Control and Optimization in Applied Mathematics (COAM)
Vol. 7, No. 2, Summer-Autumn 2022 (53-75), ©2016 Payame Noor University, Iran

DOI. 10.30473/coam.2022.62968.1195 (Cited this article)

Research Article

Solving Nonlinear Hydraulic Equations of Water Distribution Networks by Using a Trust-Region Method

Mahdi Ahmadnia[1], **Reza Ghanbari**[1,*], **Khatere Ghorbani-Moghadam**[2]

[1]Faculty of Mathematical Sciences, Department of Applied Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran.
[2]Mosaheb Institute of Mathematics, Kharazmi University, Tehran, Iran.

Received: February 10, 2022; **Accepted:** November 27, 2022 .

Abstract. In a water distribution network, in order to analyze and determine its parameters such as head and flow rate, we have to solve nonlinear hydraulic equations in each component of the network. Contrary to most of the water distribution network simulation software, solving these equations by using the gradient method, we propose a trust-region method to solve them, as the trust-region method is newer than the gradient method and has well worked in mathematical problems. To prove the effectiveness of our method, we made a comparison between our proposed method and the well-known gradient method. The results show that the trust-region method is convergent in all instances, but the gradient method diverges when the dimension of nonlinear hydraulic equations of water distribution networks increases. In addition, our results convince us that the solution obtained from the trust-region method is more accurate compared to the gradient method. Thus, using the trust-region method in solving the network equations can lead to a better hydraulic analysis of the network.

Keywords. Water distribution network, Hydraulic equations, Nonlinear equation, Trust-Region method.

MSC. 90C34; 90C40.

* Corresponding author
ahmadnia.mahdi74@gmail.com, rghanbari@um.ac.ir, kh.ghorbani@khu.ac.ir
http://mathco.journals.pnu.ac.ir

1 Introduction

Head and flow regulation in water distribution networks (WDNs) is a significant concern for water utilities. Effective head and flow control throughout pipe networks are essential to ensure rational sufficient service levels to customers for daily fluctuating demand patterns. Simulation models are applied to estimate the distribution of pipe flow rates and residual nodal heads (pressures) within pipe networks, in which these hydraulic parameters have to be computed for different loading and operating conditions [3]. For finding head, flow, and also hydraulic analysis some nonlinear equations have to be solved [15, 52].

There are many methods done to solve the nonlinear equations in WDNs [40]. An iterative method for solving these equations was first proposed by Cross [12] (This method was also used for solving gas network equations; see [8]). Cross proposed an approach that is used to solve the equations Q, ΔQ, and H related to WDNs. The number of calculations required for convergence in the Cross method depends on the convergence criterion (accuracy of the solution), the initial solution, the flow rate of the pipes, and also the resistance of the pipes (R).

Cross's method [12] solves only one equation at a time with some assumptions, such as ignoring the effect of adjacent loops. Martin and Peter [34] proposed the Newton–Raphson method for solving nonlinear WDNs equations, which solves all equations simultaneously. This method is used to solve flow and node equations. However, their approach works better in solving node equations than flow equations. Shamir and Howard [45] and Zarghamee [56] used the Newton–Raphson method for networks with valves and pumps. They investigated the convergence conditions of the Newton–Raphson method and the possibility of insolvable problems. In each iteration of the Newton–Raphson method, in order to determine the correction of the pipe discharge values, a linear equation system must be solved. This linear system is formed by the Jacobian matrix in each iteration. Liu [30] modified the Jacobian matrix to a diagonal matrix. He demonstrated that by using the diagonal matrix, the speed in solving the linear equation is accelerated fast in each iteration. Moosavian and Jaefarzadeh [36] illustrated that the approach proposed by Liu has two disadvantages. One of them is the lack of convergence in large WDNs, and the other is high fluctuations to achieve convergence. So, they suggested that some network pipes must be temporarily removed during the analysis process. They also halved the amount of correction per repetition to reduce fluctuations, but they increased the number of repetitions until the final solution was reached; see [36]. It is also important how to choose the initial solution in this method. If the wrong initial solution is chosen, then the Newton–Raphson method diverges. There are more suggestions for improving the convergence of the Newton–Raphson method. Most of these suggestions correct the pipe flow rates per repetition (see [4, 14, 28, 29, 39, 43, 44, 46, 47]). Based on the Broyden method [9], Tabesh [49] provided a relation for finding the correction rate of flow rate in each iteration. Tanyimboh et al. [50] proposed a line search method in order to accelerate the convergence.

Wood and Charles [53] used the linear theory method to solve flow equations. They showed that their proposed method is too fast and independent of the initial solution.

Also, Collins and Johnson [10] and Isaacs and Mills [22] used the linear theory for solving node equations, which is usually better for solving flow equations than node equations. Each iteration of this method assumes a value for the flow rate of the pipes based on the flow rate obtained from the previous iterations. Using this hypothetical value, a linear equation system that is approximately equivalent to a network equation system is solved. During the convergence process of this method, fluctuations occur. These fluctuations reduce the convergence rate of the linear theory method. Due to these fluctuations, Nielsen [37] proposed using a combination of linear theory and the Newton–Raphson method, so the initial solution of the Newton–Raphson method is produced by linear theory. For the purpose of increasing the speed of convergence, Bhave [6] provided a method for determining the hypothetical flow rates of each iteration. He suggested using the hypothetical mean flow rate of mth and the flow rate obtained in the mth repeat as the hypothetical flow rate of $m+1$.

The most common method currently used in many network simulation software, such as EPANET, is the gradient method. Todini and Pilati [51] introduced this method for WDNs. The gradient method finds the solution of the equations in each iteration solving a linear equation system. Although more equations need to be solved in this method, Todini and Pilati [51] have shown that this method is very computationally robust. Powell [43] solved this algorithm by using Lagrange coefficients for optimization problems with equality constraints. The gradient method is somewhat independent of the initial solution, but if the initial solution is close to the final solution, then the degree of convergence of this method is at least two [5]. See other works for solving WDNs in [2, 7, 13, 18, 20, 21, 26, 24, 32, 33]. See the summary of literature review in Table 1.

The trust region is a newer method compared to the gradient method. So far, the trust-region method for solving equations of WDNs has not been investigated and we use the trust-region method to solve WDN equations. The results show that the trust-region method is more accurate in solving WDN equations compared to the gradient method. So, the trust-region method can provide a better hydraulic analysis of WDN. Here, we use the trust-region method for solving hydraulic equations in a WDN.

The rest of our work is organized as follows. In Section 2, we provide the necessary definitions. We propose a trust-region method for solving flow equations in Section 3. In Section 4, we implement our proposed algorithm on several test problems and compare them with the gradient method. Finally, conclusion will be resented in Section 5.

2 Preliminaries

WDNs are designed in different types. Serial networks, branching networks, looped networks, and composite networks are among the types of WDNs. Here, we give some basic definitions of WDNs.

Definition 1 (Node)**.** [48] The point of intersection of several pipes, as well as the starting and endpoints of each pipe, is called a node.

Table 1: Survey of literature review

Names of authors	**Type of problem**	**Solving methods**
Cross's method [12]	Nonlinear WDNs equations	Iterative method
Martin and Peter [34]	Nonlinear WDNs equations	Newton–Raphson method
Shamir and Howard [45]	Nonlinear WDNs equations	Newton–Raphson method
Zarghamee [56]	Nonlinear WDNs equations (Networks with valves and pumps)	Newton–Raphson method
Liu [30]	Nonlinear WDNs equations	Linearization approach
Moosavian and Jaefarzadeh [36]	Nonlinear WDNs equations	Modified Liu's method
Tabesh [49]	Nonlinear WDNs equations	Iterative algorithm based on the Broyden method
Tanyimboh et al. [50]	Nonlinear WDNs equations	Iterative algorithm based on line search method
Wood and Charles [53]	Nonlinear WDNs equations	Linear theory method to solve flow equations
Collins and Johnson [10]	Nonlinear WDNs equations	Linear theory for solving node equations
Isaacs and Mills [22]	Nonlinear WDNs equations	Linear theory for solving node equations
Nielsen [37]	Nonlinear WDNs equations	Hybrid algorithm based on the linear theory and the Newton–Raphson method
Bhave [6]	Nonlinear WDNs equations	Iterative algorithm for determining the hypothetical flow rates of each iteration
Todini and Pilati [51]	Nonlinear WDNs equations	Gradient method
Powell [43]	Nonlinear WDNs equations	Iterative algorithm by using Lagrange coefficients

Definition 2 (Consumption node)**.** [48] The nodes from which water is removed are called consumption nodes.

Definition 3 (Source node)**.** [48] The nodes through which water enters the network are called source nodes.

Definition 4 (Loop)**.** [48] The closed environment that creates several interconnected pipes is called a loop.

Each WDN consists of different components, such as storage tanks, pipes, valves, pumps, and so on; see [48]. Each of these components can affect the head and flow.

The characteristics of each of these components are described by the head-flow relationship in that component. For example, the head-flow relationship for network pipes is obtained from the following relationship (see [48]):

$$h_{ij} = H_i - H_j = R_{ij} Q_{ij} |Q_{ij}|^{n-1}, \tag{1}$$

where h_{ij} is the decrease of energy (h_{ij} shows the amount of head loss in the pipe ij). Moreover, Q_{ij} and R_{ij} represent the current passing through the pipe ij and the resistance constant of the pipe ij, respectively. Also, H_i and H_j are equal to the head in nodes i and j, respectively. The direction of water flow in the pipes of a distribution network is always from more heads to fewer heads. When water is transferred from one node to another through a pipe, its hydraulic energy is reduced due to friction [27]. This shows the decrease in energy in relation to (1), which is denoted by the symbol h_{ij}. In other words, h_{ij} is equal to the amount of head loss in the pipe ij. Moreover, n is the power of water flow, and to calculate it, we use the Hazen–Williams method. In the Hazen–Williams method, the value of n is considered equal to $1,852$, and the value of R_{ij} is obtained from the following equation:

$$R_{ij} = \frac{\alpha . L_{ij}}{C_{HW_{ij}}^{1.852} . D_{ij}^{4.87}}, \tag{2}$$

where α is equal to 10.675 (in the metric system) and L_{ij}, $C_{HW_{ij}}$, and D_{ij} indicate the length of the pipe ij (in meters), the Hazen–Williams coefficient of the pipe ij, and the diameter of the pipe ij (in meters), respectively. The Hazen–Williams coefficient depends on the characteristics of the pipe, such as the material, age, and so on, and it is determined and announced by the pipe's manufacturers [23].

If the head is specified at both ends of a pipe, then the value of Q_{ij} is calculated based on (1) as follows:

$$Q_{ij} = \left(\frac{|H_i - H_j|}{R_{ij}} \right)^{(\frac{1}{n})} sgn(H_i - H_j), \tag{3}$$

where sgn is the sign function. For the hydraulic analysis, as well as determining the parameters of a WDN, the nonlinear equations in the network components must be solved. These equations are obtained according to the network components and using the two basic laws of continuity and energy survival.

Definition 5 (Continuity rule)**.** [48] According to this rule, the sum of the input current values in each node is equal to the sum of the output current values from that node. In other words,

$$\left(\sum_{ij \in IJ_j} Q_{ij} \right)_{in} - \left(\sum_{ij \in IJ_j} Q_{ij} \right)_{out} = q_j, \qquad \text{for all } j = 1, \ldots, NJ, \tag{4}$$

where q_j is the input or output flow rate of node j, IJ_j represents all the pipes connected to node j, and NJ is the number of nodes in the network. Some of the equations obtained from this relation may not be independent. In fact, the number of independent equations obtained from (4) is equal to the number of consumption nodes. Therefore, (4) is not used for source nodes.

Definition 6 (Energy rule)**.** [48] This rule is used for all loops in the distribution network to write equations. According to this law, the total head loss inside a loop is considered equal to zero, that is,

$$\sum_{ij\in IJ_L} h_{ij} = \sum_{ij\in IJ_L} R_{ij}Q_{ij}|Q_{ij}|^{n-1} = 0, \qquad \text{for all } L = 1,\dots,NL, \tag{5}$$

where, IJ_L represents all loop pipes L and NL is equal to the number of network loops. If the direction for water flow in the pipe is clockwise, then the head loss sign for that pipe in (5) will be positive; otherwise, it will be negative (In the equations written in terms of flow, the mentioned symbol is included in the coefficient of resistance of the pipe).

Note: The direction of water flow in the pipes cannot be determined before solving the network equations. For this reason, first, the direction of flow in the pipes is hypothetically determined, and the network equations are written based on it. After solving the equations, whenever the flow of a pipe is obtained as a negative number, it means that the direction of flow in this pipe is assumed to be the opposite. However, the amount of current is correct, and there is no need to solve the equations again.

According to Definitions 5 and 6, different equations can be written for the analysis of WDNs, including Flow, node, ring equations, ΔH equations, and head-flow equations. Here, we solve equations of the flow system. The number of flow equations is equal to the number of pipes in the network, and the unknown of these equations is the flow rate of the pipes. Flow equations are written by using both the laws of continuity and energy. In the flow equations, the equations derived from the law of continuity are all linear, and the equations derived from the law of energy are all nonlinear. By combining (4) and (5), the system of flow equations is obtained.

Now, using the laws of continuity and energy, we write the flow equations for the network in Figure 1 as follows:

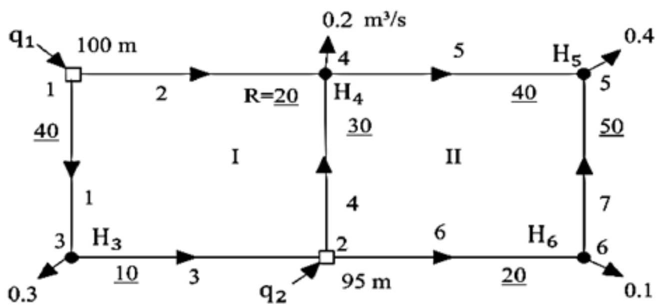

Figure 1: A small sample of WDN [51].

Node 3 : $Q_1 - Q_3 - 0.3 = 0,$

$$
\begin{aligned}
&\text{Node } 4: Q_2 + Q_4 - Q_5 - 0.2 = 0,\\
&\text{Node } 5: Q_5 + Q_7 - 0.4 = 0,\\
&\text{Node } 6: Q_6 - Q_{65} - 0.1 = 0,\\
&\text{Loop I}: 20Q_2|Q_2|^{n-1} - 30Q_4|Q_4|^{n-1} - 10Q_3|Q_3|^{n-1} - 40Q_1|Q_1|^{n-1} = 0,\\
&\text{Loop II}: 40Q_5|Q_5|^{n-1} - 50Q_7|Q_7|^{n-1} - 20Q_6|Q_6|^{n-1} + 30Q_4|Q_4|^{n-1} = 0.
\end{aligned} \tag{6}
$$

By solving a system of equations (6), the flow rate of all network pipes is obtained. Methods such as Cross, Newton-Raphson, linear theory, and Gradient have been used to solve the equations of water supply networks. In this paper, the trust-region method is used to solve these equations. We will explain more about this in the following section.

3 Trust-Region Method for Solving the System of Flow Equations

As a kind of numerical method for solving nonlinear optimization problems, the trust-region method has been widely studied in recent decades [55]. The trust-region method was first used to solve unconstrained optimization problems by Powell [43], of which the distance between the iteration points in the current iteration cycle and the cycle before should be limited. In this method, by applying the Taylor-series expansion, a quadratic model is used to approximate the objective function. It can be thought that there is a neighborhood around the current iteration point within which we trust the surrogate model. Such a neighborhood is called a trust region. The size of the trust region is tuned by the performance of the algorithm in the previous search; see [38]. See other works [11, 16, 35, 54].

The equations of WDNs are nonlinear. So far, different methods have been proposed to solve nonlinear equations; see [1, 17, 19, 25, 41]. One of the desirable and efficient methods to solve the system of nonlinear equations is to use the existing methods in optimization. In other words, solving the system of equations is equivalent to solving an optimization problem. Therefore, instead of solving the system of equations, the equivalent optimization problem can be solved. To use these methods, the system of equations must first be turned into an optimization problem. Then, using the trust-region method solves it.

In this section, first, we explain how to convert a system of equations into an optimization problem and then propose a trust-region algorithm to solve it.

3.1 Converting the System of Equations into an Optimization Problem

In this section, we explain how to convert a system of equations to an optimization problem. Consider the system of equations $r(x) = 0$, in which $r : \mathbb{R}^n \to \mathbb{R}^n$ is a vector function as follows:

$$r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_n(x) \end{bmatrix}, \tag{7}$$

where $r_i : \mathbb{R}^n \to \mathbb{R}$, for $i = 1, 2, \dots, n$, is the ith equation of the system $r(x) = 0$. The following lemmas can be obtained easily.

Lemma 1. The vector x^* is the solution to the system of equations $r(x) = 0$ if and only if $\|r(x^*)\| = 0$.

Lemma 2. The vector x^* is the solution to the system $r(x) = 0$ if and only if the optimal solution to the problem $\min \|r(x)\|$ be zero.

Proof. Suppose that x^* is the solution to $r(x) = 0$, based on Lemma 1, $\|r(x^*)\| = 0$. On the other hand, $\|r(x)\|$ is always positive, so, the solution to the problem $\min \|r(x)\|$ is zero.
The converse of the theorem follows similarly. □

Therefore, instead of solving the system of nonlinear equations, the equivalent optimization problem can be solved. If the solution to the optimization problem is zero, then the system of equations will have a solution, and the solution is equal to the solution to the optimization problem. Indeed if the solution to the optimization problem is a nonzero number, then the system of equivalent equations will not have a solution. The system of equations may also have more than one solution, in this case, the equivalent optimization problem will have several optimal solutions. Hence, the following optimization problem can be considered equivalent to solving the system $r(x) = 0$:

$$\min \ \frac{1}{2}\|r(x)\|^2. \tag{8}$$

Now, considering the system of flow equations as $r(Q) = 0$, according to (8), we have

$$\min_{Q \in \mathbb{R}^n} \frac{1}{2}\|r(Q)\|^2. \tag{9}$$

Problem (9) can also be written according to the flow equations as follows:

$$\min_{Q \in \mathbb{R}^n} \frac{1}{2}\left[\sum_{j=1}^{NJ}\left(\sum_{ij \in IJ_j} Q_{ij} + q_j\right)^2 + \sum_{L=1}^{NL}\left(\sum_{ij \in IJ_L} R_{ij} Q_{ij}^n\right)^2\right], \tag{10}$$

where R_{ij} is the constant of ijth pipe resistance constant and IJ_j and IJ_L, respectively, represent the pipes connected to node j and the pipes in the L ring. Also, NJ and NL are equal to the number of nodes and network pipes, respectively. Therefore, model (10) is an unconstrained and nonlinear optimization problem. This model can be solved by different optimization methods. Here, we use the trust-region method to solve (10). In what follows, we describe this method.

3.2 Trust-Region Method

Algorithms solving optimization problems usually start from an initial solution and then improve the current point in each iteration. For this reason, these algorithms are also known as iterative algorithms. The strategy of transition from one iteration to another is a factor that distinguishes iterative algorithms. In general, iterative methods are divided into two main categories [38]:

- Line search methods,
- Trust-region methods.

In the line search methods, first, the direction of movement is determined and then the length of the step is decided. Indeed in the methods of the trust-region, first, the length of the movement step is determined and then the direction is decided according to the selected step length. We explain the trust-region method for solving the optimization problem (9) assuming $f(Q) = \frac{1}{2}\|r(Q)\|^2$ as follows.

Suppose that Q_k is the flow rate in the kth iteration. In iterative methods for solving optimization problems, Q_{k+1} is updated as follows:

$$Q_{k+1} = Q_k + p_k, \tag{11}$$

where the vector p_k is selected in such a way that the maximum improvement for the problem objective function occurs. In each iteration of the trust-region method, for finding p_k, by using the Taylor series, an approximation of the objective function is obtained as follows:

$$f(Q_k + p) = f_k + g_k^T p + \frac{1}{2} p^T \nabla^2 f(Q_k + tp) p, \tag{12}$$

where $f_k = f(Q_k)$, $g_k = \nabla f(Q_k)$, and t is a number in the range $(0, 1)$. The Jacobian matrix (J) can be used as a suitable approximation instead of ∇f and $\nabla^2 f$ [38]. By replacing $\nabla f = J_k^T r_k$ and $\nabla^2 f = J_k^T J_k$, a suitable approximation is obtained in each iteration of the objective function of the problem. Also,

$$m_k(p) = f_k + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p = \frac{1}{2}\|r_k + J_k p\|^2, \tag{13}$$

where $m_k(p)$ is an approximation of the function $f(Q)$ around the point Q_k. The difference between the approximate function $m_k(p)$ and the function $f(Q_k + p)$ is equal to $O(\|p\|^2)$. If the value of p is small, then the difference between the two functions will be small. Therefore, in each iteration of the trust-region method, to find the suitable direction, the following optimization problem must be solved:

$$\min_{p \in \mathbb{R}^n} \; m_k(p) = f_k + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p \quad \text{s.t. } \|p\| \leq \Delta_k, \tag{14}$$

where Δ_k is the radius of the trust-region in the kth iteration. The value of Δ_k should be chosen such that $m_k(p)$ and $f(Q)$ are approximately equal in this region. Moreover,

p_k is the solution of (14). In fact, p_k is the direction in which the most reduction for $m_k(p)$ occurs. As mentioned, if p is small, then the value of $m_k(p)$ and $f(Q)$ will be close to each other. If small Δ_k with $\|p\| \leq \Delta_k$ is selected, then the two functions $m_k(p)$ and $f(Q)$ in this region behave similarly. Hence, for $f(Q)$, the largest possible reduction occurs by moving from the point Q_k in the direction of p_k. Therefore, in each iteration of the trust-region method, instead of solving the main problem, problem (14) will be solved.

As mentioned, choosing the radius of the trust region is important. If the performance of the algorithm is good, then the radius of the region must be increased in order to have a better speed of convergence. If the algorithm performance is poor, then the trust region decreases for greater accuracy. The performance of the algorithm in each iteration is determined by the following formula:

$$\rho_k = \frac{f(Q_k) - f(Q_k + p_k)}{m_k(0) - m_k(p_k)} = \frac{\|r(Q_k)\|^2 - \|r(Q_k + p_k)\|^2}{\|r(Q_k)\|^2 - \|r(Q_k) + J(Q_k)p_k\|^2}. \tag{15}$$

If $\rho_k = 1$, then the approximate function m_k and f will be closer to each other in the existing area. In other words, if the value is closer to one, then the algorithm has a better performance. The main steps of trust-region can be summarized by a pseudo code as Algorithm 2 below.

Algorithm 2 Trust-region algorithm

Input: $\hat{\Delta} > 0$, $\Delta_0 \in (0, \hat{\Delta})$, and $\eta \in [0, \frac{1}{4})$.

1: For $k = 0, 1, 2, \ldots$, do the following operations:

1-1 : Obtain the value of p_k by solving (14).

1-2 : Calculate the value of ρ_k using relation (15).

1-3 : Find Δ_{k+1}, using ρ_k by

1-3-1: **If** $\rho_k < \frac{1}{4}$, **then** $\Delta_{k+1} = \frac{1}{4}\Delta_k$.

1-3-2: **If** $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$, **then** $\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$; **else** $\Delta_{k+1} = \Delta_k$.

1-4: Find Q_{k+1}, using ρ_k and η by

1-4-1: **If** $\rho_k > \eta$, **then** $Q_{k+1} = Q_k + p_k$; **else** $Q_{k+1} = Q_k$.

Output: Q_k and $f(Q_k)$.

The details of the steps associated with Algorithm 2 are described next.
The maximum trust-region radius, the trust-region radius for the first iteration, and η should be given as input to the algorithm. The parameters of our proposed algorithm are set by IRACE PACKAGE [31] to ensure fair space, $\hat{\Delta} = 0.9$, $\Delta_0 = 0.9$ and $\eta = 0.2$.

In step 1-2, the optimal solution is obtained through an iterative process. For this purpose, in each iteration in step 1-1, a quadratic approximation (14) of the original objective function ($\frac{1}{2}\|r(Q)\|^2$) is calculated based on the solution of the previous iteration.

Then by solving the updated equation (14), the previous iteration solution improves. To solve equation (14), the Dogleg algorithm (The Dogleg algorithm is described below) is used. In step 1-2, based on the solution obtained in step 1-1, the value of ρ_k is calculated. A larger ρ_k (close to 1) indicates that the approximate function of (14) and the original objective function are close to each other.

Based on the calculated ρ_k in step 1-3, the trust-region radius of the next iteration is decided. Also, $\rho_k < \frac{1}{4}$ indicates that the approximation obtained from step 1-1 is not an appropriate approximation for the original objective function. In this case, in order to increase accuracy, the trust-region radius becomes smaller. In addition, $\rho_k > \frac{3}{4}$ indicates that the approximation obtained from step 1-1 is a very good approximation for the original objective function. In this case, in order to increase the speed of convergence, the trust-region radius increases. Moreover, $\frac{1}{4} < \rho_k < \frac{3}{4}$ indicates that the approximation obtained from step 1-1 is a normal approximation for the original objective function. In this case, the trust-region radius does not change.

In step 1-4, a decision is made based on ρ_k whether or not to accept the current iteration solution. Also, $\rho_k < \eta$ indicates that the approximation obtained from step 1-1 is not a suitable approximation for the original objective function. Therefore, accepting the solution obtained from the approximate function may complicate the convergence process of the algorithm. For this reason, in this case, the solution obtained from step 1-1 will not be accepted. In this case, step 1-1 is repeated with the same approximation function as the previous one, except that the trust-region radius is reduced in step 1-3. Hence, it is expected that repeating step 1-1 will lead to a more accurate solution.

The parameters and variables of the proposed method for solving the equations of the WDN are reported in Table 2.

Table 2: Parameters and variables of the proposed method for solving the equations of the WDN

symbol	Type	Expression
R	Parameter	The resistance constant of the pipe;
Q_0	Parameter	The initial flow of pipes (initial solution);
q	Parameter	The flow that exits (or enters) the network at each node;
Δ_0	Parameter	The radius of the trust-region is the first iteration;
$\hat{\Delta}$	Parameter	Maximum radius of the trust-region;
η	Parameter	The value used to reject or confirm the solution to each iteration;
Q	Variable	The flow that passes through the network pipes.

3.3 Dogleg Algorithm

This section describes the Dogleg algorithm. The Dogleg algorithm first removes the quadratic phrase of the objective function (14) ($\frac{1}{2}p^TJ_k^TJ_kp$) and solves the following linear optimization problem:

$$\min_{p\in\mathbb{R}^n} f_k + p^TJ_k^Tr_k \quad \text{s.t. } \|p\| \leq \Delta_k. \tag{16}$$

The solution obtained from the problem (16) is called p_k^s. Since the objective function of problem (16) is linear, it can be solved easily. We know that the value of the objective function always decreases by moving in the direction $-g_k = -J_k^T r_k$. Therefore, the lowest value of the objective function (16) is obtained for $p_k^s = -\alpha J_k^T r_k$, so that the higher α, the lower the value of the objective function (16). Given that in problem (16) $\|p_k^s\| \leq \Delta_k$, α value is equal to $\frac{\Delta_k}{\|J_k^T r_k\|}$. Hence, the solution to problem (16) is obtained from

$$p_k^s = -\frac{\Delta_k}{\|J_k^T r_k\|} J_k^T r_k. \tag{17}$$

Linearizing the objective function of problem (14) reduces the accuracy of the obtained solution. For this reason, after calculating the solution to problem (16), the Dogleg algorithm solves the following quadratic problem in order to increase accuracy:

$$\min_{\tau \geq 0} \ m_k(\tau p_k^s) \quad \text{s.t. } \|\tau p_k^s\| \leq \Delta_k. \tag{18}$$

The solution obtained from the problem (18) is called τ_k. Since the objective function (18) is a univariate quadratic function, solving problem (18) is very simple (Set the differential of the function equal to zero and solve a simple equation). Accordingly, the least value of the objective function (18) occurs for $\tau_k = \frac{\|J_k^T r_k\|^3}{\Delta_k r_k^T J_k (J_k^T J_k) J_k^T r_k}$. According to the constraints of problem (18), it must be $\|\tau_k p_k^s\| \leq \Delta_k$. Since $\|p_k^s\| = \Delta_k$ (according to (17)), it must be $\tau_k \leq 1$. Thus τ_k is obtained from (19).

$$\tau_k = \min\{1, \frac{\|J_k^T r_k\|^3}{\Delta_k r_k^T J_k (J_k^T J_k) J_k^T r_k}\}. \tag{19}$$

So, if $\tau_k < 1$, then $\tau_k p_k^s$ is a better solution to the problem (14) compared to p_k^s. The $\tau_k p_k^s$ is called p_k^c. and it obtains accordingly as follows.

$$p_k^c = -\tau_k (\frac{\Delta_k}{\|J_k^T r_k\|}) J_k^T r_k. \tag{20}$$

If $\|p_k^c\| = \Delta_k$, then the Dogleg algorithm considers $p_k = p_k^c$ as the approximation solution to (14). If $\|p_k^c\| < \Delta_k$, then we provide another direction to calculate the solution to (14) for increasing the convergence speed. To determine this direction, the following problem must be solved unconstrained:

$$\min_{p \in \mathbb{R}^n} \ m_k(p) = f_k + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p. \tag{21}$$

The solution to the problem (21) is called p_k^j. Problem (21) is an unconstrained quadratic problem. Therefore, to obtain p_k^j, it suffices to set the differential of the objective function to zero ($J_k^T r_k + \frac{1}{2} J_k^T J_k p = 0$). By solving this simple linear equation, we have

$$p_k^j = -(J_k^T J_k)^{-1}(J_k^T r_k) = -J_k^{-1} r_k. \tag{22}$$

For finding the approximation solution to problem (14), the Dogleg algorithm uses the combination of the two directions p_k^c and p_k^j. The main steps of the Dogleg algorithm can be summarized as Algorithm 3 below.

Algorithm 3 Dogleg algorithm

Input: Trust-region radius (Δ_k), Jacobian matrix (J_k) and Vector of transactions (r_k)

1: Calculate the value of p_k^c using relation (20).

1-1: **If** $\|p_k^c\| = \Delta_k$, **then** set $p_k = p_k^c$.

1-2: **else**, do the following:

1-2-1: Calculate the value of p_k^j using relation (22).

1-2-2: Set $p_k = p_k^c + \tau(p_k^j - p_k^c)$.

1-2-3: Calculate the maximum value of $\tau \in [0, 1]$ as $\|p_k\| \le \Delta_k$.

Output: Vector p_k.

3.4 Convergence of the Proposed Method

In this paper, to solve equations of WDN, the optimization problem (10) is solved using the trust-region method. problem (10) is an unconstrained optimization problem. So if $\nabla(f(Q)) = 0$, then Q will be the optimal solution. In [38], it proved that the gradient sequence created in the trust-region method converges to zero (for $\eta > 0$). Hence, to solve the problem (10) the trust-region method is convergent. In addition, the convergence of the trust-region method in the general case has also been proved in [38].

4 Numerical Results

In the hydraulic analysis software of WDN, the use of the gradient method to solve network equations is popular. Therefore, in this section, we compare the performance of the proposed method with the gradient method using several numerical examples. All executions are done on a notebook with characteristics of CPU: intel core i5 2520M 2.5GHz with 8 GB RAM under Windows 7 home premium in MATLAB R2017b software. In the following, the gradient method for solving the equations of WDN is briefly explained. Then, study examples are introduced, and finally, the performance of the two methods of trust-region and gradient are compared in terms of accuracy and speed.

4.1 Gradient Method

In order to hydraulically analyze a WDN, its hydraulic equations must be solved. The gradient method is currently used in many popular commercial software, such as WATERGEMS and EPANET to solve these equations. For solving the equations of the WDN, the gradient method solves a linear equation system in each iteration. This system includes two types of equations. The first type is continuity equations (4) that do not need to be updated in each iteration. The second type is the below equations (23), which must be updated in each iteration according to the solution of the previous iteration.

$$H_{t+1,oi} - H_{t+1,oj} - (nR_{ox}|Q_{t,ox}|^{n-1})Q_{t+1,x} = (1-n)R_{ox}Q^{n}_{t,ox}, \qquad x = 1,\dots,NP, \qquad (23)$$

where, $H_{t+1,oi}$ and $H_{t+1,oj}$ represent the head in nodes i and j in the iteration of $t+1$, respectively. Also, R_{ox} indicates the resistance of the pipe and $Q_{t,ox}$ the pipe flow x in the iteration of t. Thus the gradient method in each iteration forms a linear equation system and then solves it. For more information, we refer the reader to [51].

4.2 Examples

In this section, some study examples are introduced.

Example 1. [49] Figure 2 shows a simple WDN. This network has no valve and pump and also has two source nodes and four consumption nodes. The flow equation system of this network has seven equations. This system includes four linear equations and three nonlinear equations.

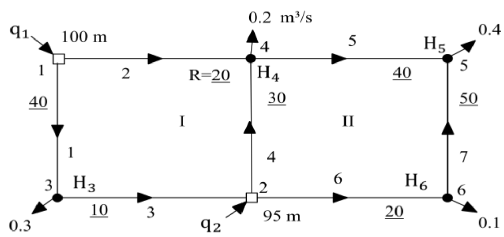

Figure 2: WDN of Example 1 [49].

Example 2. Figure 3 shows a WDN having two source nodes and four consumption nodes. This network consists of four loops. The system of equations related to this network consists of four linear equations and five nonlinear equations. Therefore, in this example, the number of nonlinear equations is more than linear equations. The resistance constant of the pipes of this network is reported in Table 3, and the amount of harvest in the consumption nodes is reported in Table 4.

Table 3: Constant resistance of network pipes of Figure 3

Pipe	R	Pipe	R
1	20	6	10.7365
2	30	7	30
3	40	8	200
4	100	9	200
5	23.53		

Table 4: Water withdrawal from network consumption nodes of Figure 3

Consumption node	2	3	4	5
Harvest rate	0	0.2	1.1590	0.7059

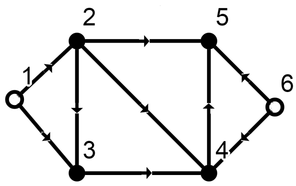

Figure 3: WDN of Example 2.

Example 3. [49] The WDN of Figure 4 consists of a pump and two spring nodes. The flow equations of this network have eleven variables. These equations consist of seven linear equations and four nonlinear equations. The pipe resistance constant of this network is given in Table 5.

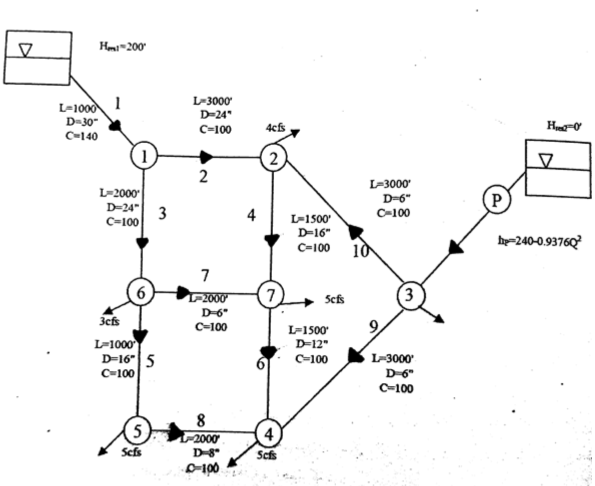

Figure 4: WDN of Example 3 [49].

Table 5: Constant resistance of network pipes of Figure 3

Pipe	R	Pipe	R
1	0.0072	7	68.5175
2	0.1202	8	16.8791
3	0.0801	9	102.7762
4	0.4329	10	102.7762
5	0.2886	11	0.3055
6	1.7573		

Example 4. The WDN of Figure 5 consists of seventeen nodes, twenty pipes, and four loops. The flow equations for this example have twenty variables. The amount of discharge from the nodes of this network, as well as the resistance constant of its pipes, is given in Table 6. The values of nodes 1 and 13 are 300 and 250, respectively.

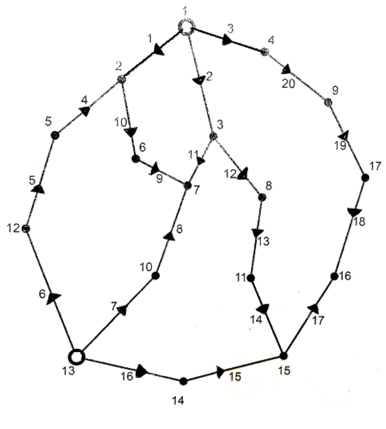

Figure 5: WDN of Example 4.

Table 6: Flow rate picked up and constant resistance of network pipes of Figure 5

Row	Constant pipe resistance R	flow rate picked up
1	40	*
2	10	0.7956
3	60	0.2089
4	60	0.3536
5	10	0.7716
6	10	0.5142
7	80	2.0484
8	120	0.1157
9	15	0.1726
10	12	0.0511
11	240	0.0931
12	80	0.2092
13	120	*
14	120	0.3559
15	10	0.8339
16	10	1.2614
17	20	0.1186
18	120	*
19	120	*
20	120	*

Example 5. The WDN of Figure 6 consists of sixty-three nodes, 110 pipes, and 48 loops. The flow equations for this example have 110 variables. The resistance constant of the pipes related to this network is written on the pipes of Figure 6. The amount of discharge from the nodes of this network is given in Table 7. The head of nodes 1 and 55 are 200 and 100, respectively.

Table 7: Flow rate picked up and constant resistance of network pipes of Figure 6

Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
flow rate picked up	-	0.3	0.2	0.1	0.3	0.5	0.3	0.7	0.2	0.5	0.3	0.2	0.1	0.4	0.1	0.3	0.5	0.2	0.5	0.1	0.2
Row	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
flow rate picked up	0.3	0.6	0.5	0.3	0.2	0.3	0.2	0.5	0.2	0.3	0.4	0.2	0.3	0.2	0.3	0.2	0.3	0.1	0.3	0.3	0.2
Row	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
flow rate picked up	0.1	0.3	0.5	0.3	0.2	0.3	0.2	0.5	0.2	0.7	-	0.3	0.2	0.5	0.4	0.5	0.3	0.5	0.1	0.5	0.6

4.3 Examining the Trust-Region Method

In the following, we compare the trust-region method with the gradient method in terms of convergence speed and accuracy. Table 8 shows the results of the trust-region and gradient methods for Examples 1 to 5. In Table 4 the stopping criterion for both methods is considered $|f(Q_k) - f(Q_{k-1})| < \epsilon$.

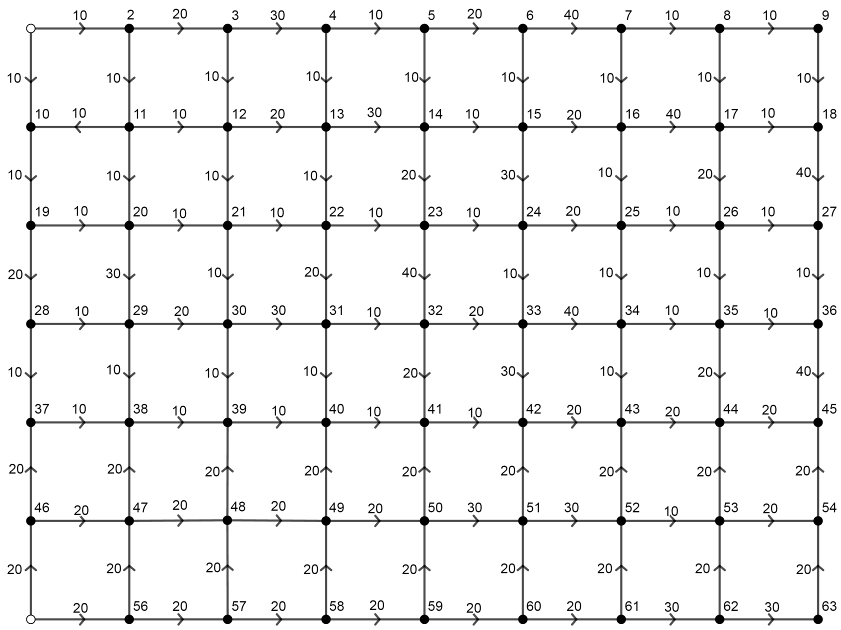

Figure 6: WDN of Example 5.

Table 8: Comparison of trust-region and gradient methods

	Convergence	Example 1	Example 2	Example 3	Example 4	Example 5
Dimension		7	9	11	20	110
Gradient	Time	0.018	0.011	0.020	0.023	-
	Iteration	5	5	5	4	-
	Accuracy	0.0054	$2.52e-05$	60.2242	0.0961	Not converge
Trust-Region	Time	0.052	0.032	0.043	0.045	1.49
	Iteration	8	8	31	6	26
	Accuracy	$2.88e-31$	$1.20e-29$	0.0224	$5.87e-28$	$9.85e-27$

By using the value of the objective function of problem (10), we can conclude that if the value of the objective function of problem (10) is low then the accuracy of the obtained solution is high. Hence table 8 compares the accuracy of the trust-region method and the gradient method based on the objective function value of problem (10). As can be seen, in Examples 1 to 4, the accuracy of the trust-region method is much better than the gradient method. Example 5 is related to a relatively large water distribution network. The gradient method does not achieve convergence in solving the hydraulic equations of this network, but the trust-region method solves the equations of this network with reasonable accuracy and implementation time.

EPANET software is a common software in the hydraulic analysis of WDNs. This software uses the gradient method to solve network equations. For a more applied comparison, the following network (Figure 7) was implemented in EPANET software. Also, the hydraulic equations governing this network were solved by the trust-region method.

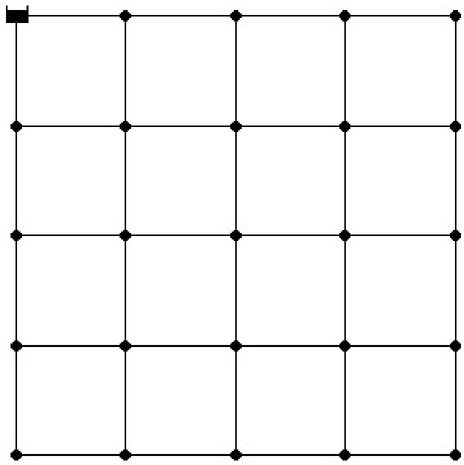

Figure 7: Implement a WDN in the EPANET software.

Table 9 compares the accuracy of the trust-region method with the accuracy of solving equations by EPANET.

Table 9: Comparison of trust region and EPANET software

Method	Solution accuracy	Dimension
EPANET (Gradient)	0.0080	40
Trust-region	0.0010	40

As can be seen, solving the network equations using the trust-region method is more accurate than the solution obtained from the EPANET software.

In general, more convergence and better accuracy are the advantages of the trust-region method compared to the gradient method. Hence, using the trust-region method compared to the gradient method can provide a better hydraulic analysis of a water distribution network.

The gradient method has performed somewhat better in terms of convergence speed. Therefore, changes in the method of trust-region to increase the speed of convergence can be considered for future research.

5 Conclusion

Here, for solving nonlinear hydraulic equations, we proposed a trust-region method. We solved some randomly generated test examples and made a comparative study to show the effectiveness of our proposed algorithm with the gradient method. The results showed that the trust-region method is more accurate than the gradient method, and also, the results show that the gradient method can not be converged when the dimensions of the problem become high, while the trust-region method solved these equations with suitable accuracy. Therefore, using the trust-region method can provide a better hydraulic analysis of a WDN.

Acknowledgments

The first and second authors thank the Research Council of Ferdowsi University of Mashhad and Optimization Laboratory of Ferdowsi University of Mashhad and the third author thanks the Mosaheb Institute of Mathematics, Kharazmi University for supporting this work.

Data Availability Statement

Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Amat S., Busquier S., Gutiérrez J.M. (2003). "Geometric constructions of iterative functions to solve nonlinear equations", Journal of Computational and Applied Mathematics, 157, 197-205.

[2] Aragones D.G., Calvoa G.F., Galan A. (2021). "A heuristic algorithm for optimal cost design of gravity-fed water distribution networks. a real case study", Applied Mathematical Modelling, 95, 379-395.

[3] Ates S. (2017). "Hydraulic modelling of control devices in loop equations of water distribution networks", Flow Measurement and Instrumentation, 53, 243-260.

[4] Bermúdez J.R., Estrda F.R.L., Besanćon G., Palomo G.V., Torres L., Hernández H.R. (2018). "Modeling and simulation of a hydraulic network for leak diagnosis", Mathematical and Computational Applications, 23.

[5] Bertsekas D.P. (2014). "Constrained optimization and Lagrange multiplier methods", Academic Press.

[6] Bhave P.R. (1991). "Analysis of flow in water distribution networks", Technomic Publishing Co., Inc., Lancaster.

[7] Brkić D. (2011). "Iterative methods for looped network pipeline calculation", Water Resources Management, 25, 2951-2987.

[8] Brkić D., Hansen P. (2009). "An improvement of hardy cross method applied on looped spatial natural gas distribution networks", Applied Energy, 86, 1290-1300.

[9] Broyden C.G. (1965). "A class of methods for solving nonlinear simultaneous equations", Mathematics of Computation, 19, 577-593.

[10] Collins A.G., Johnson R.L. (1975). "Finite-element method for water-distribution networks", American Water Works Association, 67, 385-389.

[11] Costa C.M., Grapiglia G.N. (2020). "A subspace version of the Wang–Yuan augmented Lagrangian-trust-region method for equality constrained optimization", Applied Mathematics and Computation, 387.

[12] Cross H. (1936). "Analysis of flow in networks of conduits or conductors", University of illinois at urbana champaign, College of Engineering.

[13] Djebedjiana B., Abdel-Gawad H. A.A., Ezzeldin R.M. (2021). "Global performance of metaheuristic optimization tools for water distribution networks", Ain Shams Engineering Journal, 12, 223-239.

[14] Donachie R.P. (1974). "Digital program for water network analysis", Journal of the Hydraulics Division, 100, 393-403.

[15] Elhay S., Piller O., Deuerlein J., Simpson A. (2016). "A robust, rapidly convergent method that solves the water distribution equations for pressure-dependent models", Journal of Water Resources Planning and Management, 142(2), 04015047-1.

[16] El-Sobky B. Elnaga Y.A. (2018). "A penalty method with trust-region mechanism for nonlinear bilevel optimization problem", Journal of Computational and Applied Mathematics, 340, 360-374.

[17] Fan J., Pan J. (2011). "An improved trust-region algorithm for nonlinear equations", Computational Optimization and Applications, 48, 59-70.

[18] Giustolisi O., Laucelli D. (2011). "Water distribution network pressure-driven analysis using the enhanced global gradient algorithm (EGGA)", Journal of Water Resources Planning and Management, 137, 498-510.

[19] Grosan C., Abraham A. (2008). "A new approach for solving nonlinear equations systems", IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 38, 698-714.

[20] Hamlehdar M., Yousef H., Noorollahi Y., Mohammadi M. (2022). "Energy recovery from water distribution networks using micro hydropower: A case study in Iran", Energy, 252, 124024.

[21] Huzsvár T., Wéber R., Déllei Á., Hös C. (2021). "Increasing the capacity of water distribution networks using fitness function transformation", Water Research, 201, 117362.

[22] Isaacs L.T., Mills K.G. (1980). "Linear theory methods for pipe network analysis", Journal of the Hydraulics Division, 106, 1191-1201.

[23] Jeppson R.W. (1976). "Analysis of flow in pipe networks", Butterworth-Heinemann.

[24] Jerez D.J., Jensen H.A., Beer M., Broggi M. (2021). "Contaminant source identification in water distribution networks: A Bayesian framework", Mechanical Systems and Signal Processing, 159.

[25] Kelley C.T. (2003). "Solving nonlinear equations with Newton's method", Society for Industrial and Applied Mathematics.

[26] Koşucu M.M., Albay E., Demirel M.C. (2022). "Extending EPANET hydraulic solver capacity with rigid water column global gradient algorithm", Journal of Hydro-Environment Research, 42, 31-43.

[27] Kumar S.M., Narasimhan S., Bhallamudi S.M. (2010). “Parameter estimation in water distribution networks”, Water Resources Management, 24, 1251-1272.

[28] Lam C.F., Wolla M. (1972). “Computer analysis of water distribution systems: Part II–numerical solution”, Journal of the Hydraulics Division, 98, 447-460.

[29] Lemieux P.F. (1972). “Efficient algorithm for distribution networks”, Journal of the Hydraulics Division, 98, 1911-1920.

[30] Liu K. (1969). “The numerical analysis of water supply networks by digital computers, in: Thirteenth congress of the international association for hydraulic Research”, Applied Soft Computation, 1, 36-43.

[31] López-Ibáñez M., Dubois-Lacoste J., Pérez Cáceres L., Birattari M., Stützle T. (2016). “The irace package: Iterated racing for automatic algorithm configuration”, Operations Research Perspectives, 3, 43-58.

[32] Mabrok M.A., Saadb A., Ahmed T., Alsayab H. (2022). “Modeling and simulations of water network distribution to assess water quality: Kuwait as a case study”, Alexandria Engineering Journal, 61, 11859-11877.

[33] Mankad J., Natarajan B., Srinivasan B. (2022). “Integrated approach for optimal sensor placement and state estimation: A case study on water distribution networks”, ISA Transactions, 123, 272-285.

[34] Martin D., Peters G. (1963). “The application of newton’s method to network analysis by digital computer”, Journal of the Institute of Water Engineering, 17.

[35] Mohades M.M., Kahaei M.H., Mohades H. (2021). “Haplotype assembly using Riemannian trust-region method”, Digital Signal Processing, 112, 102999.

[36] Moosavian N., Jaefarzadeh M. (2014). “Multistage linearization method for hydraulic analysis of water distribution network”, Journal of Computational Methods in Engineering, 32, 173-187.

[37] Nielsen H.B. (1989). “Methods for analyzing pipe networks”, Journal of Hydraulic Engineering, 115, 139-157.

[38] Nocedal J., Stephen S.J. (2006). “Numerical optimization”, Springer.

[39] Nogueira A.C. (1993). “Steady-state fluid network analysis”, Journal of Hydraulic Engineering, 119, 431-436.

[40] Ormsbee L.E. (2008). “The history of water distribution network analysis: The computer age”, 8th Annual Water Distribution Systems Analysis Symposium, 1-6.

[41] Petkovic M., Neta B., Petkovic L., Dzunic J. (2012). “A hybrid method for non-linear equations”, Elsevier,

[42] Powell M.J.D. (1970). “A hybrid method for non-linear equations”, Numerical Methods for Nonlinear Algebraic Equations, 14, 87-114.

[43] Powell M.J.D. (1978). “Algorithms for nonlinear constraints that use Lagrangian functions”, Mathematical Programming, 14, 224-248.

[44] Rao H., Bree D.W. (1977). “Extended period simulation of water systems–Part A”, Journal of the Hydraulics Division, 103, 97-108.

[45] Shamir U.Y., Howard C.D. (1968). “Water distribution systems analysis”, Journal of the Hydraulics Division, 94, 219-234. .

[46] Sheng Z., Luo D. (2020). “A Cauchy point direction trust-region algorithm for nonlinear equations”, Mathematical Problems in Engineering, 2020.

[47] Simpson A., Elhay S. (2011). “Jacobian matrix for solving water distribution system equations with the Darcy–Weisbach head-loss model”, Journal of Hydraulic Engineering, 137, 696-700.

[48] Swamee P.K., Sharma A.K. (2008). “Design of water supply pipe networks”, John Wiley & Sons.

[49] Tabesh M. (1998). “Implications of the pressure dependency of outflows of data management, mathematical modelling and reliability assessment of water distribution systems”, PhD Thesis, University of Liverpool.

[50] Tanyimboh T., Tahar B., Templeman A. (2003). “Pressure-driven modelling of water distribution systems”, Water Science and Technology: Water Supply, 3, 255-261.

[51] Todini E., Pilati S. (1988). “A gradient algorithm for the analysis of pipe networks”, Computer Applications in Water Supply, 1-20.

[52] Walski T. (2018). “Water distribution system analysis before the digital age”, WDSA/CCWI Joint Conference Proceedings, 1.

[53] Wood D.J., Charles C.O. (1972). “Hydraulic network analysis using linear theory”, Journal of the Hydraulics Division, 98, 1157-1170.

[54] Yang P., Jiang Y.L., Xu K.L. (2019). “A trust-region method for H2 model reduction of bilinear systems on the Stiefel manifold”, Journal of the Franklin Institute, 356, 2258-2273.

[55] Yuan Y. (2015). “Recent advances in trust-region algorithms”, Mathematical Programming, 151, 249-281.

[56] Zarghamee M.S. (1971). “Mathematical model for water distribution systems”, Journal of the Hydraulics Division, 97, 1-14.

How to Cite this Article:
Ahmadnia, M., Ghanbari, R., Ghorbani-Moghadam, Kh. (2022). “Solving nonlinear hydraulic equations of water distribution networks by using a trust-region method”, Control and Optimization in Applied Mathematics, 7(2): 53-75. doi: 10.30473/coam.2022.62968.1195