Comparison of Gradually Varied Flow Computation Algorithms for Open-Channel Network

Adlul Islam[1]; N. S. Raghuwanshi[2]; R. Singh[3]; and D. J. Sen[4]

Abstract: This paper presents a comparison of two algorithms—the forward-elimination and branch-segment transformation equations—for separating out end-node variables for each branch to model both steady and unsteady flows in branched and looped canal networks. In addition, the performance of the recursive forward-elimination method is compared with the standard forward-elimination method. The Saint–Venant equations are discretized using the four-point implicit Preissmann scheme, and the resulting nonlinear system of equations is solved using the Newton–Raphson method. The algorithm using branch-segment transformation equations is found to be at least five times faster than the algorithm using the forward-elimination method. Further, the algorithm using branch-segment transformation equations requires less computer storage than the algorithm using the forward-elimination method, particularly when only nonzero elements of the global matrix are stored. Comparison between the Gauss-elimination method and the sparse matrix solution technique for the solution of the global matrix revealed that the sparse matrix solution technique takes less computational time than the Gauss-elimination method.

DOI: 10.1061/(ASCE)0733-9437(2005)131:5(457)

CE Database subject headings: Simulation; Gradually varied flow; Open channels; Algorithms; Computation; Networks; Hydraulic models.

Introduction

Hydraulic simulation models are frequently used for flow routing in open-channel networks. Over the years, a number of hydraulic simulation models have been developed to study the flow behavior in canal systems. The flow in an open-channel network is represented by unsteady, gradually varying, one-dimensional flow equations, commonly known as the Saint–Venant equations. The Saint–Venant equations, which consist of equations for conservation of mass and momentum, are nonlinear hyperbolic partial differential equations and cannot be solved analytically. Among different numerical methods, the implicit finite-difference method has been widely used for the solution of one-dimensional unsteady open-channel flow problems (Amein and Fang 1970; Amein and Chu 1975; Joliffe 1984; Liu et al. 1992; Choi and Molinas 1993; Nguyen and Kawano 1995; Sen and Garg 2002).

Flow modeling in a single channel using a four-point implicit Preissmann scheme is relatively simple, as it involves solution of a banded set of linear equations that can be solved using an efficient banded matrix solver. However, the banded nature of the resulting system of equations is lost while modeling flow in a channel network consisting of a number of interconnected channels joined at a number of junction points. This is due to the presence of junction continuity and energy equations. Further, due to the presence of backwater effects at the channel junctions, the entire network needs to be considered as a single unit, and flow in all channels needs to be simulated simultaneously (Akan and Yen 1981). Simultaneous solution of the entire channel network requires large computer memory and computational effort. To reduce this computational difficulty, a number of algorithms for simulation of flow in a channel network have been developed. Fread (1973) used an iterative method to simulate transient flow in a canal network assuming tributary flow as lateral inflow. Akan and Yen (1981) used overlapping segment techniques so that a large network could be decomposed and solved as a series of smaller Y-shaped networks. Joliffe (1984) used a sparse matrix algorithm to store and solve the linearized set of equations. Swain and Chin (1990) employed a matrix diagonalization scheme to make the matrix more diagonally banded and solved the matrix using a banded matrix routine.

Other studies have considered specific node-numbering schemes to reduce the bandwidth of the resulting solution matrix (Kao 1980; Chaudhry and Schulte 1986; Schulte and Chaudhry 1987) and different forms of recursive relationships (Choi and Molinas 1993; Nguyen and Kawano 1995). Use of a specific node–numbering scheme for a general channel network is difficult, whereas use of recursive relationships requires relatively less computer storage and computation time. The recursive relationships developed by Choi and Molinas (1993) and Nguyen and Kawano (1995) require simultaneous solution of $2p \times 4$ matrix, where p is the total number of computational points in the network. But these algorithms have certain limitations. The algo-

[1]Research Scholar, Dept. of Agricultural & Food Engineering, Indian Institute of Technology, Kharagpur-721 302, India.

[2]Professor, Dept. of Agricultural & Food Engineering, Indian Institute of Technology, Kharagpur-721 302, India. E-mail: nsr@agfe.iitkgp.ernet.in

[3]Professor, Dept. of Agricultural & Food Engineering, Indian Institute of Technology, Kharagpur-721 302, India.

[4]Associate Professor, Dept. of Civil Engineering, Indian Institute of Technology, Kharagpur-721 302, India.

Note. Discussion open until March 1, 2006. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on May 18, 2004; approved on August 24, 2004. This paper is part of the *Journal of Irrigation and Drainage Engineering*, Vol. 131, No. 5, October 1, 2005. ©ASCE, ISSN 0733-9437/2005/5-457–465/$25.00.

rithm presented by Choi and Molinas (1993) is limited to three converging or diverging segments at the junction, whereas the algorithm presented by Nguyen and Kawano (1995) is applicable to nonlooped channel networks containing junctions of up to four branches, and requires a specific node-numbering order.

The other category of algorithms involves reducing the number of equations for the simultaneous solution phase (Wood et al. 1975; Schaffranek et al. 1981; Sen and Garg 2002). The simultaneous solution algorithms of Schaffranek et al. (1981) and Sen and Garg (2002) require simultaneous solution of $4M \times 4M$ matrix (M is the number of branches), and are applicable in looped as well as dendritic channel networks.

The simultaneous solution algorithm presented by Sen and Garg (2002) uses the Newton–Raphson method for the solution of the nonlinear system of equations. The forward-elimination phase of this algorithm separates out the end nodes of each branch and gives two equations per branch. This phase requires solution of $2(N-1)\times 2N$ matrix, where N is the number of computational points/nodes per branch. The variables corresponding to end nodes (active variables) of each branch (obtained after the forward-elimination phase), along with necessary boundary conditions, result in a global matrix of size $4M \times 4M$. This global matrix is then solved using the Gauss elimination method in the simultaneous-solution phase. The flow variables at the intermediate nodes for each branch are computed in the back substitution phase. This solution algorithm is reported to be 80 times faster than the band solver, and has reduced storage requirements. This algorithm, however, does not take advantage of the banded nature of equations and solves full equations in the simultaneous-solution phase. Further, the matrix is banded in the forward-elimination phase and thus an efficient solution and more compact storage are possible. Fread (1971) presented recurrent formulae for solving banded systems of equations to reduce the computer storage requirement and computational time. In comparison to the standard Gauss-elimination method, this approach reduced the storage requirement from $4N^2$ to $8N$ and the number of computations from $2(N^2+2N^3)$ to $38N$ (Fread 1973).

In the branch model developed by Schaffranek et al. (1981), the Saint–Venant equations are linearized by splitting up each term in the equations into a linear term and a coefficient that is a function of the independent variable. This model uses branch-segment transformation equations to relate the end nodes of a branch, and the resultant branch equations ($2M$ equations) along with the boundary conditions are solved using the Gauss-elimination method with maximum pivot strategy. This model also requires simultaneous solution of $4M \times 4M$ matrix. Swain and Chin (1990) observed that for stringent-solution convergence criterion (i.e., stage convergence of 0.003 m and discharge convergence of 2.83×10^{-3} m^3/s) the network model, which uses the Newton–Raphson method for solving the systems of nonlinear partial differential equations, is three times faster than the branch model. However, for larger values of the convergence criteria (i.e., stage convergence of 0.003 m and discharge convergence greater than 2.83×10^{-2} m^3/s), the branch model solves the system of equations three to five times faster than the network model.

This paper presents a comparison of two algorithms for modeling flow in branched and looped canal networks. The first algorithm (Algorithm-1) uses the forward-elimination approach (Sen and Garg 2002) and the second algorithm (Algorithm-2) uses branch-segment transformation equations (Schaffranek et al. 1981) for linking end-nodes of a branch. To reduce the storage requirements and computation time in the forward-elimination phase for Algorithm-1, only nonzero elements $[2(N-1)\times 4]$ are stored and solved efficiently using recursive relationships. In addition, the performance of this algorithm is compared with the standard forward-elimination method. Further, a comparison is made between the Gauss-elimination method and the sparse matrix solution technique for the solution of the global matrix.

Governing Equations

The gradually varied one-dimensional unsteady open-channel flow equations are described by the Saint–Venant equations. These equations describe the equation of continuity (conservation of mass) and equation of motion (conservation of momentum), and are expressed as

Continuity equation

$$\frac{\partial Q}{\partial x}+\frac{\partial A}{\partial t}-q=0 \tag{1}$$

Dynamic equation

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{\beta Q^2}{A}\right)+gA\frac{\partial Z}{\partial x}+gAS_f=0 \tag{2}$$

where Q=discharge; A=cross-sectional area; q=lateral inflow; β=momentum correction factor; g=acceleration due to gravity; Z=water surface elevation above horizontal datum; S_f=friction slope; x=longitudinal distance along the channel length; and t=time.

The friction slope term S_f can be estimated using the Manning's equation

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

where n=Manning's roughness coefficient; and R=hydraulic radius (ratio of the cross-sectional area to the wetted perimeter).

Boundary Conditions

The numerical solution of the Saint–Venant equations requires specifications of the boundary conditions at the source and outlet nodes. The continuity and momentum equations described above are applicable within channel reaches only. For other elements of a canal network such as junctions or hydraulic structures, two equations equivalent to the Saint–Venant equations are required. The hydraulic conditions at the junctions can be expressed by mass and energy conservation equations. Assuming no change in storage volume within the junction, the continuity equation can be written as:

$$\sum Q_i=\sum Q_o \tag{4}$$

Assuming that the junction losses and the differences in velocity heads at the junctions are negligible, the energy conservation equation at the junction points can be approximated as (Akan and Yen 1981; Naidu et al. 1997)

$$h_i+z_i=h_o+z_o \tag{5}$$

where h=depth of water; z=bed elevation; and subscripts i and o represent inflow and outflow, respectively. If the bed elevations of the channels meeting at a junction are the same, Eq. (5) can be written as

J. Irrig. Drain Eng. 2005.131:457-465.

$$\begin{matrix} \Delta Z_2 & \Delta Q_2 & \Delta Z_3 & \Delta Q_3 & \Delta Z_4 & \Delta Q_4 & & \Delta Z_{N-1} & \Delta Q_{N-1} & \Delta Z_1 & \Delta Q_1 & \Delta Z_N & \Delta Q_N \end{matrix}$$

$$\begin{bmatrix}
a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} & & & & & & & & & \\
a_{4,3} & a_{4,4} & a_{4,5} & a_{4,6} & & & & & & & & & \\
 & & a_{5,5} & a_{5,6} & a_{5,7} & a_{5,8} & & & & & & & \\
 & & a_{6,5} & a_{6,6} & a_{6,7} & a_{6,8} & & & & & & & \\
 & & & & & & \bullet & & & & & & \\
 & & & & & & & \bullet & & & & & \\
 & & & & & & & a_{2N-3,2N-3} & a_{2N-3,2N-2} & & & a_{2N-3,2N-1} & a_{2N-3,2N} \\
 & & & & & & & a_{2N-2,2N-3} & a_{2N-2,2N-2} & & & a_{2N-2,2N-1} & a_{2N-2,2N} \\
a_{1,3} & a_{1,4} & & & & & & & & a_{1,1} & a_{1,2} & & \\
a_{2,3} & a_{2,4} & & & & & & & & a_{2,1} & a_{2,2} & &
\end{bmatrix} = \begin{bmatrix} Fc_2 \\ Fm_2 \\ Fc_3 \\ Fm_3 \\ \bullet \\ \bullet \\ Fc_{N-1} \\ Fm_{N-1} \\ Fc_1 \\ Fm_1 \end{bmatrix}$$

Fig. 1. Jacobian matrix for a branch

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

Solution Algorithm

For the solution of the Saint–Venant equations, Eqs. (1) and (2) are discretized using the four-point implicit Preissmann scheme (Cunge et al. 1980). Discretization of Eq. (1) and Eq. (2) results in a system of nonlinear equations. These systems of nonlinear equations are then solved using an iterative Newton–Raphson method. Application of the Newton–Raphson method for the jth section, which links two consecutive computational points j and $j+1$, results in the following equation

$$\mathbf{B}_j \Delta \mathbf{S}_j^{m+1} + \mathbf{C}_j \Delta \mathbf{S}_{j+1}^{m+1} + \mathbf{D}_j = 0 \quad (7)$$

where

$$\Delta \mathbf{S}_{j+1}^{m+1} = \begin{bmatrix} \Delta Z_{j+1}^{m+1} \\ \Delta Q_{j+1}^{m+1} \end{bmatrix}; \quad \mathbf{B}_j = \begin{bmatrix} a_{2j-1,2j-1} & a_{2j-1,2j} \\ a_{2j,2j-1} & a_{2j,2j} \end{bmatrix}$$

$$\mathbf{C}_j = \begin{bmatrix} a_{2j-1,2j+1} & a_{2j-1,2j+2} \\ a_{2j,2j+1} & a_{2j,2j+2} \end{bmatrix}; \quad \mathbf{D}_j = \begin{bmatrix} RFc_j \\ RFm_j \end{bmatrix}$$

where $j = 1, 2, 3, \ldots, (N-1)$ and

$$a_{2j-1,2j-1} = \frac{\partial Fc_j}{\partial Z_j}; \quad a_{2j-1,2j} = \frac{\partial Fc_j}{\partial Q_j}$$

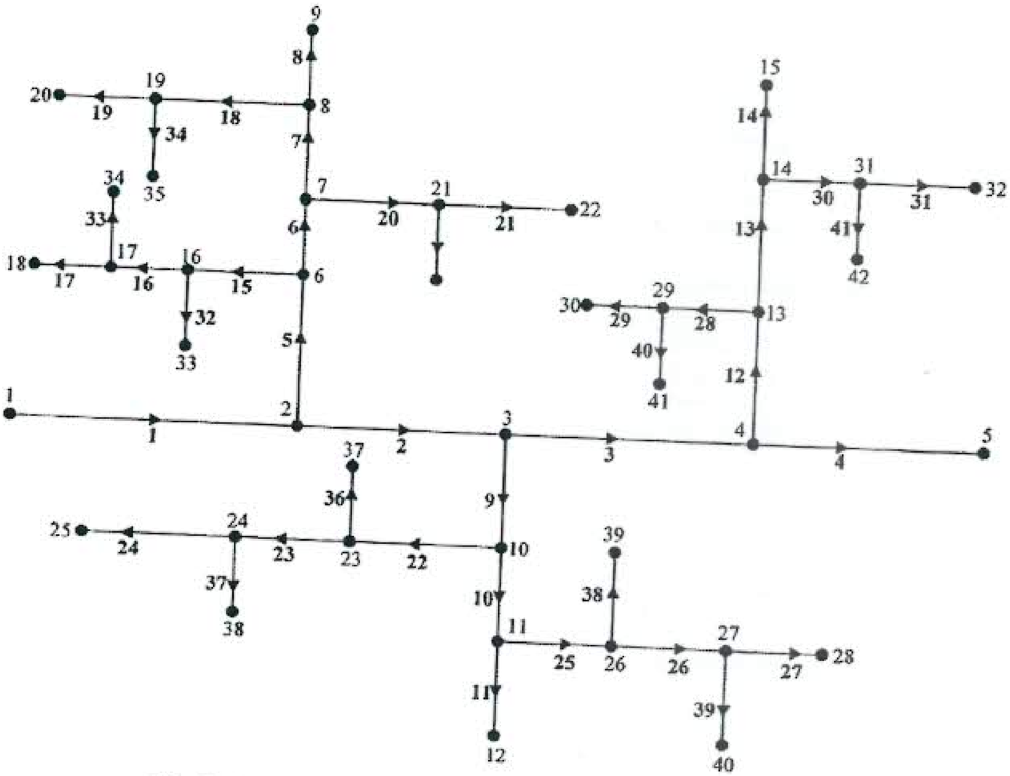


Fig. 2. Branched canal Network-1 [adapted from Naidu et al. (1997)]

J. Irrig. Drain Eng. 2005.131:457-465.

Downloaded from ascelibrary.org by Adlul Islam on 08/27/12. For personal use only. No other uses without permission. Copyright (c) 2012. American Society of Civil Engineers. All rights reserved.

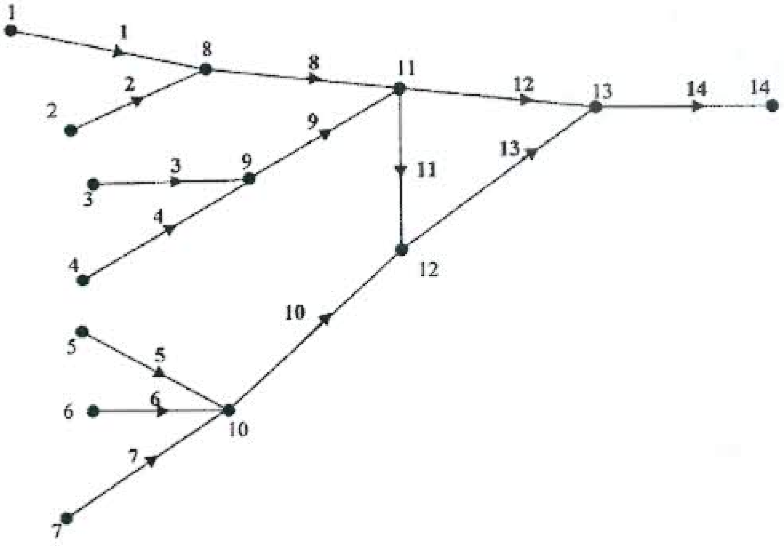


Fig. 3. Looped canal Network-2

$$a_{2j-1,2j+1}=\frac{\partial Fc_j}{\partial Z_{j+1}};\quad a_{2j-1,2j+2}=\frac{\partial Fc_j}{\partial Q_{j+1}}$$

$$a_{2j,2j-1}=\frac{\partial Fm_j}{\partial Z_j};\quad a_{2j,2j}=\frac{\partial Fm_j}{\partial Q_j}$$

$$a_{2j,2j+1}=\frac{\partial Fm_j}{\partial Z_{j+1}};\quad a_{2j,2j+2}=\frac{\partial Fm_j}{\partial Q_{j+1}}$$

Here, the functions Fc and Fm represent continuity and momentum equations, respectively; and vector **D** represents the residuals for continuity and momentum equations. The updated values of Z and Q at $(I+1)$th iteration are computed as follows

$$\mathbf{S}^{I+1}=\mathbf{S}^{I}-\Delta\mathbf{S} \tag{8}$$

For the solution of branched and looped canal networks, the following two algorithms are used:

Algorithm 1: In the solution algorithm developed by Sen and Garg (2002), the Jacobian matrix for $(N-1)$ sections was arranged as shown in Fig. 1 and solved using the Gauss-elimination method to separate out the variables corresponding to end nodes. In the present study, nonzero elements of the Jacobian matrix are stored as $2(N-1)\times 4$ matrix and solved using the following relationships:

$$a'_{l,2}=-\left(\frac{a_{l-1,2}}{a_{l-1,1}}\right)a_{l,1}+a_{l,2} \tag{9}$$

$$a'_{l,3}=-\left(\frac{a_{l-1,3}}{a_{l-1,1}}\right)a_{l,1}+a_{l,3} \tag{10}$$

$$a'_{l,4}=-\left(\frac{a_{l-1,4}}{a_{l-1,1}}\right)a_{l,1}+a_{l,4} \tag{11}$$

$$X'_l=-\left(\frac{X_{l-1}}{a_{l-1,1}}\right)a_{l,1}+X_l \tag{12}$$

$$a'_{k,1}=\frac{a'_{l,3}}{a'_{l,2}}\left[\left(\frac{a_{l-1,2}}{a_{l-1,1}}\right)a_{k,1}-a_{k,2}\right]-\left[\left(\frac{a_{l-1,3}}{a_{l-1,1}}\right)a_{k,1}\right] \tag{13}$$

Table 1. Characteristics of Canal Network-1

Canal number	Length (m)	Bed width (m)	Side slope	Bed slope	Manning's roughness coefficient	Number of reaches
1	2,500	10.00	2.0	0.00013	0.015	20
2	2,000	8.50	2.0	0.00015	0.016	20
3	1,700	7.00	2.0	0.00016	0.017	15
4	1,500	5.00	2.0	0.00017	0.018	15
5	1,500	5.00	2.0	0.00020	0.020	15
6	1,400	4.00	2.0	0.00021	0.020	15
7	1,200	3.00	2.0	0.00022	0.020	15
8	1,000	2.00	2.0	0.00024	0.022	10
9	1,400	3.50	1.0	0.00025	0.022	15
10	1,200	2.70	1.0	0.00022	0.022	15
11	1,000	1.75	2.0	0.00024	0.022	15
12	1,300	2.50	2.0	0.00022	0.022	15
13	1,200	1.50	1.0	0.00025	0.022	15
14	1,000	1.00	2.0	0.00022	0.022	15
15, 18	1,000	1.50	2.0	0.00024	0.022	10
16, 21	1,000	1.00	1.0	0.00025	0.022	10
17, 26	1,000	1.75	2.0	0.00024	0.022	10
19	900	0.90	0.9	0.00025	0.022	10
20, 23	1,100	1.50	2.0	0.00024	0.022	10
22	1,200	1.75	2.0	0.00024	0.022	10
24	1,000	1.00	1.0	0.00025	0.025	10
25	1,200	2.00	2.0	0.00024	0.020	10
27	900	1.50	2.0	0.00024	0.022	10
28	900	1.50	1.0	0.00025	0.022	10
29	800	1.00	1.0	0.00025	0.022	8
30	800	1.25	2.0	0.00024	0.022	8
31	700	0.75	2.0	0.00024	0.022	8
32–41	700	0.50	1.0	0.00050	0.030	5

J. Irrig. Drain Eng. 2005.131:457-465.

Table 2. Characteristics of Canal Network-2

Canal number	Length (m)	Bed width (m)	Side slope	Bed slope	Manning's roughness coefficient	Number of reaches
1, 2, 8, 9	1,500	10	1	0.00027	0.022	15
3, 4	3,000	10	1	0.00047	0.025	30
5, 6, 7, 10	2,000	10	1	0.00030	0.022	20
11	1,200	10	Vertical	0.00033	0.022	12
12	3,600	20	Vertical	0.00025	0.022	36
13	2,000	20	Vertical	0.00025	0.022	20
14	2,500	30	Vertical	0.00016	0.022	25

$$a'_{k,2} = \frac{a'_{l,4}}{a'_{l,2}}\left[\left(\frac{a_{l-1,2}}{a_{l-1,1}}\right)a_{k,1} - a_{k,2}\right] - \left[\left(\frac{a_{l-1,4}}{a_{l-1,1}}\right)a_{k,1}\right] \quad (14)$$

$$X'_k = \frac{X'_l}{a'_{l,2}}\left[\left(\frac{a_{l-1,2}}{a_{l-1,1}}\right)a_{k,1} - a_{k,2}\right] - \left[\left(\frac{X_{l-1}}{a_{l-1,1}}\right)a_{k,1} - X_k\right] \quad (15)$$

where l denotes the row number; X denotes RFm or RFc as the case may be; and $k=2N-3$ and $2N-2$.

Eqs. (9)–(15) are sequentially applied for $l=2,4,6,8,\ldots,2(N-2)$.

Algorithm 2: In this algorithm branch-segment transformation equations (Schaffranek et al. 1981) were developed by combining a series of equations between two consecutive nodes in a branch. By rearranging, Eq. (7) can be written as

$$\Delta \mathbf{S}_{j+1}^{m+1} = \mathbf{F}_j \Delta \mathbf{S}_j^{m+1} + \mathbf{G}_j \quad (16)$$

where

$$\mathbf{F}_j = -\mathbf{C}_j^{-1}\mathbf{B}_j; \quad \mathbf{G}_j = -\mathbf{C}_j^{-1}\mathbf{D}_j$$

Eq. (16) is in a recursive form with respect to ΔS_{j+1} and repeated application of this equation for sections 1 through $(N-1)$ results in the following equation for a branch

$$\Delta \mathbf{S}_N^{m+1} = \mathbf{F}\Delta \mathbf{S}_1^{m+1} + \mathbf{G} \quad (17)$$

where

$$\mathbf{F} = \mathbf{F}_j \mathbf{F}_{j-1} \ldots \mathbf{F}_3 \mathbf{F}_2 \mathbf{F}_1$$

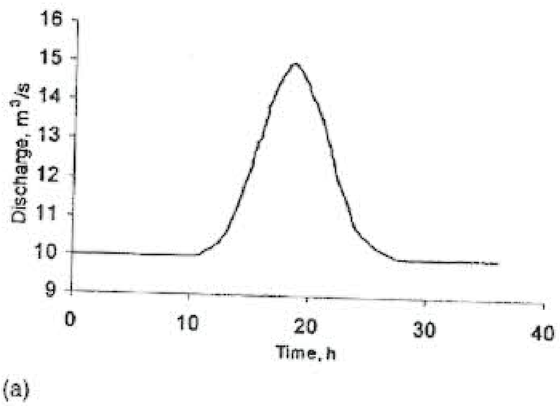
and

$$\mathbf{G} = \mathbf{G}_j + \mathbf{F}_j\{\mathbf{G}_{j-1} + \mathbf{F}_{j-1}[\mathbf{G}_{j-2} + \mathbf{F}_{j-2}(\mathbf{G}_{j-3} + \cdots)\cdots + \mathbf{F}_3(\mathbf{G}_2 + \mathbf{F}_2\mathbf{G}_1)]\cdots\}$$

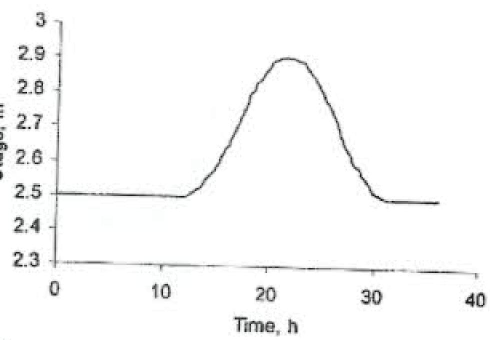
Table 3. Boundary Conditions for Canal Network-1

Node number	Flow depth (m)
5	0.9111
9	1.6559
12	0.9759
15	0.9127
18	1.6021
20	1.8784
22	1.6729
25	1.3622
28	1.4766
30	1.1741
32	1.0749
33	1.4777
34	1.7107
35	2.0070
37	1.2190
38	1.4745
39	1.3719
40	1.6091
41	1.3310
42	1.2535

Eq. (17) now represents two equations for one branch, and hence $2M \times 2M$ matrix for M branches. It is to be noted here that the branch-segment transformation equations are used here with the Newton–Raphson method and hence results are obtained in terms of error variables $(\Delta Z, \Delta Q)$, whereas in the branch model developed by (Schaffranek et al. 1981) the branch-segment transformation equations are used for solving the linearized form of the Saint–Venant equations. Use of the Newton–Raphson method for the solution of nonlinear system of equations in both the algorithms will enable comparison of forward-elimination (Algorithm-1) and branch-segment transformation equations (Algorithm-2) for establishing two nodal equations per branch.



(a)



(b)

Fig. 4. Boundary condition for canal Network-2: (a) discharge hydrographs at nodes 1, 2, 3, 4, 5, 6, 7; (b) stage hydrograph at node 14

J. Irrig. Drain Eng. 2005.131:457-465.

Table 4. Computed Discharge and Depth for Canal Network-1

	Model-1			Naidu et al. (1997)		
Canal number	Discharge (m^3/s)	Up-stream depth (m)	Down-stream depth (m)	Discharge (m^3/s)	Up-stream depth (m)	Down stream depth (m)
1	40.0000	2.1925	1.8665	39.9958	2.1921	1.8664
2	24.6315	1.8665	1.6824	24.6298	1.8664	1.6821
3	16.9734	1.6824	1.5050	16.9722	1.6821	1.5045
4	11.3153	1.5050	0.9111	11.3176	1.5045	0.9111
5	15.3685	1.8665	1.6506	15.3660	1.8664	1.6503
6	10.0926	1.6506	1.5166	10.0897	1.6503	1.5165
7	6.3536	1.5166	1.5334	6.3520	1.5165	1.5333
8	3.8339	1.5334	1.6559	3.8329	1.5333	1.6559
9	7.6582	1.6824	1.3405	7.6576	1.6821	1.3403
10	3.9378	1.3405	1.0036	3.9379	1.3403	1.0036
11	1.9536	1.0036	0.9759	1.9537	1.0036	0.9759
12	5.6580	1.5050	1.4055	5.6547	1.5045	1.4048
13	2.8793	1.4055	0.9848	2.8783	1.4048	0.9845
14	1.4036	0.9848	0.9127	1.4032	0.9845	0.9127
15	5.2759	1.6506	1.6355	5.2762	1.6503	1.6352
16	3.0794	1.6355	1.4233	3.0786	1.6352	1.4232
17	2.3156	1.4233	1.6021	2.3152	1.4232	1.6021
18	2.5197	1.5334	1.7128	2.5191	1.5333	1.7128
19	1.4327	1.7128	1.8784	1.4325	1.7128	1.8784
20	3.7390	1.5166	1.6026	3.7377	1.5165	1.6025
21	2.2070	1.6026	1.6729	2.2065	1.6025	1.6729
22	3.7203	1.3405	1.2660	3.7197	1.3403	1.2658
23	2.5736	1.2660	1.3425	2.5728	1.2658	1.3424
24	1.4614	1.3425	1.3622	1.4612	1.3424	1.3622
25	1.9842	1.0036	1.1098	1.9842	1.0036	1.1098
26	1.4621	1.1098	1.2815	1.4624	1.1098	1.2815
27	1.0850	1.2815	1.4766	1.0853	1.2815	1.4766
28	2.7787	1.4055	1.2696	2.7764	1.4048	1.2693
29	1.7092	1.2696	1.1741	1.7083	1.2693	1.1741
30	1.4757	0.9848	1.0017	1.4751	0.9845	1.0016
31	1.0341	1.0017	1.0749	1.0338	1.0016	1.0749
32	2.1964	1.6355	1.4777	2.1976	1.6352	1.4777
33	0.7639	1.4233	1.7107	0.7634	1.4233	1.7107
34	1.0870	1.7128	2.0070	1.0866	1.7128	2.0070
35	1.5320	1.6026	1.7769	1.5313	1.6025	1.7769
36	1.1467	1.2660	1.2190	1.1469	1.2658	1.2190
37	1.1122	1.3425	1.4745	1.1116	1.3424	1.4745
38	0.5221	1.1098	1.3719	0.5218	1.1098	1.3719
39	0.3771	1.2815	1.6091	0.3770	1.2815	1.6091
40	1.0695	1.2696	1.3310	1.0681	1.2693	1.3310
41	0.4416	1.0017	1.2535	0.4413	1.0016	1.2535

Simultaneous Solution of Global Matrix

For the solution of the global matrix, the Gauss-elimination method with maximum pivot strategy is used. Since this global matrix is sparse, it is also solved using the sparse matrix solution technique. The sparse matrix solution technique requires storage of only nonzero elements along with row and column identifiers, and hence results in reduced storage requirements. The sparse matrix solution technique used in this study employs two partially packed arrays, one for storing nonzero elements and the other for column identification. In this algorithm the row having the minimum number of nonzero elements is selected as a pivotal row, and the column having the largest absolute value within the pivotal row is selected as the pivotal column to avoid instability (Gupta and Tanji 1977).

Thus, in this study, four different models were developed using different combinations of algorithms and global matrix solution techniques. These models are referred to as Model-1 (Algorithm-1 with the Gauss-elimination method), Model-2 (Algorithm-1 with the sparse matrix solution technique), Model-3 (Algorithm-2 with the Gauss-elimination method), and Model-4 (Algorithm-2 with the sparse matrix solution technique).

Table 5. Computed Discharge and Depth for Canal Network-2

	Model-1		
Canal number	Discharge (m^3/s)	Upstream stage (m)	Downstream stage (m)
1	10.0000	3.6938	3.5802
2	10.0000	3.6938	3.5802
3	10.0000	4.2434	3.5802
4	10.0000	4.2434	3.5802
5	10.0000	3.8363	3.7396
6	10.0000	3.8363	3.7396
7	10.0000	3.8363	3.7396
8	20.0000	3.5802	3.2508
9	20.0000	3.5802	3.2508
10	30.0000	3.7396	3.1723
11	10.2563	3.2508	3.1723
12	29.7437	3.2508	2.8850
13	40.2563	3.1723	2.8850
14	70.0000	2.8850	2.5000

Comparison of Algorithms

For comparing the performance of the two algorithms and global matrix solution approaches, computer programs were written in C language. The program consists of five main components/functions: (1) functions for computations of coefficients of vectors $\mathbf{B}_j$, $\mathbf{C}_j$, and $\mathbf{D}_j$; (2) a function for forward-elimination/branch-segment transformation; (3) a function for establishing continuity and energy equations at junction nodes; (4) a function for the Gauss-elimination/sparse matrix solution technique, and (5) a function for back substitution. For comparing the accuracy and computational efficiency of the algorithms described above, two canal networks (Network-1 and Network-2) were selected. Network-1 is a branched canal network having 41 branches (Naidu et al. 1997) and Network-2 is a looped canal network with 14 branches. Network-1 is divergent type and Network-2 is convergent type, representing typical irrigation and drainage systems, respectively. Network-1 and Network-2 are shown in Fig. 2 and Fig. 3 and their general characteristics are given in Table 1 and Table 2, respectively. Results related to the performance of the recursive relationship in the forward-elimination phase of Algorithm-1, comparison between Algorithm-1 and Algorithm-2, and comparison between the sparse matrix solution technique and Gauss-elimination method for the solution of the global matrix are presented in the following section.

J. Irrig. Drain Eng. 2005.131:457-465.

Table 6. Comparison of Solution Time and Global Matrix Size for Steady-Flow Simulation

Model	Algorithm	Global matrix solver[a]	Canal Network-1: Time per iteration (s)	Canal Network-1: Global matrix size (number of variable)	Canal Network-2: Time per iteration (s)	Canal Network-2: Global matrix size (number of variable)
	Algorithm-1[b]	GE	0.4177	26896	—	—
Model-1	Algorithm-1[c]	GE	0.3788	26896	0.0830	3136
Model-2	Algorithm-1[c]	S	0.3204	408	0.0810	140
Model-3	Algorithm-2	GE	0.0829	26896	0.0283	3136
Model-4	Algorithm-2	S	0.0213	367	0.0254	126

[a]GE=Gauss-elimination; S=sparse matrix solution technique.

[b]Forward-elimination done using standard method.

[c]Forward-elimination done using recursive relationship.

Results and Discussion

Steady-State Flow Simulation

The Saint–Venant equations, with time derivates equal to zero, were solved using the finite-difference method for simulating the steady-state flow condition for the chosen branch canal network (Network-1) and the looped canal network (Network-2). The boundary conditions for canal Network-1 and Network-2 are given in Table 3 and Fig. 4, respectively. In the case of Network-1, the inflow at node 1 was 40 m^3/s. For convergence of solution, a tolerance of 0.0001 was used for $|\Delta Q/Q|$ and $|\Delta h/h|$ for all the four models.

The number of iterations (six for Network-1 and four for Network-2) required to converge the solution was the same for all four models since they employed the same solution technique. Furthermore, all four models gave the same values of computed discharge and depth at the upstream and downstream ends, and hence results for only Model-1 are presented here. Computed discharge and depths (upstream and downstream) for Network-1 are presented in Table 4 and these values are almost identical to those given by Naidu et al. (1997). The maximum variation in discharge and upstream and downstream depths was 0.13% (branch 40), 0.05% (branch 28), and 0.05% (branch 12), respectively. This slight variation may be due to the iterative solution technique used by Naidu et al. (1997). The computed discharge and water surface elevation for Network-2 are given in Table 5.

Network-1 was selected to compare the performance of the recursive relationship in the forward-elimination phase of Algorithm-1. As expected, the use of recursive relationships in the forward-elimination phase resulted in faster solution (0.3788 s per iteration) as compared to the standard forward-elimination method (0.4177 s per iteration) (Table 6). It is to be noted here that in both cases only three rows [i.e., when the first row is the pivotal row, second row, and last two rows (Fig. 1)] and four columns are updated.

Though the solution was obtained after the same number of iterations, time taken per iteration was different for all the models (Table 6). For Network-1, Algorithm-2 is about five (Model-3 versus Model-1) to fifteen times (Model-4 versus Model-2) faster than Algorithm-1. In the case of Network-2, Algorithm-2 is about three to four times faster than Algorithm-1.

The models using the sparse matrix solution technique are faster than the models using the Gauss-elimination method. For Network-1, the sparse matrix solution technique was about 1.2 (Model-2 versus Model-1) to four times (Model-4 versus Model-3) faster than the Gauss-elimination method. Similarly, for Network-2, the sparse matrix solution technique took less time as compared to the Gauss-elimination method. Thus, the model using the branch-segment transformation technique for linking end-node variables and the sparse matrix solution technique for solving the global matrix resulted in faster solution as compared to other models.

Unsteady-State Flow Simulation

For unsteady-flow simulation, Network-2 was considered and steady-state discharge and depth were used as initial conditions. The computed stage and discharge hydrographs at a few selected

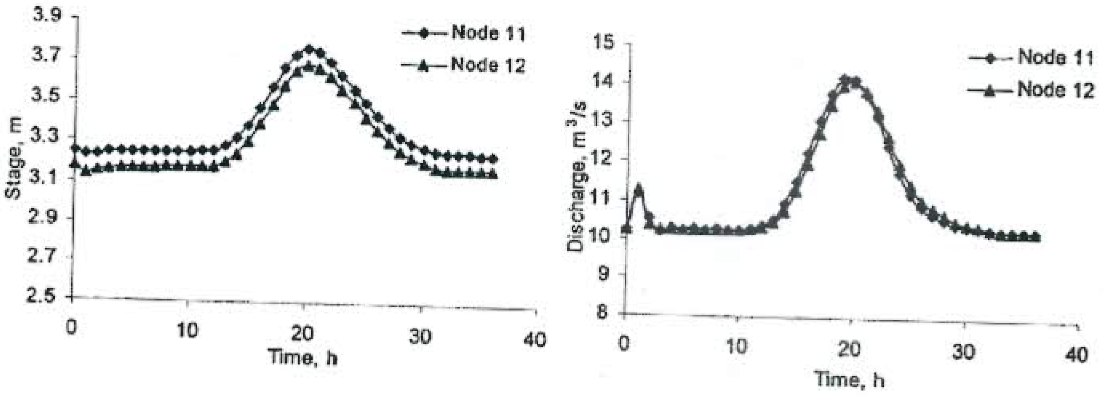


Fig. 5. Computed stage and discharge hydrographs at selected nodes

J. Irrig. Drain Eng. 2005.131:457-465.

Table 7. Comparison of Solution Time and Global Matrix Size for Unsteady-Flow Simulation

Model	Algorithm	Global matrix solver[a]	Total simulation time (s)	Global matrix size (number of variables)
	Algorithm-1[b]	GE	13.446	3136
Model-1	Algorithm-1[c]	GE	12.625	3136
Model-2	Algorithm-1[c]	S	11.375	168
Model-3	Algorithm-2	GE	3.168	3136
Model-4	Algorithm-2	S	3.086	140

[a]GE = Gauss-elimination; S = sparse matrix solution technique.

[b]Forward-elimination done using standard method.

[c]Forward-elimination done using recursive relationship.

nodes are shown in Fig. 5. Similar to the steady-state case, all the models gave the same depth and discharge. Furthermore, the number of iterations required for the convergence of the solution was also the same for all the models. However, the total computation time required by different models was different (Table 7). In this case Algorithm-2 was found to be faster (about 4 times) than Algorithm-1, and the sparse matrix solution technique was faster than the Gauss-elimination method.

Storage Requirements

When the Gauss-elimination method is used for solving the global matrix, the matrix size for both the algorithms remains the same ($4M \times 4M$). In the case of the sparse matrix solution technique, only nonzero elements are stored, and the number of nonzero elements depends on the approaches used for linking end nodes. In general, the branch-segment transformation equations (Algorithm-2) result in six nonzero active variables (three for the continuity and three for the momentum equation) per branch, whereas the forward-elimination method (Algorithm-1) results in eight nonzero active variables per branch. For Network-2, the respective numbers of nonzero active variables for Algorithm-1 and Algorithm-2 are 168 and 140 (Table 7). The number of nonzero elements was further reduced in the case of steady flow due to absence of the time derivatives in the Saint–Venant equations. The respective numbers of nonzero elements for Algorithm-1 and Algorithm-2 are 408 and 367; and 140 and 126, respectively, for Network-1, and Network-2 (Table 6). Thus, while storing only nonzero elements for the solution of the global matrix, the storage requirement is less for Algorithm-2 than Algorithm-1.

Conclusions

For the solution of gradually varied flow in open-channel networks, the Saint–Venant equations were solved using the four-point implicit finite-difference scheme, and the Newton–Raphson method was used for solution of discretized nonlinear equations. A comparison was made between two different approaches for developing branch nodal equations—namely, forward-elimination (Algorithm-1) and branch-segment transformation equations (Algorithm-2). The results showed that the branch-segment transformation equations (Algorithm-2) were faster than the forward-elimination method (Algorithm-1). Also, the algorithm using branch-segment transformation equations required less computer storage than the algorithm using the forward-elimination approach. Comparison between the recursive forward-elimination and the standard forward-elimination approach showed that the recursive-forward-elimination took less time (about 10%) per iteration than the standard forward-elimination method.

The performance of the Gauss-elimination and the sparse matrix solution technique was compared for the solution of the global matrix. The sparse matrix solution technique was found to be faster than the Gauss-elimination method particularly for large and complex networks. Thus, it can be concluded that a model using the branch-segment transformation equations for linking end nodes, and the sparse matrix solution technique for solving the global matrix, results in significant savings in computation time and storage requirements for simulating open-channel flow under both steady and unsteady conditions.

Notation

The following symbols are used in this paper:

A = cross-sectional area;
a = elements of the Jacobian matrix;
$\mathbf{B}_j, \mathbf{C}_j$ = vector representing the Jacobian for jth section;
$\mathbf{D}$ = vector of residuals for continuity and momentum equation;
$\mathbf{F}, \mathbf{G}$ = transformation matrix for a branch;
Fc = function representing continuity equation;
$\mathbf{F}_j, \mathbf{G}_j$ = transformation matrix for jth computational section;
Fm = function representing momentum equation;
g = acceleration due to gravity;
h = depth of water;
M = total number of branches in the network;
N = number of nodes in a branch;
n = Manning's roughness coefficient;
p = total number of nodes in the network;
Q = discharge;
q = lateral inflow;
R = hydraulic radius;
$\mathbf{S}$ = solution vector;
S_f = friction slope;
t = time;
x = longitudinal distance;
Z = water surface elevation;
z = bed elevation; and
β = momentum correction factor.

Subscripts

i = index for incoming channels;
j = index for computational point;
k, l = index for row number;
o = index for outgoing channels; and
m = index for time.

References

Akan, A. O., and Yen, B. C. (1981). "Diffusion-wave flood routing in channel networks." *J. Hydraul. Div., Am. Soc. Civ. Eng.*, 107(6), 719–732.

Amein, M., and Chu, H. (1975). "Implicit numerical modeling of unsteady flows." *J. Hydraul. Div., Am. Soc. Civ. Eng.*, 101(6), 717–731.

Amein, M., and Fang, C. S. (1970). "Implicit flood routing in natural channels." *J. Hydraul. Div., Am. Soc. Civ. Eng.*, 96(12), 2481–2500.

Chaudhry, M. H., and Schulte, A. M. (1986). "Computation of steady-

J. Irrig. Drain Eng. 2005.131:457-465.

state, gradually varied flows in parallel channels." *Can. J. Civ. Eng.*, 13(1), 39–45.
Choi, G. W., and Molinas, A. (1993). "Simultaneous solution algorithm for channel networks modeling." *Water Resour. Res.*, 29(2), 321–328.
Cunge, J. A., Holly, F. M., and Verwey, A. (1980). *Practical aspects of computational river hydraulics*, Pitman, London.
Fread, D. L. (1971). "Discussion of 'Implicit flood routing in natural channels' by M. Amein and C. S. Fang." *J. Hydraul. Div., Am. Soc. Civ. Eng.*, 97(7), 1156–1159.
Fread, D. L. (1973). "Technique for implicit dynamic routing in rivers with tributaries." *Water Resour. Res.*, 9(4), 918–926.
Gupta, S. K., and Tanji, K. K. (1977). "Computer program for solution of large, sparse, unsymmetric systems of linear equations." *Int. J. Numer. Methods Eng.*, 11, 1251–1259.
Joliffe, I. B. (1984). "Computation of dynamic waves in channel networks." *J. Hydraul. Eng.*, 110(10), 1358–1370.
Kao, K. H. (1980). "Improved implicit procedure for multichannel surge computations." *Can. J. Civ. Eng.*, 7, 502–512.
Liu, F., Feyen, J., and Berlamont, J. (1992). "Computation method for regulating unsteady flow in open channels." *J. Irrig. Drain. Eng.*, 118(5), 674–689.
Naidu, B. J., Bhallamudi, S. M., and Narasimhan, S. (1997). "GVF computation in tree-type networks." *J. Hydraul. Eng.*, 123(8), 700–708.
Nguyen, Q. K., and Kawano, H. (1995). "Simultaneous solution for flood routing in channel networks." *J. Hydraul. Eng.*, 121(10), 744–750.
Schaffranek, R. W., Baltzer, R. A., and Goldberg, D. E. (1981). "A model for simulation of flows in singular and interconnected channels." *Techniques of Water-Resources Investigation Report, Chapter C3*, U.S. Geological Survey, Denver.
Schulte, A. M., and Chaudhry, M. H. (1987). "Gradually-varied flow in open-channel networks." *J. Hydraul. Res.*, 25(3), 357–371.
Sen, D. J., and Garg, N. K. (2002). "Efficient algorithm for gradually varied flows in channel networks." *J. Irrig. Drain. Eng.*, 128(6), 351–357.
Swain, E. D., and Chin, D. A. (1990). "Model of flow in regulated open-channel networks." *J. Irrig. Drain. Eng.*, 116(4), 537–556.
Wood, E. F., Harley, B. M., and Perkins, F. E. (1975). "Transient flow routing in channel networks." *Water Resour. Res.*, 11(3), 423–430.

J. Irrig. Drain Eng. 2005.131:457-465.