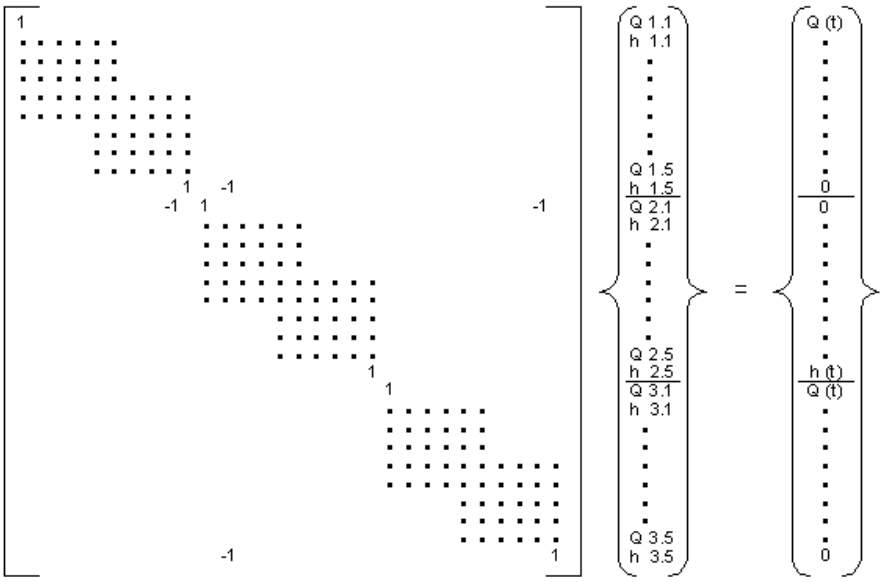

Figure 3. Form of global equation (10) for the three-branched network of Fig. 2 obtained on using quadratic finite elements to solve the flow equations.

NOTE: The upstream boundary conditions have been assumed to be in the form of Q(t) and the downstream boundary condition in the form of h(t).

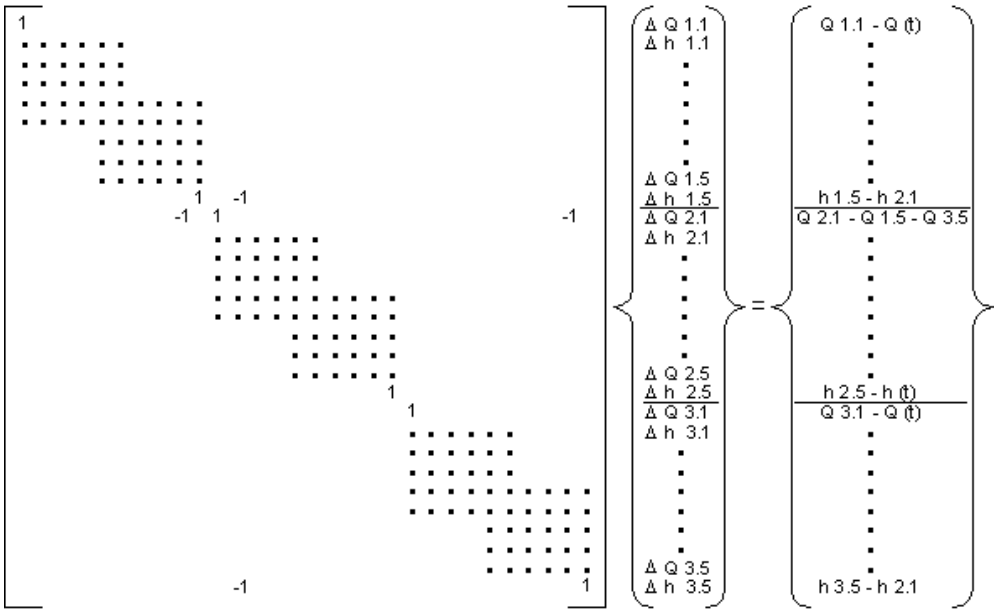

Figure 4. Form of the global equation (14) for the three-branched network of Fig. 2.

For the simple three-branched network of Fig. 2, the arrangement of the global equation (14) would be as shown in Fig. 4. It is seen that the placement of the coefficients is quite similar to that of Fig. 3, although their actual values would be different in the two figures.

The point to observe from the figures is that the matrix $[W]$ is banded except for some terms arising out of the implementation of the internal boundary conditions. This increases the overall bandwidth of the matrix, which in turn increases the computational effort for solving the corresponding system of equation by any standard method, say the Gaussian elimination method. To reduce the bandwidth, the procedure followed by Sen and Garg (1998) is applied to the present system of equation. Thus, the downstream end variables of each branch and their corresponding equations are separated out. Of course, the upstream ends could have been chosen instead. The matrix $[W]$ represented by Fig. 4 is now partitioned and rearranged as shown in Fig. 5.

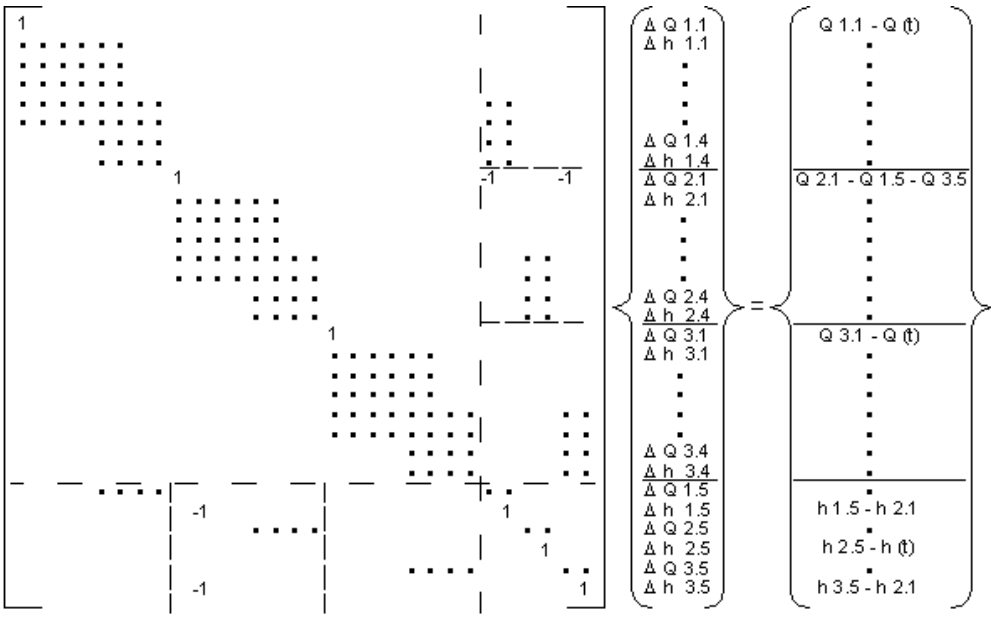

Figure 5. Global equations rearranged after separating out the lowest nodes of each branch.

This double-bound band form of matrix $[W]$ is advantageous in that a forward elimination may be performed for all the variables of the branches, excepting those of the separated out nodes, which would be referred to as the "Active Variables" from now on.

The forward elimination process, as may be observed from Fig. 5, can be carried out one branch at a time, taken in any order. In a parallel-processing machine, this operation could be done simultaneously for all the branches using an array of processors. Further, the non-zero coefficients of a branch in the double-bound band form of matrix $[W]$ lie

within narrow bands, thus requiring very small active memory as well as less solution time.

In the end, we are left with only the equations pertaining to the Active Variables (corresponding to the lower right hand corner of the matrix of Fig. 5). These equations (six in our case) are solved simultaneously to get the error vector $\{\Delta\Phi\}$ corresponding to the Active Variables.

In the back-substitution phase, the Active Variables of each branch are used to evaluate other variables of the branch. As in the forward elimination phase, the operation may be carried out for one branch at a time or simultaneously (as in a parallel processor).

In the entire process, the number of simultaneous equations to be solved is just twice the number of branches in the network. Comparing the present results with that obtained by Sen and Garg (1998), it may be concluded that the order of element (i.e., linear, quadratic, cubic, etc.) would not affect the solution time significantly. The computational scheme also shows that the node numbers for each branch need not have any relation to that of the other branches and each branch may be numbered independently.

ALGORITHM FOR SOLVING FLOW EQUATIONS WITH IMPLICIT FINITE DIFFERENCE SCHEMES

We shall now demonstrate the applicability of the algorithm towards solving the system of equation obtained by applying the finite difference procedure. For convenience, we refer to the space interval in a computational grid as a reach, as shown in Fig. 2 (d). Application of a standard finite difference scheme, say the Preissmann four-point implicit scheme (Cunge et al., 1980), to equations (1) and (2) results in a system of equation for a reach as:

$$[P]^r \left\{\frac{\partial \Phi}{\partial t}\right\}^r + \{Q\}^r = 0 \qquad (16)$$

where $\left\{\partial \Phi / \partial t\right\}^r$ is the temporal derivative of the variable $\{\Phi\}^r = \{ Q_1 , h_1 , Q_2 , h_2 \}^T$ in which Q_i and h_i are the discharges and depths at the end nodes 'i' of the reach 'r'; i varying between 1 and 2. Derived from two equations and involving four variables, matrix $[P]^r$ would be of size 2x4 and $\{Q\}^r$ of size 2x1.

On writing all the reach equations for a branch, we obtain the branch equation

$$[P]^b \left\{\frac{\partial \Phi}{\partial t}\right\}^b + \{Q\}^b = 0 \qquad (17)$$

where $\{\Phi\}^b = \{ Q_{b.1} , h_{b.1} , Q_{b.2} , h_{b.2} , \ldots\ldots, Q_{b.N} , h_{b.N} \}^T$ in which Q_j and h_j (j varying from $b \cdot 1$ to $b \cdot N$) are the discharges and depths at the j^{th} node of the b^{th} branch.

The branch equations can now be assembled for the entire network and the appropriate internal and external boundary conditions applied. This leads to a global system of equations of the form

$$[\,P\,]\left\{\frac{\partial \Phi}{\partial t}\right\}+\{\,Q\,\}=0 \tag{18}$$

where $\{\,\Phi\,\}$ contains all the variables of the system.

On replacing the temporal derivatives with a difference equation in time (assuming known initial values of the time dependent variables Q and h) we obtain an equation of the form

$$[\,R\,]\{\,\Phi\,\}+\{\,S\,\}=0 \tag{19}$$

where $[\,R\,]$ is a global coefficient matrix; $\{\,S\,\}$ is a global column matrix and $\{\,\Phi\,\}$ contains the unknown system variables at time *(t + Δt)*. Equation (19) is quite similar to equation (10) and the solution procedure is also similar. Denoting equation (19) as

$$f\,(\Phi)=0 \tag{20}$$

we may obtain a Jacobian matrix $[\,W\,]$ and a column vector { F } as in equation (12) and (13). The corresponding global matrix (14) for the example network of Fig. 2 would be as shown in Figure. 6.

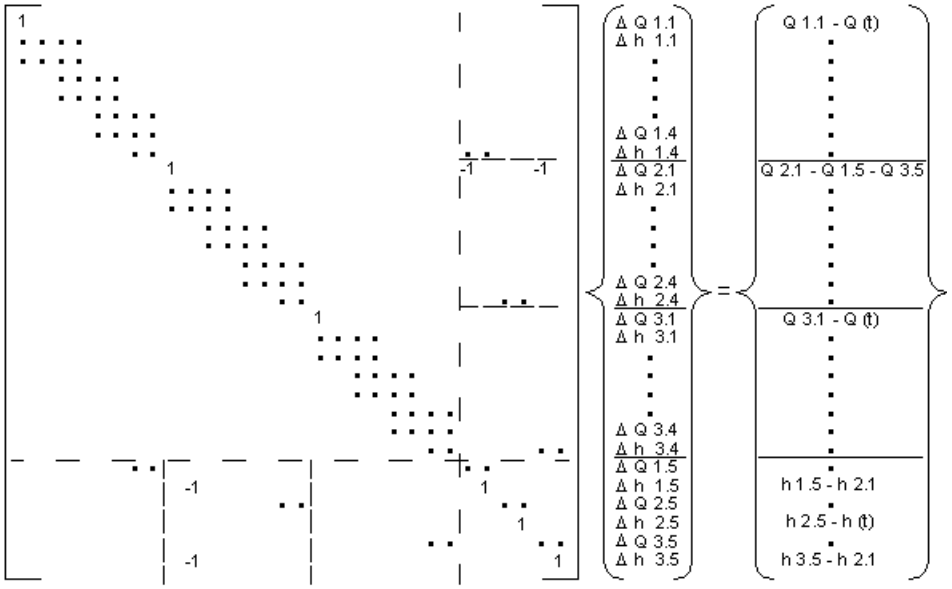

Figure 6. Form of the global equation (14) for the three-branched network of Fig. 2 obtained on using an implicit finite difference scheme to solve the flow equations.

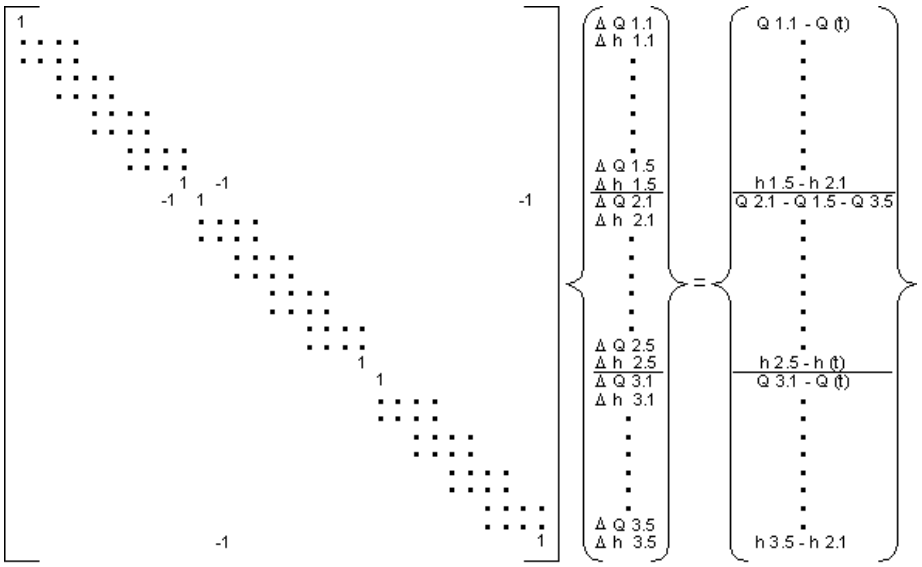

Figure 7. Global equations of Fig. 6 rearranged after separating out the lowest nodes of each branch.

Here again, it is noted that matrix $[W]$ is banded except for the internal boundary condition terms. As in the finite element implementation, matrix $[W]$ is partitioned as shown in Figure. 7 considering the downstream variables of each branch as the Active Variables. Thus, the forward elimination phase of the Gaussian elimination routine may be carried out for all the branches independently. What remain finally are the equations containing the Active Variables, which are then solved simultaneously. Lastly, the back-substitution phase is carried out, again taking the branches independently.

CONCLUSION

An efficient solution algorithm for solving the open channel flow equations with the method of finite elements or finite differences is developed to route flood hydrograph or carry out steady state gradually varied flow computations within a dendritic channel network. Using the algorithm, the maximum size of active matrix storage required during simultaneous solution of equations is reduced to the order of twice the number of branches in the network. The algorithm is organized in three phases: the forward elimination phase, the simultaneous solution phase, and the back-substitution phase. The main phases, namely the first and the last, may be processed independently for each branch. This allows the algorithm to be adapted to parallel processors as well. Even for a serial machine, since all the branches need not be taken together, the memory requirement at any point of time would be that for storing only the non-zero coefficients of one branch. The algorithm does not require any special node or element numbering schemes. The solution algorithm, though presented for dendritic networks, may be extended to looped channel networks as well.

References

Adeff, S. E. and Wang, S. Y. (1985) “Hydrodynamic model of river flow on a microcomputer” in Hydraulics and Hydrology in the Small Computer Age, Volume. 2, W. R. Wardrop (ed.), ASCE, Reston, Va., 1017-1023.

Akan, A. O., and Yen, B. C. (1981) “Diffusion wave flood routing in channel networks.” J. Hydr. Engrg. , ASCE, 107(6), 719-732.

Ball, J. E. (1985) “An algorithm for routing unsteady flows in urban drainage networks” J Hydr. Res., 23(4), 327-341.

Chaudhry, M. H (1993) Open channel hydraulics, Prentice Hall of India Ltd., New Delhi.

Chaudhry, M. H., and Schulte, A. (1986) “Computation of steady-state, gradually varied flows in parallel channels.”, Can. J. Civ. Engrg., Ottawa, Canada, 13(1) , 39-45.

Choi, G. W., and Molinas, A. (1990) “Watershed hydrograph simulation through dendritic channel networks.” in Watershed Planning and Analysis in Action, R. E. Riggins, E. B. Jones, R. Singh, and P. A. Rechard (eds.), ASCE, 303-312.

Choi, G. W., and Molinas, A. (1993) “Simultaneous solution algorithm for channel network modelling.” Water Resources Res., 29(2), 321-328.

Cooley, R. L., and Moin, S. A. (1976) “Finite Element solution of St. Venant equation”, J. Hydr. Engrg., ASCE, 102(6), 759-775.

Cunge, J. A., Holly, Jr. F. M., and Verwey, A. (1980) Practical aspects of computational river hydraulics. Pitman Publishing, London, U.K.

Cunge, J. A., and Wegner, M. (1964) “Integration numerique des equations d’ecoulement de Barre de St. Venant par un schema implicite de differences fines.” La Houille Blanche, Grenoble, France, 19(1), 33-39.

Fread, D.L. (1973) “Technique for implicit dynamic routing in rivers with tributaries.” Water Resources Res., 9(4), 918-926.

Joliffe, I.B. (1984). “Computation of dynamic waves in channel networks” J. Hydr. Engrg., ASCE, 110(10), 1358-1370.

Kao, K. H. (1980) “Improved implicit procedure for multichannel surge computations.” Can. J. Civ. Engrg., Ottawa, Canada, 7, 501-512.

Naidu, B.J., Bhallamudi, S. Murty and Narasimhan, S. (1997) “GVF computation in tree type channel network.”, J. Hydr. Engrg. , ASCE, 123(8), 700-708.

Nguyen, Q. K., and Kawano, H. (1995). “Simultaneous solution for flood routing in channel networks.” J. Hydr. Engrg. , ASCE, 121(10), 744-750.

Preissmann, A. (1961) “Propagation des intumences dans les canaux et rivieres”. Paper presented at the 1961 First Congress of the French Association for Computation, AFCAL Grenoble, France, pp 433-442.

Schaffranek, R. W., Baltzer, R. A., and Goldberg, D. E. (1981) “A model for simulation of flow in singular and interconnected channels.” Techniques of water resources investigations of the United States geological Survey, U.S. Geological Survey, Washington, D.C.

Schulte, A. M. , and Chaudhry, M. H. (1987) “Gradually varied flows in open channel netwroks” , J Hydr. Res., 25(3), 357-371.

Sen, D. J., and Garg, N. K. (1998) “An efficient solution algorithm for dendritic channel networks using FEM”, J. Hydr. Engrg. , ASCE, 124(8), 831-839.

Szymkiewicz, R. (1991) “Finite element method for the solution of St. Venant equation in an open channel network” J. of Hydrology, Amsterdam, The Netherlands, 122, 275-281.

Wylie, E. B. (1972) “Water surface profiles in divided channels.” J. Hydr. Res., 10(3), 352-341.