Hindawi
Advances in Mathematical Physics
Volume 2021, Article ID 6692067, 12 pages
https://doi.org/10.1155/2021/6692067

Research Article

Application of Third-Order Schemes to Improve the Convergence of the Hardy Cross Method in Pipe Network Analysis

Majid Niazkar [1] and Gökçen Eryılmaz Türkkan [2]

[1] Department of Civil and Environmental Engineering, Shiraz University, Shiraz, Iran
[2] Department of Civil Engineering, Bayburt University, Bayburt, Turkey

Correspondence should be addressed to Majid Niazkar; mniazkar@shirazu.ac.ir

Received 27 November 2020; Revised 25 January 2021; Accepted 23 February 2021; Published 10 March 2021

Academic Editor: Leopoldo Greco

In this study, twenty-two new mathematical schemes with third-order of convergence are gathered from the literature and applied to pipe network analysis. The presented methods were classified into one-step, two-step, and three-step schemes based on the number of hypothetical discharges utilized in solving pipe networks. The performances of these new methods and Hardy Cross method were compared by solving a sample pipe network considering four different scenarios (92 cases). The results show that the one-step methods improve the rate of convergence of the Hardy Cross method in 10 out of 24 cases (41%), while this improvement was found to be 39 out of 56 cases (69.64%) and 5 out of 8 cases (62.5%) for the two-step and three-step methods, respectively. This obviously indicates that the modified schemes, particularly the three-step methods, improve the performance of the original loop corrector method by taking lower number of iterations with the compensation of relatively more computational efforts.

1. Introduction

Water distribution networks (WDNs) are a critical infrastructures that affect the daily life of each and everyone who consumes purified water. In essence, management and design of WDNs require the knowledge of flow and pressure fields within WDNs. Such knowledge is exclusively provided through pipe network analysis. This analysis is mainly aimed at solving a set of first-order nonlinear differential algebraic equations with iterative methods. Hardy Cross method, which exploits the Newton-Raphson method, is one of the methods available for this purpose. However, it has several disadvantages to be employed as an adequate hydraulic solvers: (1) It requires an initial guess that conserves continuity equations in the very first iteration in advance of analyzing WDNs [1]. (2) It has a slow rate of convergence, and (3) coding of this method may be harder than other available methods [2]. These shortcomings confine the application of Hardy Cross method to solving WDNs [3].

Despite the challenges mentioned, several studies were conducted in literature about this method in a bit to improve its performance. Lopes [1] implemented the Hardy Cross method for solving pipe networks. Demir et al. [4] proposed a spreadsheet to apply a modified version of the Hardy Cross method for time-dependent simulation of WDNs. Moosavian and Jaefarzadeh [5] suggested a modified Hardy Cross method for analyzing WDNs. Baugh and Liu [6] investigated a general characterization of the Hardy Cross method. Niazkar and Afzali [7] linked MATLAB with MS Excel to solve WDNs under steady-state conditions using three discharge-based methods including the Hardy Cross method. Türkkan et al. [8] studied the implementation of the Hardy Cross method by the programming language C# for teaching calculations of WDNs. Gokyay [9] presented an MS Excel Visual Basic for Applications program for design of closed-loop pipe networks using the Hardy Cross method.

Although the Hardy Cross method is considered one of the oldest approaches for the analysis of pipe networks, it is not being obsoleted because of its advantages [6]. Not only this method is adaptable with handy calculations, but also it can be easily implemented in spreadsheets [4]. Moreover, the Hardy Cross method solves water networks without forming matrix equations [10]. Nonetheless, the convergence of this method is low in comparison to some of matrix-based approaches [11, 12]. Therefore, one of the main challenges of this method is its relatively low rate of convergence.

In this study, new modified variants of the loop corrector method with third-order rate of convergence were applied for solving WDNs. For this purpose, twenty-two schemes were gathered from the literature to improve the rate of convergence of the loop corrector method. The implementations of these new modified versions are elaborated in details for a simple water network using four scenarios. The performance of the original Hardy Cross method was compared with the proposed modified versions. The results demonstrate that some of the recommended modifications improve the rate of convergence of the Hardy Cross method significantly.

2. Method and Materials

2.1. Analysis of Water Distribution Network. Analysis of WDN may be conducted under three different conditions [13]: (1) steady-state condition, (2) extended-period simulation, and (3) transient condition. The rate of change of the state variables of water networks with time is the main difference among these conditions. In the first condition, the temporal variation of state variables is not taken into account whereas abrupt changes of state variables are simulated in the third condition. Unlike these two extreme conditions, the time of network analysis is divided into several intervals in which the steady-state condition governs in each interval in the second condition. Therefore, any modification to the analyzing WDN under the steady-state condition can be simply extended to the extended-period simulation. In this research, the water network analysis is conducted under the steady-state condition.

The governing equations in the steady-state condition comprise (1) water continuity equation and (2) energy equation. The unknown state variables of a pipe network, i.e., pipe flow rate (Q) and nodal hydraulic head (h), are determined by solving these two equations [7]. The iteration-based methods of water distribution networks can be classified as two main approaches [10]: (1) matrix-based and (2) nonmatrix-based methods. The former ones form a matrix equation comprising the governing equations and attempt to solve these equations simultaneously whereas the latter deals with solving equations separately in two steps: First, linear continuity equations are satisfied, and second, nonlinear energy equations are solved iteratively. The former method includes the linear theory, Newton-Raphson, finite element method, and gradient global algorithm while the loop corrector method is the latter approach [14].

The analysis of water network is nothing but solving two governing equations. The first equation balances water for each node (Equation (1)) while the second equation satisfies the energy in the network for each loop (Equation (2)).

$$\sum_{i=1}^{n_1} Q_i = Q_d, \tag{1}$$

$$\sum_{j=1}^{n_2} K Q_j^n = 0, \tag{2}$$

where n_1 is the number of pipes attached to a typical node whose water demand is Q_d, n_2 is the number of pipes in a typical loop, K is the pipe coefficient, and n is the power. The two last parameters depend on the resistance equation adopted. To be more specific, K and n are equal to $8fL/g\pi^2D^5$ and 2, respectively, if the Darcy-Weisback (D-W) equation is utilized as the resistance equation. In this relation, L is pipe length, f is the D-W friction factor, g is the gravity acceleration, π is the pi number, and D denotes the pipe diameter. In calculations conducted in this study, the D-W equation was used, and f was calculated by the explicit Swamee-Jain's relation, i.e., $f = 1.325/[\ln((\varepsilon/3.71D) + (5.74/\mathrm{Re}^{0.9}))]^2$, where ε/D is relative roughness and Re is the Reynolds number [15].

2.2. Original Loop Corrector Method. The Hardy Cross method, as a nonmatrix-based approach, initiates the analysis procedure with a set of pipe flow rates with presumed directions which satisfies the continuity equations. In this regard, the energy balance in each loop is exclusively considered to find a loop correction ($\varDelta Q$) to refine the initial guess (Equation (3)).

$$\varDelta Q_i = -\frac{f(Q_n)}{f'(Q_n)}. \tag{3}$$

In Equation (3), $f(Q_n)$ is the energy equation, which may give either a negative, positive, or zero value; while $f'(Q_n)$ is the first derivative of the energy equation with positive values.

The energy function and its first derivative are evaluated using initial guesses for pipe flow rates in the first iteration. In the next iteration(s), these functions are evaluated using the corrected values of the pipe flow rates which were achieved right at the end of previous iteration. Equation (3) is applied for each loop to find the loop correction in each loop. It should be noted that the energy function, i.e., $f(Q_n)$, is a summation of energy flowing through all pipes in a typical loop.

As the loop correction is determined for each loop, the discharge values of the pipes in that loop are refined by algebraically adding this loop corrector to each pipe flow rate. If the water in a typical pipe is assumed to flow in the direction of the loop correction, the loop correction is added; while in case the pipe flow rate and loop corrector have opposite directions, the loop corrector values are subtracted from the pipe flow rate. Moreover, if a pipe is a part of two adjacent loops, the loop corrector of both loops should be utilized to refine the corresponding flow rate. This refinement procedure of discharge values in pipes will be proceeded until the loop corrector values become negligible which practically means two things: (1) The pipe flow rates will no longer refine if the procedure goes on, and (2) the numerator of Equation (3), i.e., the energy function, approaches enough close to zero which means that the energy equation is acceptably satisfied.

Since Equation (3) is the truncated Taylor series, the Newton-Raphson, the rate of convergence of the original Hardy Cross method is second-order. This rate of convergence is relatively low in comparison to other methods and can be considered a major shortcoming of this method. In order to improve this approach, modified versions of Hardy Cross methods were proposed which has third-order of convergence in this research.

2.3. New Variants of Loop Corrector Methods with Third-Order Convergence. In light of obtaining more precise results in relatively fewer numbers of iterations, the loop corrector method is improved by utilizing higher order of convergence schemes. In this regard, twenty-two third-order convergence Newton-Raphson schemes are selected from the literature. In order to enhance the rate of convergence of the loop corrector method, these schemes are applied as the substitution of Equation (3). In this study, these modified algorithms are classified into three groups: (1) one-step scheme, (2) two-step schemes, and (3) three-step schemes. The first group only uses one discharge, while the second and third groups utilize one and two additional discharges, so-called the hypothetical flow rates, in each loop calculation. The relations of the hypothetical flow rates used in the twenty-two algorithms are presented in the following equations:

$$Q_m = Q_n - \frac{f(Q_n)}{f'(Q_n)}, \tag{4}$$

$$Q_{m1} = Q_n - \frac{f(Q_n)}{2f'(Q_n)}, \tag{5}$$

$$Q_{m2} = Q_n + \frac{f(Q_n)}{f'(Q_n)}, \tag{6}$$

$$Q_{m3} = Q_n + \frac{f(Q_n)}{2f'(Q_n)}, \tag{7}$$

$$Q_r = 0.5Q_m + 0.5Q_n, \tag{8}$$

$$Q_{r1} = \frac{1}{3}Q_n + \frac{2}{3}Q_m, \tag{9}$$

where Q_m, Q_{m1}, Q_{m2}, Q_{m3}, Q_r, and Q_{r1} are hypothetical pipe flow rates exclusively computed to calculate the loop corrector.

For better clarification, the step-by-step process of solving WDNs using one of the modified schemes (Amat et al.'s algorithm) is depicted in Figure 1. As shown, this two-step scheme uses one hypothetical discharge (Q_m^i), and consequently, flow velocity (V_m^i), the Reynolds number (Re_m^i), D-W friction factor (f_m^i), and pipe coefficient (K_m^i) are calculated for Q_m^i. In other words, these four variables are required to be computed for each hypothetical discharge. This requirement inevitably adds more computational efforts to the analysis of WDNs in comparison with the original Hardy Cross method when formulas of the modified schemes contain one or two hypothetical discharges, while this may be compensated with more accuracy achieved by the third-order convergence algorithms.

The modified versions of the loop corrector method are presented in the following:

(1) Schemes with one discharge per pipe for each loop calculation (one-step modified schemes):

(i) Chebyshev's algorithm [16]: This one-step algorithm uses the evaluations of one energy function, i.e., $f(Q_n)$, using the pipe flow rates in the nth iteration (Q_n) and one first and one second derivatives of energy equation, i.e., $f^{'}(Q_n)$, $f^{''}(Q_n)$, respectively. This algorithm is shown below:

$$\Delta Q_i = -\frac{f(Q_n)}{f'(Q_n)} - \frac{f^2(Q_n)f''(Q_n)}{2f'^3(Q_n)} \tag{10}$$

(ii) Halley's algorithm [17]: This algorithm uses the evaluations of one energy function and one first and one second derivatives of energy equation:

$$\Delta Q_i = -\frac{2f(Q_n)f'(Q_n)}{2f'^2(Q_n) - f(Q_n)f''(Q_n)} \tag{11}$$

(iii) Abbasbandy's algorithm [18]: Abbasbandy's algorithm computes one energy function and one first and one second derivatives of energy equation:

$$\Delta Q_i = -\frac{f(Q_n)}{f'(Q_n)} - \frac{f^2(Q_n)f''(Q_n)}{2f'^3(Q_n)} - \frac{f^3(Q_n)f''^2(Q_n)}{2f'^5(Q_n)} \tag{12}$$

(iv) Chun's first algorithm [19]: This one-step algorithm computes the loop corrector using one function and one first-derivative energy equation evaluation:

$$\Delta Q_i = -\frac{f(Q_n)}{f'(Q_n)} - \frac{f^2(Q_n)f'(Q_n)}{2f'^3(Q_n)} \tag{13}$$

(v) Chun and Kim's first algorithm [20]: This algorithm comprises one function, one first and one second-derivative evaluations

$$\Delta Q_i = -\frac{f'(Q_n)f(Q_n)\left[f''(Q_n)f(Q_n) + 2 + 2f'^2(Q_n)\right]}{2f'^2(Q_n)\left[1 + f'^2(Q_n)\right] - f(Q_n)f''(Q_n)} \tag{14}$$

(vi) Chun and Kim's third algorithm [20]: Chun and Kim's third algorithm has one function, one

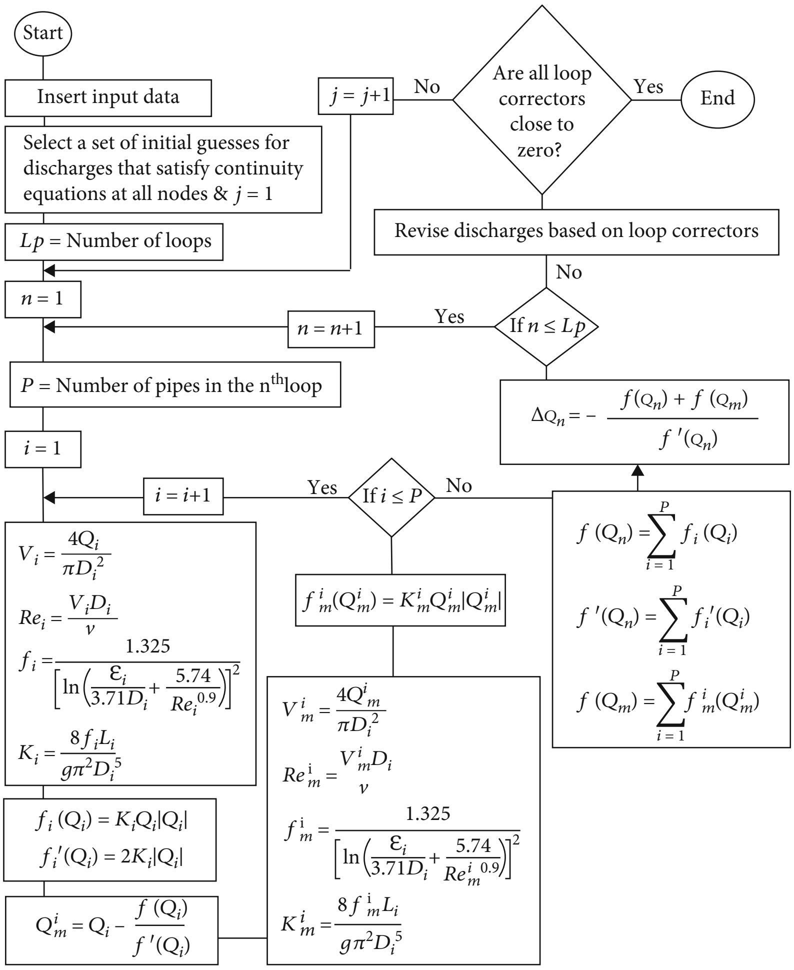
Start
Insert input data
Select a set of initial guesses for discharges that satisfy continuity equations at all nodes & $j = 1$
Lp = Number of loops
$n = 1$
P = Number of pipes in the nthloop
$i = 1$
$i = i+1$
Yes
If $i \leq P$
No
$V_i = \frac{4Q_i}{\pi D_i^2}$
$Re_i = \frac{V_i D_i}{v}$
$f_i = \frac{1.325}{\left[\ln\left(\frac{\varepsilon_i}{3.71D_i} + \frac{5.74}{Re_i^{0.9}}\right)\right]^2}$
$K_i = \frac{8f_iL_i}{g\pi^2 D_i^5}$
$f_i(Q_i) = K_iQ_i|Q_i|$
$f_i'(Q_i) = 2K_i|Q_i|$
$Q_m^i = Q_i - \frac{f(Q_i)}{f'(Q_i)}$
$f_m^i(Q_m^i) = K_m^iQ_m^i|Q_m^i|$
$V_m^i = \frac{4Q_m^i}{\pi D_i^2}$
$Re_m^i = \frac{V_m^iD_i}{v}$
$f_m^i = \frac{1.325}{\left[\ln\left(\frac{\varepsilon_i}{3.71D_i} + \frac{5.74}{Re_m^{i\,0.9}}\right)\right]^2}$
$K_m^i = \frac{8f_m^iL_i}{g\pi^2D_i^5}$
$f(Q_n) = \sum_{i=1}^{P} f_i(Q_i)$
$f'(Q_n) = \sum_{i=1}^{P} f_i'(Q_i)$
$f(Q_m) = \sum_{i=1}^{P} f_m^i(Q_m^i)$
$\Delta Q_n = -\frac{f(Q_n) + f(Q_m)}{f'(Q_n)}$
If $n \leq Lp$
Yes
No
$n = n+1$
Revise discharges based on loop correctors
Are all loop correctors close to zero?
No
Yes
$j = j+1$
End

Figure 1: Flowchart of Amat et al.'s algorithm for solving a typical WDN.

first-derivative, and one second-derivative function evaluations

$$\Delta Q_i = -\frac{f^2(Q_n)f''(Q_n) + 2f(Q_n)f'^2(Q_n)}{2f'^3(Q_n)}. \quad (15)$$

(2) Schemes with two-discharge per pipe for each loop calculation (two-step modified schemes):

(i) Weerakoon and Fernando's algorithm [21]: This two-step algorithm utilizes the evaluations of one energy function and two first derivatives of energy equation:

$$\Delta Q_i = -\frac{2f(Q_n)}{f'(Q_m) + f'(Q_n)} \quad (16)$$

(ii) Frontini and Sormani's algorithm [22]: This algorithm evaluates one energy function and two first derivatives of energy equation in two steps:

$$\Delta Q_i = -\frac{f(Q_n)}{f'(Q_{m1})} \quad (17)$$

(iii) Amat et al.'s algorithm [23]: This two-step algorithm calculates the loop corrector by evaluating two energy functions and one first derivative of energy equation:

$$\Delta Q_i = -\frac{f(Q_n) + f(Q_m)}{f'(Q_n)} \quad (18)$$

(iv) Özban's algorithm [24]: This algorithm comprises the evaluation of one energy function and two first-derivatives of energy equation in two steps:

$$\Delta Q_i = -\frac{f(Q_n)}{2f'(Q_n)} - \frac{f(Q_n)}{2f'(Q_m)} \quad (19)$$

(v) Kou et al.'s algorithm [25]: This algorithm has two steps for determining the loop corrector

Table 1: Summary of the third-order convergence algorithms.

Algorithms	Year	No. of steps	No. of function evaluation	No. of first-derivative evaluation	No. of second-derivative evaluation
(a) One-step modified schemes					
Chebyshev's algorithm	1964	1	1	1	1
Halley's algorithm	1995	1	1	1	1
Abbasbandy's algorithm	2003	1	1	1	0
Chun's first algorithm	2006	1	1	1	0
Chun and Kim's first algorithm	2010	1	1	1	1
Chun and Kim's third algorithm	2010	1	1	1	1
(b) Two-step modified schemes					
Weerakoon and Fernando's algorithm	2000	2	1	2	0
Frontini and Sormani's third algorithm	2003	2	1	2	0
Amat et al.'s algorithm	2003	2	2	1	0
Özban's algorithm	2004	2	1	2	0
Kou et al.'s algorithm	2006	2	2	1	0
Chun's second algorithm	2006	2	2	2	0
Chun's third algorithm	2006	2	2	1	0
Jisheng et al.'s first algorithm	2006	2	1	2	0
Jisheng et al.' second algorithm	2006	2	1	2	0
Darvishi and Barati's algorithm	2007	2	2	1	0
Zhou's algorithm	2007	2	1	1	0
Ham et al.'s algorithm	2008	2	2	1	0
Chun and Kim's second algorithm	2010	2	1	2	0
Chun and Kim's fourth algorithm	2010	2	1	2	0
(c) Three-step modified schemes					
Jisheng et al.'s third algorithm	2006	2	1	2	0
Zavalani's algorithm	2014	2	1	2	0

for each loop including the evaluation of two functions and one first derivatives of energy equation:

$$\Delta Q_i = -\frac{f(Q_{m2}) - f(Q_n)}{f'(Q_n)} \tag{20}$$

(vi) Chun's second algorithm [19]: Chun's second algorithm utilizes two functions and two first-derivative energy equation evaluations in two steps:

$$\Delta Q_i = -\frac{f(Q_n)}{f'(Q_n)} - \frac{2f(Q_m)}{f'(Q_n) + f'(Q_m)} \tag{21}$$

(vii) Chun's third algorithm [19]: This algorithm calculates loop corrector by evaluating two energy function and one first-derivative energy equation in tow steps:

$$\Delta Q_i = -\frac{f(Q_n)}{f'(Q_n)} - \frac{f(Q_n)f(Q_m)}{[f(Q_n) - f(Q_m)]f'(Q_n)} \tag{22}$$

(viii) Jisheng et al.'s first algorithm [26]: Jisheng et al.'s first algorithm first calculates a hypothetical flow rate and computes loop corrector afterwards. It uses one function and two first-derivative energy equation evaluations:

$$\Delta Q_i = -f'(Q_{m3})\frac{f(Q_n)}{f'^2(Q_n)} \tag{23}$$

(ix) Jisheng et al.'s second algorithm [26]: This two-step algorithm calculates Q_m similar to Jisheng et al.'s first algorithm while it computes loop corrector using different formulations as shown:

$$\Delta Q_i = -\frac{f(Q_n)}{2f'(Q_n)} - f'(Q_{m3})\frac{f(Q_n)}{2f'^2(Q_n)} \tag{24}$$

(x) Darvishi and Barati's algorithm [27]: This two-step algorithm has two function evaluations and one first-derivative evaluation:

$$\Delta Q_i = -\frac{f(Q_n)}{f'(Q_n)} - \frac{f(Q_m)}{f'(Q_n)} \tag{25}$$

(xi) Zhou's algorithm [28]: This α-power mean Newton's algorithm computes the loop corrector in two steps while it uses a free parameter (α) as the exponent of the first derivative of energy equations. In this study, the magnitude of this free parameter is assumed to be -1:

$$\Delta Q_i = -\frac{2^{1/\alpha} f(Q_n)}{\operatorname{sign}\left(f'(Q_n)\right)\left[f'^{\alpha}(Q_n) + f'^{\alpha}(Q_m)\right]^{1/\alpha}} \tag{26}$$

where sign is the sign function

(xii) Ham et al.'s algorithm [29]: Ham et al.'s algorithm is a two-step algorithm having two function evaluations and one first derivate of energy equation

$$\Delta Q_i = -\frac{f(Q_n)}{f'(Q_n)} - \frac{f^2(Q_n) + f'^2(Q_n) + f(Q_n)f'(Q_n)}{f^2(Q_n) + f'^2(Q_n)}\frac{f(Q_m)}{f'(Q_n)} \tag{27}$$

(xiii) Chun and Kim's second algorithm [20]: This algorithm calculates loop corrector in tow steps as below:

$$\Delta Q_i = -\frac{f(Q_n)\left[2 + 3f'^2(Q_n) - f'(Q_n)f'(Q_m)\right]}{f'(Q_n) + 2f'^3(Q_n) + f'(Q_m)} \tag{28}$$

(xiv) Chun and Kim's fourth algorithm [20]: The calculation of loop corrector using this algorithm requires the evaluations of one function and two first derivatives:

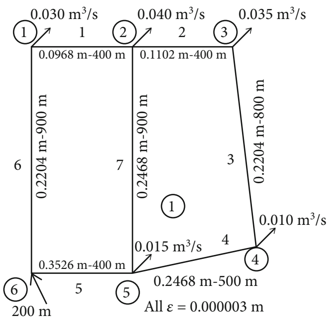

Figure 2: Layout, pipe properties, and nodal demands of the sample pipe network.

$$\Delta Q_i = -\frac{3f(Q_n)}{2f'(Q_n)} + \frac{f(Q_n)f'(Q_m)}{2f'^2(Q_n)}. \tag{29}$$

(3) Schemes with three-discharge per pipe for each loop calculation (three-step modified schemes):

(i) Jisheng et al.'s third algorithm [26]: This algorithm computes the loop corrector using one function evaluation and two first-derivative energy equation evaluations:

$$\Delta Q_i = -\frac{f(Q_n)}{2f'(Q_n)} - \frac{f(Q_n)}{4f'(Q_r) - 2f'(Q_n)} \tag{30}$$

(ii) Zavalani's algorithm [30]: Zavalani's algorithm is a three-step algorithm presented in the following:

$$\Delta Q_i = -\frac{4f(Q_n)}{f'(Q_n) + 3f'(Q_{r1})}. \tag{31}$$

2.4. Comparison of the Modified Algorithms with Third-Order of Convergence. The twenty-two schemes presented in the previous section are compared in Table 1 in respect to the number of functions and derivatives required to be evaluated. According to this table, most of these methods are two-step algorithms, which use two discharges for each pipe in each loop calculation. Obviously, when an additional discharge for each pipe is required in a modified scheme, it demands to compute V, Re, f, and K for the new discharge values, as shown in Figure 1. This certainly requires much more calculations, which bring about a trade-off between accuracy and computational efforts.

3. Results and Discussion

A sample pipe network was solved using twenty-three methods, which include twenty-two modified algorithms and the original Hardy Cross method, to compare

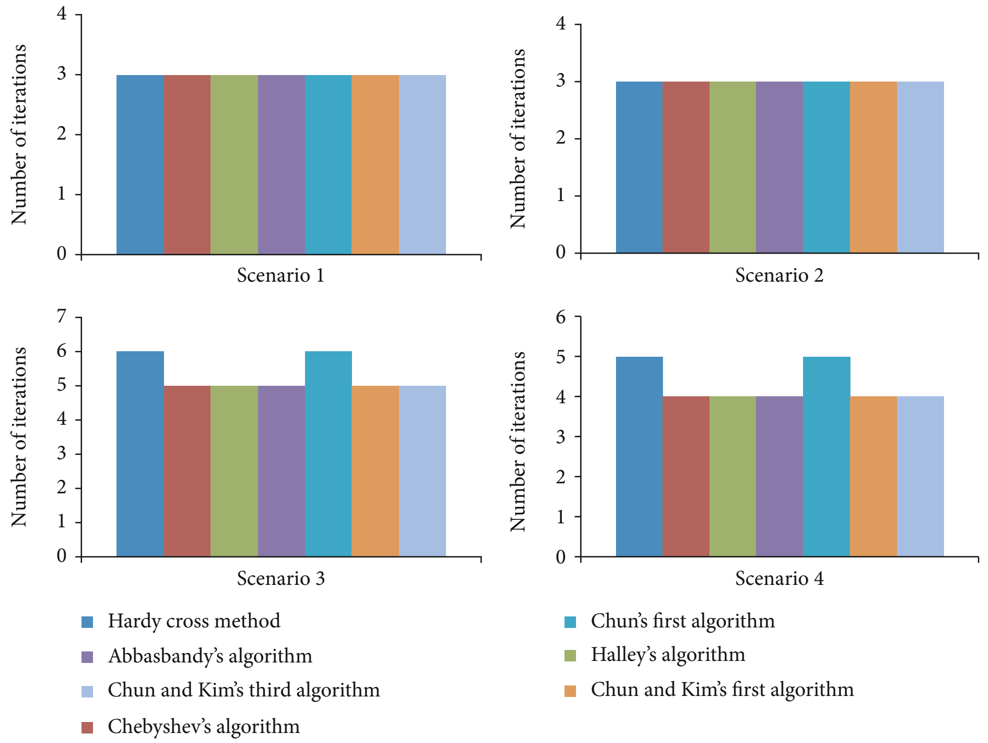

FIGURE 3: Comparison of number of iterations for solving the sample pipe network using one-step loop corrector methods (from left to right: Hardy Cross method, Chebyshev's algorithm, Halley's algorithm, Abbasbandy's algorithm, Chun's first algorithm, Chun and Kim's first algorithm, and Chun and Kim's third algorithm).

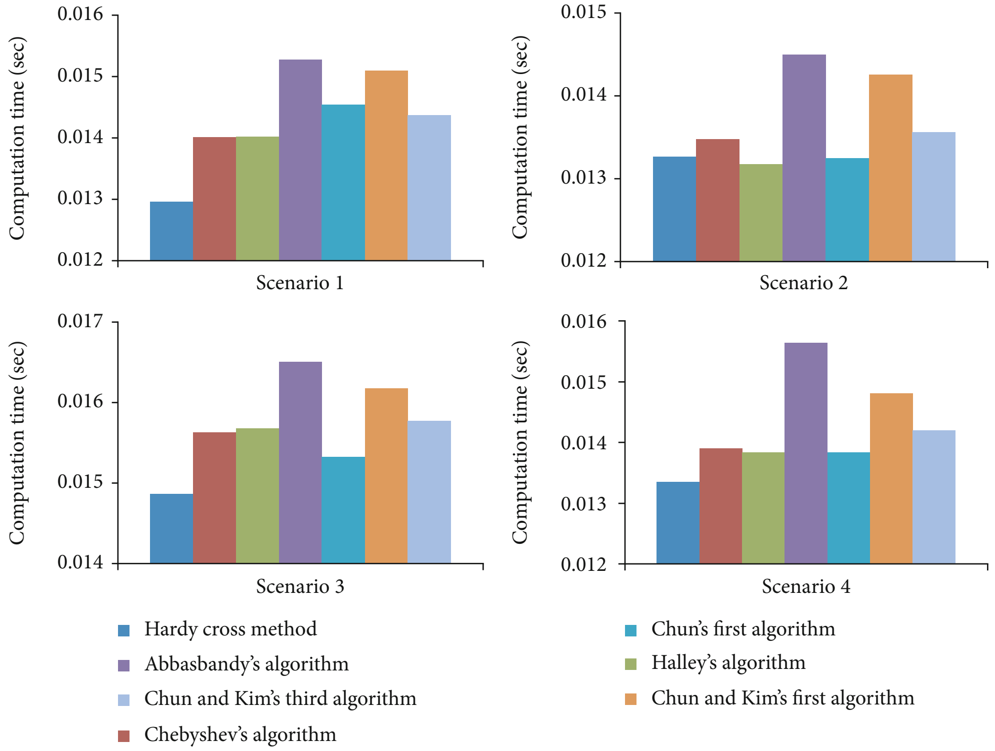

FIGURE 4: Comparison of computation time for solving the sample pipe network using one-step loop corrector methods (from left to right: Hardy Cross method, Chebyshev's algorithm, Halley's algorithm, Abbasbandy's algorithm, Chun's first algorithm, Chun and Kim's first algorithm, and Chun and Kim's third algorithm).

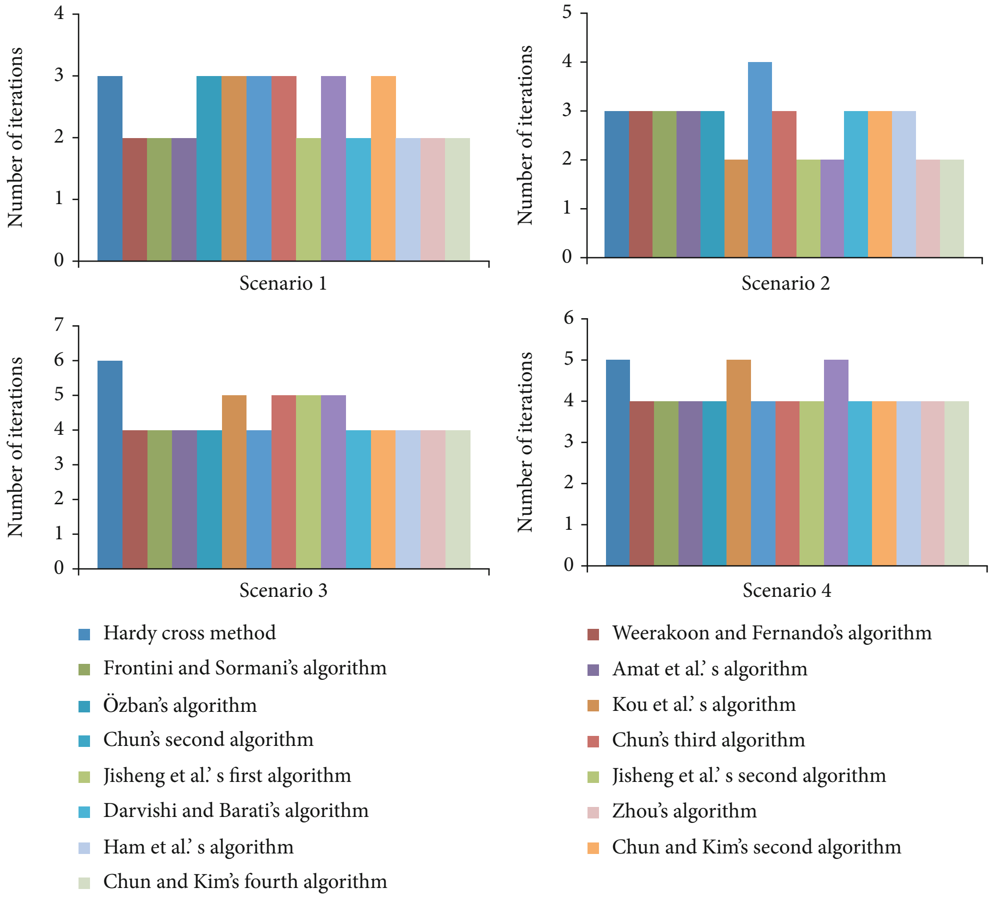

Figure 5: Comparison of number of iterations for solving the sample pipe network using two-step loop corrector methods (from left to right: Hardy Cross method, Weerakoon and Fernando's algorithm, Frontini and Sormani's algorithm, Amat et al.'s algorithm, Özban's algorithm, Kou et al.'s algorithm, Chun's second algorithm, Chun's third algorithm, Jisheng et al.'s first algorithm, Jisheng et al.'s second algorithm, Darvishi and Barati's algorithm, Zhou's algorithm, Ham et al.'s algorithm, Chun and Kim's second algorithm, and Chun and Kim's fourth algorithm).

performances of different schemes. This network, which is adopted from the literature [2, 14], consists of two loops, seven pipes, and six nodes. Figure 2 depicts the layout, pipe characteristics, water demands at nodal points, and the reservoir hydraulic head connected to this pipe network.

Each and every algorithm was coded in MATLAB and MS Excel to solve the sample pipe networks for four different scenarios (92 cases overall), while applications of these programs were previously recommended for implementing numerical modeling [31–33] and in particular for pipe network analysis [2, 7, 9]. The detail of the four different scenarios is presented in the following:

(1) The first scenario: The initial guess is close but not too close to the final solution. Each discharge is kept constant during the computation of each loop, and discharges are modified only after the end of the calculation of each iteration

(2) The second scenario: The initial guess is close but not too close to the final solution. Each discharge can be modified during the computation of each loop and each iteration

(3) The third scenario: The initial guess is far from the final solution. Each discharge is constant during the computation of each loop, and discharges are modified only after the end of the calculation of each iteration

(4) The fourth scenario: The initial guess is far from the final solution. Each discharge can be modified during the computation of each loop and each iteration.

The detail results achieved for solving the sample pipe network using 23 methods for four scenarios are presented in the following:

3.1. Results of the One-Step Modified Schemes. The performance of the six on-step methods in solving WDNs is compared with that of the original Hardy Cross method in Figure 3. As shown, these methods solved the sample pipe network in three iterations for the first and second scenarios, whereas the third and fourth scenarios demand four, five, or six iterations of running these methods. This implies that the initial guess has an impact on the number of iterations required for solving a typical WDN using loop corrector

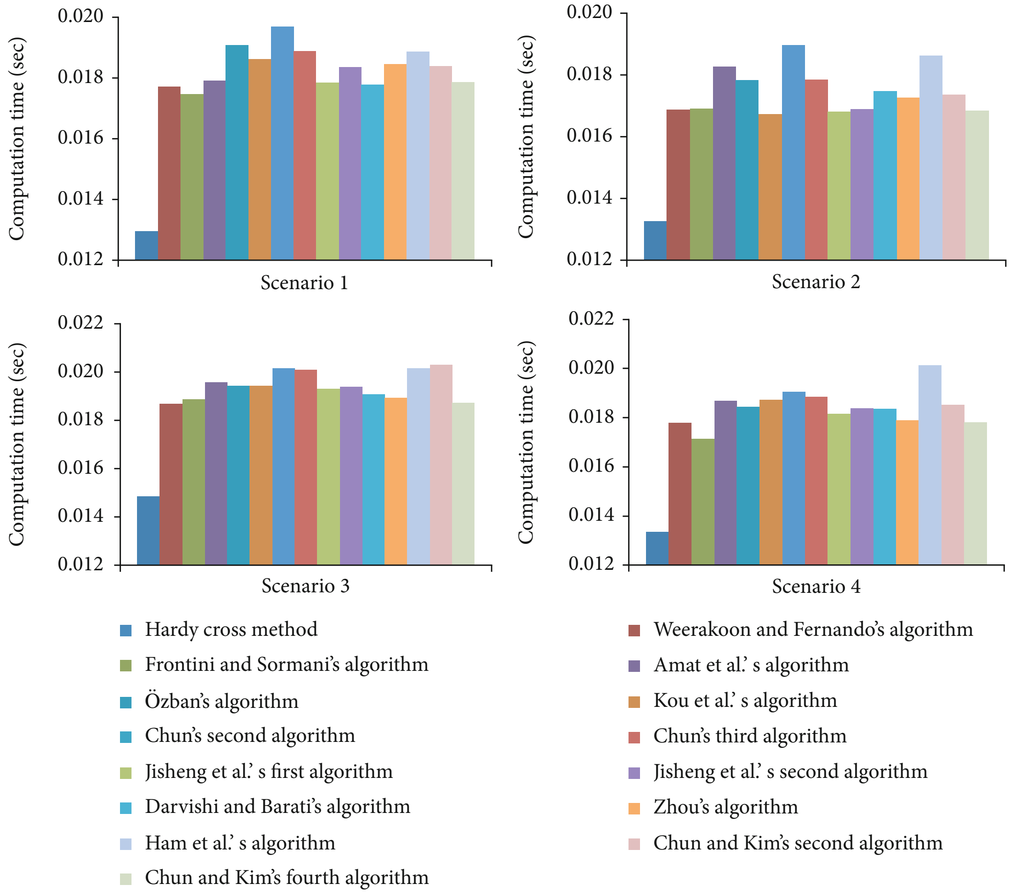

Figure 6: Comparison of computation time for solving the sample pipe network using two-step loop corrector methods (from left to right: Hardy Cross method, Weerakoon and Fernando's algorithm, Frontini and Sormani's algorithm, Amat et al.'s algorithm, Özban's algorithm, Kou et al.'s algorithm, Chun's second algorithm, Chun's third algorithm, Jisheng et al.'s first algorithm, Jisheng et al.'s second algorithm, Darvishi and Barati's algorithm, Zhou's algorithm, Ham et al.'s algorithm, Chun and Kim's second algorithm, and Chun and Kim's fourth algorithm).

methods. To be more specific, the closeness of initial guess and the final solution would reduce the number of iterations in the process of analyzing WDNs. Furthermore, the performances of all schemes seem to be similar in the first two scenarios. However, based on results of the third and fourth scenarios shown in Figure 3, Chun's first algorithm was performed as the original Hardy Cross method, while the rest of one-step schemes improve the performance of the latter. Finally, Figure 3 shows that the one-step methods reach the final solutions using a lower number of iterations in 10 out of 24 cases (41%).

Figure 4 compares the computation time for solving the sample pipe network using six one-step loop corrector methods and the original Hardy Cross method. As shown, the lowest computation time in each scenario belongs to the Hardy Cross method, whereas the highest computation time was achieved by Abbasbandy's algorithm for four scenarios. By considering both Figures 3 and 4, it is observed that Chebyshev's algorithm and Halley's algorithm performed better than other one-step methods based on the iteration number and computation time. Although Chun's first algorithm gained relatively low computation time in Figure 4, it utilized the same iteration numbers as the Hardy Cross method (6 and 5 iterations) for solving the third and fourth scenarios in Figure 3, respectively.

3.2. Results of the Two-Step Modified Schemes. The numbers of iterations required by the two-step schemes and the original Hardy Cross method to solve the sample WDN are depicted in Figure 4 for the four scenarios. As shown, a few two-step algorithms improved solving the sample network by satisfying the stopping criterion by lower number of iterations. To be more specific, eight algorithms took 2 iterations for the first scenario, whereas the original Hardy Cross method and other two-step loop corrector methods required three iterations. For the second scenario, Kou et al.'s algorithm, Jisheng et al.'s first and second algorithms, and Chun and Kim's second and fourth algorithms outperformed others by solving the sample network in two iterations, while Chun's second algorithm and the others solved it in four and three iterations, respectively. Furthermore, the results of the third scenario illustrated in Figure 5 demonstrate that all two-step methods decreased the number of iterations achieved by the Hardy Cross method. Likewise, the two-step algorithms (except Kou et al.'s algorithm and Jisheng et al.'s second algorithm) reduced the number of iterations

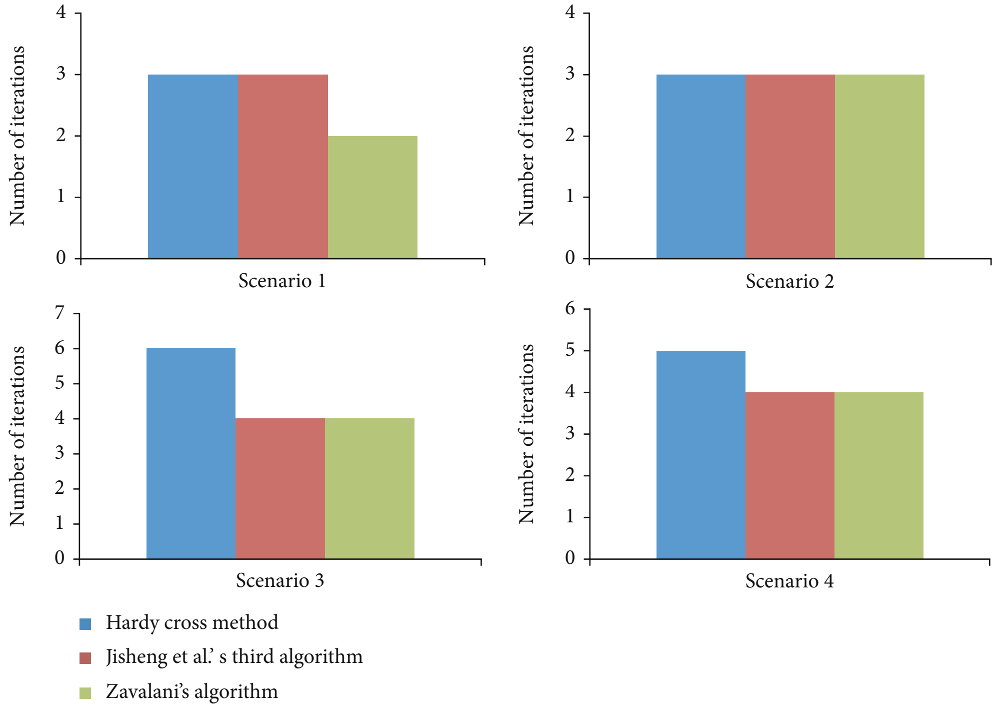

FIGURE 7: Comparison of number of iterations for solving the sample pipe network using three-step loop corrector methods (from left to right: Hardy Cross method, Jisheng et al.'s third algorithm, and Zavalani's algorithm).

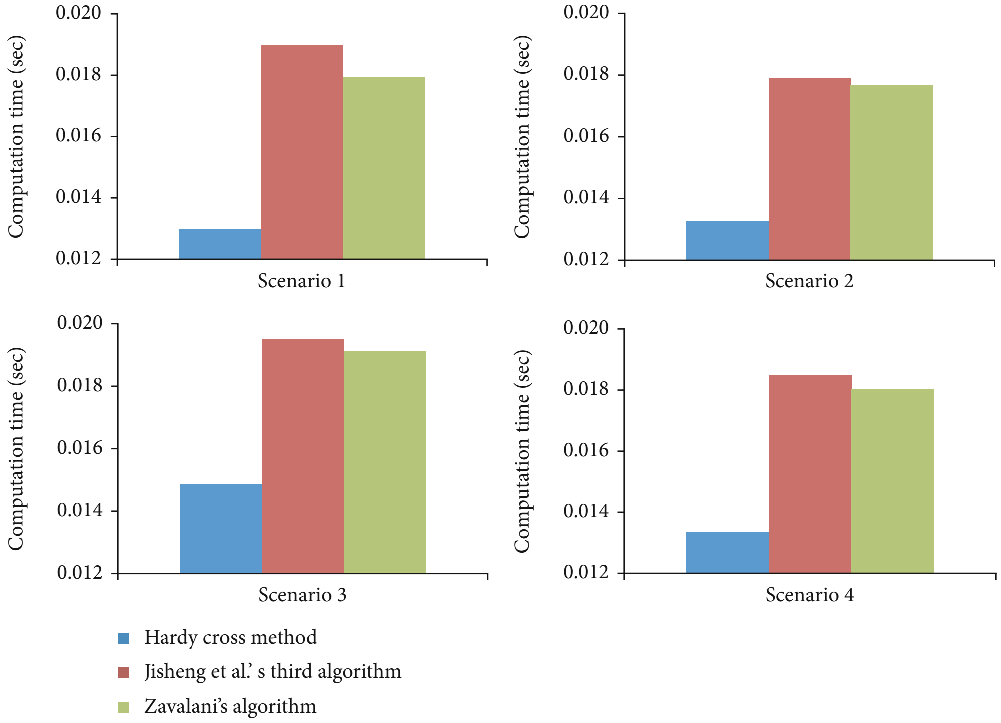

FIGURE 8: Comparison of computation time for solving the sample pipe network using three-step loop corrector methods (from left to right: Hardy Cross method, Jisheng et al.'s third algorithm, and Zavalani's algorithm).

obtained by the Hardy Cross method from 5 to 4. According to Figure 5, Chun and Kim's second and fourth algorithms yielded to the lowest number of iterations by taking into account the four scenarios. In summary, Figure 5 demonstrates that the two-step methods improved the rate of convergence of the Hardy Cross method in 39 out of 56 cases (69.64%), while Chun's second algorithm performed worse than the latter in the second scenario.

The computation time required for solving the sample pipe network using the two-step algorithms and the Hardy Cross method is compared in Figure 6. As shown, the lowest computation time was obtained by the Hardy Cross method, whereas Chun's second algorithm (for the first three scenarios) and Ham et al.'s algorithm (for the fourth scenario) achieved the highest computation time. Based on Figures 5 and 6, Chun and Kim's fourth algorithm outperformed other two-step methods because it relatively solved the four-scenario of the sample pipe network by taking the lowest iteration numbers and computation time.

3.3. Results of the Three-Step Modified Schemes. Figure 7 compares the performances of the three-step modified schemes and the original Hardy Cross method. As shown, Zavalani's algorithm solved the first scenario in two iterations, while Jisheng et al.'s third algorithm and the original Hardy Cross method needed three iterations for this purpose. Additionally, these three methods used the same iteration numbers in the second scenario. Also, Figure 7 obviously indicates that both three-step methods improve the convergence rate of the original Hardy Cross method in the third and fourth scenarios. To be more precise, they solved the third scenario in four iterations, whereas the original Hardy Cross method took six iterations. Moreover, Zavalani's algorithm and Jisheng et al.'s third algorithm analyzed the fourth scenario in four iterations, while five iterations were required for the original Hardy Cross method to reach the final solutions of the fourth scenario. Based on Figure 7, the three-step methods took a lower number of iterations in 5 out of 8 cases (62.5%). Therefore, many cases considered demonstrate that the proposed loop corrector methods improve the rate of convergence of the Hardy Cross method.

Figure 8 shows the computation time of the Hardy Cross method and the three-step schemes for solving the sample WDN. According to Figure 8, the lowest and highest computation times were obtained by the Hardy Cross method and Jisheng et al.'s third algorithm, respectively, while Zavalani's algorithm required a slightly lower computation time than the latter for all four scenarios. Based on Figures 7 and 8, Zavalani's algorithm overall performed better than Jisheng et al.'s third algorithm in the analysis of the sample WDN.

4. Conclusions

In this paper, twenty-two new loop corrector methods with third-order of convergence are proposed. Despite the original loop corrector method, i.e., Hardy Cross method, these new methods theoretically have one higher order of convergence. A sample water network was analyzed using four scenarios (92 cases overall) for comparing the performance of these new versions of loop corrector methods with the original Hardy Cross method. The results indicate that the closeness of initial guesses to the final solutions was found to be an important factor in number of iterations required to solve a typical WDS. However, considering different scenarios reveals that one-step, two-step, and three-step schemes improve the rate of convergence of Hardy Cross method by 41%, 69.64%, and 62.5%, respectively. Additionally, one of the two-step methods, Chun's third algorithm, was found to solve the sample network in more number of iterations that the Hardy Cross method for one out of four scenarios. Moreover, based on comparing the iteration number and computation time of the four scenarios of the sample pipe network, Chebyshev's algorithm and Halley's algorithm performed better than other one-step methods, while Chun and Kim's fourth algorithm and Zavalani's algorithm outperformed other two-step and three-step methods, respectively. Finally, the improvement obtained by applying the modified schemes demonstrates that the rate of convergence of the loop corrector method can be considerably increased by adopting these schemes.

Data Availability

The pipe network data used in this study are reported in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. M. G. Lopes, "Implementation of the Hardy-Cross method for the solution of piping networks," *Computer Applications in Engineering Education*, vol. 12, no. 2, pp. 117–125, 2004.

[2] M. Niazkar and S. H. Afzali, "Analysis of water distribution networks using MATLAB and Excel spreadsheet: h-based methods," *Computer Applications in Engineering Education*, vol. 25, no. 1, pp. 129–141, 2017.

[3] D. H. Huddleston, V. J. Alarcon, and W. Chen, "Water distribution network analysis using excel," *Journal of Hydraulic Engineering*, vol. 130, no. 10, pp. 1033–1035, 2004.

[4] S. Demir, K. Yetilmezsoy, and N. Manav, "Development of a modified Hardy-Cross algorithm for time-dependent simulations of water distribution networks," *Fresenius Environmental Bulletin*, vol. 17, no. 8a, pp. 1045–1053, 2008.

[5] N. Moosavian and M. R. Jaefarzadeh, "Hydraulic analysis of water supply networks using a modified Hardy Cross method," *International Journal of Engineering*, vol. 27, no. 9, pp. 1331–1338, 2014.

[6] J. Baugh and S. Liu, "A general characterization of the Hardy Cross method as sequential and multiprocess algorithms," *Structure*, vol. 6, pp. 170–181, 2016.

[7] M. Niazkar and S. H. Afzali, "Analysis of water distribution networks using MATLAB and Excel spreadsheet: Q-based methods," *Computer Applications in Engineering Education*, vol. 25, no. 2, pp. 277–289, 2017.

[8] Y. A. Türkkan, T. G. Eryılmaz, and H. Yılmaz, "A visual application for teaching pipe flow optimization in engineering

curricula," *Computer Applications in Engineering Education*, vol. 28, no. 1, pp. 154–159, 2020.

[9] O. Gokyay, "An easy MS Excel software to use for water distribution system design: a real case distribution network design solution," *Journal of Applied Water Engineering and Research*, vol. 8, no. 4, pp. 290–297, 2020.

[10] E. Luvizotto Jr., M. C. Cavichia, P. Vatavuk, and J. G. P. Andrade, "Nonmatrix gradient method for the simulation of water distribution networks," *Journal of Water Resources Planning and Management*, vol. 139, no. 4, pp. 433–439, 2013.

[11] E. Creaco and M. Franchini, "Comparison of Newton-Raphson global and loop algorithms for water distribution network resolution," *Journal of Hydraulic Engineering*, vol. 140, no. 3, pp. 313–321, 2014.

[12] E. Todini and L. A. Rossman, "Unified framework for deriving simultaneous equation algorithms for water distribution networks," *Journal of Hydraulic Engineering*, vol. 139, no. 5, pp. 511–526, 2013.

[13] T. M. Walski, D. V. Chase, D. A. Savic, W. Grayman, S. Beckwith, and E. Koelle, *Advanced Water Distribution Modeling and Management*, Haestad Press, Waterbury, CT, USA, 2003.

[14] M. Niazkar and N. Talebbeydokhti, "Comparison of explicit relations for calculating Colebrook friction factor in pipe network analysis using h-based methods," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 44, no. 1, pp. 231–249, 2020.

[15] M. Niazkar, "Revisiting the Estimation of Colebrook Friction Factor: A Comparison between Artificial Intelligence Models and C-W based Explicit Equations," *KSCE Journal of Civil Engineering*, vol. 23, no. 10, pp. 4311–4326, 2019.

[16] J. F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea Publishing Company, 1964.

[17] T. R. Scavo and J. B. Thoo, "On the geometry of Halley's method," *The American Mathematical Monthly*, vol. 102, no. 5, pp. 417–426, 1995.

[18] S. Abbasbandy, "Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method," *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 887–893, 2003.

[19] C. Chun, "Construction of Newton-like iteration methods for solving nonlinear equations," *Numerische Mathematik*, vol. 104, no. 3, pp. 297–315, 2006.

[20] C. Chun and Y. I. Kim, "Several new third-order iterative methods for solving nonlinear equations," *Acta Applicandae Mathematicae*, vol. 109, no. 3, pp. 1053–1063, 2010.

[21] S. Weerakoon and T. G. I. Fernando, "A variant of Newton's method with accelerated third-order convergence," *Applied Mathematics Letters*, vol. 13, no. 8, pp. 87–93, 2000.

[22] M. Frontini and E. Sormani, "Some variant of Newton's method with third-order convergence," *Applied Mathematics and Computation*, vol. 140, no. 2-3, pp. 419–426, 2003.

[23] S. Amat, S. Busquier, and J. M. Gutiérrez, "Geometric constructions of iterative functions to solve nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 157, no. 1, pp. 197–205, 2003.

[24] A. Y. Özban, "Some new variants of Newton's method," *Applied Mathematics Letters*, vol. 17, no. 6, pp. 677–682, 2004.

[25] J. Kou, Y. Li, and X. Wang, "A modification of Newton method with third-order convergence," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 1106–1111, 2006.

[26] K. Jisheng, L. Yitian, and W. Xiuhua, "A uniparametric Chebyshev-type method free from second derivatives," *Applied Mathematics and Computation*, vol. 179, no. 1, pp. 296–300, 2006.

[27] M. T. Darvishi and A. Barati, "A third-order Newton-type method to solve systems of nonlinear equations," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 630–635, 2007.

[28] X. Zhou, "A class of Newton's methods with third-order convergence," *Applied Mathematics Letters*, vol. 20, no. 9, pp. 1026–1030, 2007.

[29] Y. Ham, C. Chun, and S. G. Lee, "Some higher-order modifications of Newton's method for solving nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 222, no. 2, pp. 477–486, 2008.

[30] G. Zavalani, "A modification of Newton method with third-order convergence," *Applied Mathematics and Computation*, vol. 2, no. 4, pp. 98–101, 2014.

[31] F. Motaman, G. R. Rakhshandehroo, M. R. Hashemi, and M. Niazkar, "Application of RBF-DQ method to time-dependent analysis of unsaturated seepage," *Transport in Porous Media*, vol. 125, no. 3, pp. 543–564, 2018.

[32] M. Niazkar, "An excel VBA-based educational module for bed roughness predictors," *Computer Applications in Engineering Education*, 2020.

[33] M. Niazkar and S. H. Afzali, "Streamline performance of excel in stepwise implementation of numerical solutions," *Computer Applications in Engineering Education*, vol. 24, no. 4, pp. 555–566, 2016.