Computational and Applied Mathematics (2025) 44:210
https://doi.org/10.1007/s40314-025-03158-2

A note on local convergence of iterative processes for pipe network analysis

Huong Luu[1] · **Marek Chrobak**[1]

Received: 29 May 2024 / Revised: 21 February 2025 / Accepted: 24 February 2025 /
Published online: 3 April 2025
© The Author(s) 2025

Abstract
Analysis of pipe networks involves computing flow rates and pressure differences on pipe segments in the network, given the external inflow/outflow values. This analysis can be conducted using iterative methods, among which the algorithms of Hardy Cross and Newton-Raphson have historically been applied in practice. In this note, we address the mathematical analysis of the local convergence of these algorithms. The loop-based Newton–Raphson algorithm converges quadratically fast, and we provide estimates for its convergence radius to correct some estimates in the previous literature. In contrast, we show that the convergence of the Hardy Cross algorithm is only linear. This provides theoretical confirmation of experimental observations reported earlier in the literature.

Keywords Pipe network analysis · Iterative processes · Local convergence

Mathematics Subject Classification 34A34 · 49M15 · 76M25

1 Introduction

The analysis of pipe networks is a well-studied problem in civil engineering that arises in the design and modeling of large-scale hydraulic networks, such as water municipal systems, sewage systems, or substance distribution networks. In this paper we focus on the most common formulation, where the objective is to determine the flow rates and pressure differences in all pipe segments within the network, given its topology, nodal inflows/outflows, and the parameters of the pipes and fluid (diameter, friction, fluid density, etc.) Stephenson (1984). At a fundamental level, this problem is analogous to the analysis of electrical circuits. These two problems share two similar physical principles: mass (or flow) conservation and energy conservation. The law of flow conservation states that at each node, the total inflows and outflows are equal; and the principle of conservation of energy requires that around a closed loop the total sum of pressure differences is zero. However, in contrast to their electrical counterparts, in large-scale pipe networks the equation governing the relationship between

✉ Huong Luu
hluu008@ucr.edu

[1] Department of Computer Science, University of California at Riverside, Riverside, USA

flow and pressure difference in a pipe segment is non-linear.[1] This equation, known as the Darcy-Weisbach equation, asserts that the pressure difference in a pipe segment e is proportional to the square of its flow: $|\Delta_e p| = \mu_e q_e^2$, where $|\Delta_e p|$ is the positive pressure difference, q_e is the flow value, and μ_e is a constant coefficient that is inversely proportional to the pipe's diameter, and proportional to the friction factor, pipe length, and fluid density.

Pipe network analysis boils down to solving the system of the above-mentioned equations, namely the flow conservation equations for all nodes and the energy conservation equations for all pipe segments. The specific choice of variables, be it pressure values or flow values, leads to different formulations of these equations, with three common approaches known as the loop method, the flow method, and the node method (Boulos et al. 2006). (See Sect. 2 and Appendix B.) In either formulation, the resulting system of equations is non-linear, and computing its exact solution is not feasible. Instead, iterative techniques that compute approximate solutions have been widely adopted in the scientific literature and in practice.

Past work One of the earliest approaches to pipe network analysis was proposed by Cross (1936). He introduced both the loop and node variants of his method,[2] that we denote *HC-loop* and *HC-node*, respectively. Due to its simplicity, Algorithm HC-loop has been used to analyze small networks with spreadsheets or hand calculations. Another frequently used approach involves applying the generic Newton-Raphson algorithm for solving non-linear equations. This algorithm can be applied to any of the above three settings: flow, loop or node, and we use notations *NR-flow*, *NR-loop* and *NR-node* for its respective variants.

In this work we focus mainly on algorithms based on the loop method. This method assumes that some initial flow satisfying the boundary conditions and flow conservation is given, and its unknowns are the flow adjustments along the network's cycles with respect to this initial flow (Sect. 2 for a detailed description).

While there is a fair amount of literature on experimental performance of these algorithms, relatively little is known about their provable convergence properties. In Cross (1936), Cross claimed that Algorithm HC-loop's convergence is "sufficiently rapid", though this claim was not supported by rigorous analysis. Adams (1955) provided an analysis of the accuracy of the flow corrections generated by Algorithm HC-loop by comparing it with the exact correction values. This result is useful for convergence analysis of simple systems that consist of only one cycle; however, for larger systems, the mathematical analysis is incomplete. Empirical results in Gameiro Lopes (2004) show fast convergence of HC-loop for several networks in the dataset. Altman and Boulos reported in Altman and Boulos (1995) an attempt to analyze local convergence of NR-flow; however, as we explain in Appendix A, their analysis is flawed. In Brkić (2009), Brkić conducted a comparative study of NR-loop and HC-loop, and his experimental results support the earlier observations that NR-loop exhibits faster convergence.

Our results We present an analysis of the local convergence of Algorithms NR-loop and HC-loop in Sects. 3 and 4. We show that Algorithm NR-loop converges quadratically fast if the initial flow is sufficiently close to the solution, and we provide a bound on the radius of its quadratic convergence. The quadratic convergence of the Newton–Raphson method is not surprising—in fact it is known to converge quadratically under some fairly mild assumptions. Our aim here was to correct and refine the analysis of the convergence radius provided in paper (Altman and Boulos 1995).

[1] This non-linearity is due to fluid turbulence. In small-scale networks, such as those that arise in microfluidics, flows tend to be laminar and their behavior can be modeled with linear equations.

[2] H. Cross referred to these approaches, respectively, as the methods of balancing flows and balancing heads.

For Algorithm HC-loop, we prove a similar local convergence property, although in this case the convergence rate is only linear. We also show that this bound is tight by exhibiting an example where Algorithm HC-loop's convergence is not better than linear. These results provide a theoretical confirmation of experimental observations, discussed earlier, about the superiority of NR-loop over HC-loop.

Additional comments As for other methods, Shamir and Howard commented in Shamir (1968); Shamir and Howard (1977) that Algorithm NR-node might not converge if it encounters a singular derivative matrix or if it enters and infinite loop, although no specific examples were given there. In Appendix B we provide an example on which Algorithm NR-node enters an infinite looop. A fairly extensive empirical comparison of different formulations of both Algorithm HC and Algorithm NR was presented by Wood (1981). The results indicate that the loop methods encounter fewer convergence issues than the respective node methods, and that the Newton–Raphson algorithm is more reliable for all formulation variants.

The network model we focus on in the paper is quite basic: the only network elements are pipes and junctions, and the boundary conditions are specified by external inflows and outflows. The loop method does not extend naturally to models that capture other features of practical pipe networks (see (Stephenson (1984), Chapter 3), for example), so in these models typically other formulations of the Newton–Raphson algorithm are used. For instance, in some network models the boundary conditions may be specified by pressure values at some nodes, rather than only inflows and outflows. And in more complex networks other hydraulic elements such as pumps, valves, and reservoirs are used to regulate and control pressure and flow. Shamir and Howard discussed in Shamir (1968) a special procedure to incorporate these elements into the system of governing equations. They also noted that these elements introduce complications to the convergence analysis as their characteristic equations may not have continuous derivatives. For recent progress on designing algorithms for the analysis of more complex pipe networks we refer the reader to the work by Srinivasan et al. (2023), who study gas-pipe networks involving compressors. Besides the issues of feasibility and solution uniqueness, this paper also discusses a method for improving numerical conditioning of the Newton–Raphson algorithm.

2 Preliminaries

Flow graphs A pipe system can be represented by an undirected graph $G = (V, E)$, where $|V| = n$ and $|E| = m$. In G, the pipe junctions are represented by vertices and pipe segments by edges. We assume that G is biconnected, as otherwise its biconnected components can be analyzed separately. For notational reasons, it will be convenient to assign orientations to edges. To this end, we assume that $V = \{1, 2, ..., n\}$, and we define the orientation of an edge e between two different vertices u and v as being from u to v if $u < v$, and from v to u otherwise. For simplicity, the edges in E will be identified by numbers $1, 2, ..., m$. Let D be the $m \times n$ *incident matrix* where $D_{ev} = 1$ if edge e is incident with vertex v and is oriented toward v, $D_{ev} = -1$ if e is oriented away from v, and $D_{ev} = 0$ if e is not incident with v. The specification of a flow graph also involves the *consumption vector* $w \in \mathbb{R}^n$ that satisfies $\sum_{v=1}^{n} w_v = 0$. This vector represents external inflows and outflows, which serve as the boundary conditions of the flow.

Cycles A closed loop in the pipe network corresponds to a cycle in its flow graph G. More generally, we define a *cycle* in G as an even-degree subgraph of G. A cycle is called *simple* if it is connected and all of its vertices have degree two. If we view cycles as vectors in $\mathbb{Z}_2^m$ ($\mathbb{Z}_2$

is the finite field of two elements 0 and 1), the set of all cycles in G forms a linear space over $\mathbb{Z}_2$ with vector addition being the operation of symmetric difference. The dimension of this space is $k = m - n + 1$. A *cycle basis* in G is a collection of k linearly independent simple cycles. Each cycle in G can be obtained as a linear combination of the cycles in the basis. For our analysis, we assume that an arbitrary cycle basis C is provided for the flow graph G. The *total length of* C, denoted ℓ_C, is defined as the sum of the lengths of its cycles.

Each simple cycle in a basis C has two possible orientations, which we refer to as *clockwise* and *counter-clockwise*. These orientations are determined by the two cyclic orderings of vertices in the cycle. The clockwise orientation can be chosen arbitrarily and independently for each cycle in the basis. An edge e on cycle c is called *clockwise on* c if its orientation agrees with the clockwise orientation of c, and otherwise e is called *counter-clockwise on* c. It is worth noting that an edge can be part of multiple cycles and may be clockwise in one cycle while being counter-clockwise in another.

Given a cycle basis C, we will number its cycles 1, 2, ..., k. We can then define the *edge-cycle matrix* A of size $m \times k$ representing C. The value $A_{ec} \in \{-1, 0, 1\}$ indicates the relationship between the orientations of edge e and cycle c. Specifically, it is 1 if edge e is clockwise on c, -1 if e is counter-clockwise on c, and 0 if e is not part of c.

The flow method A flow in G is represented by a vector $q \in \mathbb{R}^m$, with q_e denoting the flow on edge e. The sign of q_e indicates the flow direction: $q_e > 0$ iff the flow direction of q agrees with the orientation of e and vice versa. Recall that $\Delta_e p$ denotes the pressure difference along an edge e; that is, if e is oriented from u to v then $\Delta_e p$ is equal to the pressure at u minus the pressure at v. From the Darcy-Weisbach equation, $\Delta_e p$ can be written as $\Delta_e p = |q_e|\, q_e$. (For simplicity, we assume that for all edges e the coefficient μ_e in the Darcy-Weisbach equations is equal 1. Our derivation can be easily adapted to arbitrary coefficients, although the final formulas can't be then expressed in terms of graph invariants only.)

The goal of the flow method is to compute a flow q that satisfies:

$$\begin{cases} D^\top q + w = \bar{0} & (1) \\ A^\top U\, q = \bar{0} & (2) \end{cases}$$

where $U = \mathrm{diag}(|q_e|)$ for $e = 1, 2, \cdots, m$, that is, U is the diagonal matrix whose entries are the absolute values of the flows, and $\bar{0}$ denotes the zero-vector of the appropriate dimension.

Equation (1) is the flows conservation property, and Equation (2) is the energy conservation property. Equation (2) states that the pressure differences along each cycle need to add up to 0. The energy conservation principle applies to all cycles in G, although for our purposes it is sufficient to include only the k equations for cycles in the given cycle basis C.

The loop method The loop method is fundamentally equivalent to the flow method. In the loop method, the flow conservation equations (1) are eliminated by the introduction of an arbitrary *reference flow* $\psi \in \mathbb{R}^m$ that already satisfies this equation. Then, the flows in G can be expressed as:

$$q(x) = \psi + A\, x, \tag{3}$$

where $x \in \mathbb{R}^k$ is a variable whose value represents the flow adjustments along cycles in C. Thus the flow on an edge e is $q_e(x) = \psi_e + \sum_{c=1}^{k} A_{ec} x_c$. In other words, along each cycle c, x_c is added to the flows on its clockwise edges and subtracted from the flows on its counter-clockwise edges. In this setting, we define the error function $F : \mathbb{R}^k \mapsto \mathbb{R}^k$ to represent the deviation from the energy conservation equations and, expressed using function, the objective of the loop method is to find an x that satisfies

$$F(x) = A^\top U(x)\, q(x) = \bar{0}, \tag{4}$$

where $U(x)$ is the $m \times m$ diagonal matrix defined above for the flow $q(x)$. That is, if $F(x) = [f_1(x), f_2(x), ..., f_k(x)]^\top$ then $f_c(x) = \sum_{e=1}^{m} A_{ec}\, q_e(x)|q_e(x)| = 0$, for each cycle $c = 1, 2, ..., k$ in the cycle basis C.

We assume that this system has a solution and that the solution is unique. Both existence and uniqueness can be established by using the equivalence to the flow method, and reducing the flow method equations to solving a convex optimization problem (Singh and Kekatos 2021). In Singh and Kekatos (2021), a different proof of uniqueness is also given.

Iterative algorithms Our focus is on the analysis of solving the non-linear system (4) using two iterative algorithms discussed in the introduction: NR-loop and HC-loop. In general, an iterative algorithm starts with some initial candidate solution $x^{(0)}$, and then it produces successive approximate solutions $x^{(t)}$, attempting to reduce the value of $\|F(x^{(t)})\|$. The process repeats until some stopping criteria are satisfied. Such criteria may involve, for example, an upper bound on the maximum number of iterations, or the improvement being below a desired value. This improvement can be measured either by $\|x^{(t)} - x^{(t-1)}\|$ or $\|F(x^{(t)}) - F(x^{(t-1)})\|$.

An iterative algorithm is called *convergent* if it generates a sequence of values that converges to the actual solution, that is $F(x^{(t)}) \underset{t}{\rightarrow} \bar{0}$, if it's allowed to run indefinitely. The efficiency of an iterative process can be measured by its rate of convergence, which captures how quickly the generated sequence approaches the solution. The sequence $\left\{x^{(t)}\right\}$ is said to *converge to a solution* x^* *with* ω *-order* (Jay 2001), for some $\omega \geq 1$, if there exist two constants $\bar{\varepsilon} \geq 0$ and $\bar{t} \geq 1$ such that for all $t \geq \bar{t}$:

$$\|x^{(t)} - x^*\| \leq \bar{\varepsilon}\, \|x^{(t-1)} - x^*\|^{\omega}.$$

For $\omega = 1$ we additionally require that $\bar{\varepsilon} < 1$, and in this case the convergence is said to be *linear*. For $\omega = 2$ the convergence is called *quadratic*.

3 Analysis of Newton–Raphson algorithm

The Newton–Raphson algorithm is an iterative process widely used for finding zeros of non-linear functions. For single-argument functions $\mathbb{R} \rightarrow \mathbb{R}$, the algorithm obtains a new approximation value as the intersection of the tangent line to its graph at the current value with the x-axis. This concept can be naturally extended to multi-variate functions $\mathbb{R}^k \rightarrow \mathbb{R}^k$ by using the $k \times k$ Jacobian matrix, which represents the first-order partial derivatives.

Given a flow graph G, a cycle basis C of G and a reference flow ψ, we apply the Newton–Raphson algorithm (NR-loop) to solve the system of equations (4) starting from some initial solution $x^{(0)}$. (Note that we can as well assume that $x^{(0)} = \bar{0}$, for otherwise we can add $x^{(0)}$ to the reference flow and set up the equations for this modified reference flow.) At each step $t \geq 1$, the new solution $x^{(t)}$ is obtained from the previous one using the formula:

$$x^{(t)} \;=\; x^{(t-1)} - F'(x^{(t-1)})^{-1}\, F(x^{(t-1)})\,, \tag{5}$$

where $F'(x)$ denotes the Jacobian of F at x. The partial derivatives $\frac{\partial f_c}{\partial x_d}(x)$ are always well defined and equal to $2\sum_{e=1}^{m} A_{ec}\, A_{ed}\, |q_e(x)|$. Thus, $F'(x)$ can be expressed compactly in matrix form as:

$$F'(x) \;=\; 2\, A^\top\, U(x)\, A\,. \tag{6}$$

It is interesting to note that from the setup, it might seem that the flows $q(x^{(t)})$ obtained by applying the adjustment $x^{(t)}$ would be different for different cycle bases of G. However, upon closer examination, we find that it is not the case. This observation aligns with Nielsen's finding in Nielsen (1989). We offer a straightforward proof below.

Proposition 1 *Given a flow graph G, a reference flow ψ and and initial approximation $x^{(0)}$, the flows obtained at each step of Algorithm NR-loop are independent of the choice of cycle basis for G.*

Proof Let A, B be two edge-cycle matrices that correspond to two different cycle bases of G. Using A and B, set up the two error functions F_A, F_B and let their respective Jacobians be F'_A, F'_B. Denote by $x_A^{(t)}$, $x_B^{(t)}$ the adjustments at step t acquired from applying Algorithm NR-loop on F_A, F_B respectively. As explained earlier in this section, we can assume that $x_A^{(0)} = x_B^{(0)} = \bar{0}$. It is sufficient to show that the flow values on the edges are the same after the first step, no matter whether we use cycle base A or B, for we can then simply repeat this argument inductively to all steps. That is, we will show that $A\, x_A^{(1)} = B\, x_B^{(1)}$.

We have $B = AW$ for some $k \times k$ change of basis matrix W. For any given flow, its corresponding matrix U is independent of the cycle base, so $U_A(\bar{0}) = U_B(\bar{0})$. Assuming that $F'_A(\bar{0})$ and $F'_B(\bar{0})$ are invertible, we thus have:

$$\begin{aligned}
B\, x_B^{(1)} &= -B\, F'_B(\bar{0})^{-1}\, F_B(\bar{0}) \\
&= -\tfrac{1}{2} \cdot B\, [B^\top U_B(\bar{0})\, B]^{-1}\, B^\top U_B(\bar{0})\, \psi \\
&= -\tfrac{1}{2} \cdot A\, W\, [W^\top A^\top U_A(\bar{0}) A W]^{-1}\, W^\top A^\top U_A(\bar{0})\, \psi \\
&= -\tfrac{1}{2} \cdot A\, W\, W^{-1} [A^\top U_A(\bar{0}) A]^{-1}\, (W^\top)^{-1}\, W^\top A^\top U_A(\bar{0})\, \psi \\
&= -\tfrac{1}{2} \cdot A\, [A^\top U_A(\bar{0}) A]^{-1}\, A^\top U_A(\bar{0})\, \psi \\
&= -A\, F'_A(\bar{0})^{-1}\, F_A(\bar{0}) = A\, x_A^{(1)}\,.
\end{aligned}$$

Since the new flows are the same, we have $U_A(x_A^{(1)}) = U_B(x_B^{(1)})$, which in turn implies that the new Jacobian $F'_B(x_B^{(1)})$ is invertible if and only if $F'_A(x_A^{(1)})$ is invertible. This completes the proof. □

The convergence of the Newton–Raphson algorithm for non-linear systems has been studied extensively for decades. The rate of convergence can be as fast as quadratic if the function satisfies certain conditions, for example, when the initial approximation is in close neighborhood of the root and the Jacobians at all successive approximations are invertible. However, the root itself being unknown, it is generally difficult to determine if the initial point is "sufficiently" close to it. The Kantorovich (1948, 1964) theorem circumvents this challenge by giving the local convergence conditions in terms of the initial point and some general properties of the function. Our statement of this theorem, given below, follows the formulation in Ortega (1968) but is adapted to functions on Euclidean spaces with one unique root.

For $s \in \mathbb{R}^k$ and $r \in \mathbb{R}$, by $B(s, r)$ we denote the ball centered at s with radius r, that is $B(s, r) = \{x \in \mathbb{R}^k : \|x - s\| \le r\}$.

Theorem 1 *(Kantorovich) Let $F : \mathbb{R}^k \mapsto \mathbb{R}^k$ be a differentiable function, $F'(x)$ be the $k \times k$ Jacobian matrix of $F(x)$, and $x^{(0)} \in \mathbb{R}^k$ be an initial approximation of the Newton-Raphson process for $F(x) = \bar{0}$. Assume that:*

(i) $F'(x^{(0)})$ *is invertible and* $\|F'(x^{(0)})^{-1}\| \leq \beta$
(ii) $\|F'(x^{(0)})^{-1}\, F(x^{(0)})\| \leq \eta$
(iii) $\|F'(x) - F'(y)\| \leq L\|x - y\|$ *for all* $x, y \in \mathbb{R}^k$

for some $\beta, \eta, L > 0$.

With these assumptions, if $h = \beta\eta L < \frac{1}{2}$ *then the sequence* $\left\{x^{(t)}\right\}$ *generated by the Newton–Raphson's iteration process starting at* $x^{(0)}$ *is well-defined, contained in the ball* $B(x^{(0)}, r)$ *for* $r = (1 - \sqrt{1-2h})/(\beta L)$*, and converges quadratically to the unique solution* x^* *of* $F(x) = \bar{0}$.

Any induced matrix norm $\|\cdot\|$ can be used in this theorem. In our work, we assume the infinity norm on $\mathbb{R}^k$, that is $\|\cdot\| = \|\cdot\|_\infty$. Specifically, the norm of a vector $x \in \mathbb{R}^k$ is $\|x\| = \max_{1\leq c\leq k} |x_c|$, and the norm of a matrix $M \in \mathbb{R}^{k\times k}$ is $\|M\| = \max_{1\leq c\leq k} \sum_{d=1}^{k} |M_{cd}|$.

Given a flow graph G, we can use Theorem 1 to estimate the radius r of quadratic convergence expressed in terms of properties of F and attributes of G. To this end, we need to estimate the constants in (i), (ii) and (iii) from Theorem 1. (Before proceeding, we stress that, as the loop method computation is expressed in terms of flow-adjustment vectors $x \in \mathbb{R}^k$, the parameters β, η and L in the theorem may depend on the chosen cycle basis, even though, by Proposition 1, the actual sequence of edge-flow vectors in $\mathbb{R}^m$ obtained in this computation is not.)

Computing $F(x^{(0)})$ is straightforward, while the inverse of $F'(x^{(0)})$ requires more computation. However, since no provably accurate and efficient estimates for $\|F'(x)^{-1}\|$ are known, we can use $\beta = \|F'(x^{(0)})^{-1}\|$ and $\eta = \|F'(x^{(0)})^{-1}\, F(x^{(0)})\|$.

Our attention is now directed towards the bound (iii), which involves the Lipschitz condition on the Jacobian matrix F'. Let $\ell = \ell_C$ be the total length of C. We present a general estimate of L in terms of ℓ:

Claim 1 $\|F'(x) - F'(y)\| \leq L\|x - y\|$, *where* $L = 2k\,(\ell - k + 1)$.

Proof The derivation for this bound uses the formulas (6) for the Jacobian of $F(x)$ and (3) for the flow values. First, compute:

$$F'(x)_{cd} \;=\; \frac{\partial f_c}{\partial x_d}(x) \;=\; 2\sum_{e=1}^{m} A_{ec}\, A_{ed}\, |\psi_e + \sum_{i=1}^{k} A_{ei}\, x_i|$$

Then,

$$\begin{aligned}
\|F'(x) - F'(y)\| \;&=\; \max_{1\leq c\leq k} \sum_{d=1}^{k} \Big| \frac{\partial f_c}{\partial x_d}(x) - \frac{\partial f_c}{\partial x_d}(y) \Big| \\
&=\; \max_{1\leq c\leq k} \sum_{d=1}^{k} 2 \Big| \sum_{e=1}^{m} A_{ec}\, A_{ed} \Big(|\psi_e + \sum_{i=1}^{k} A_{ei}\, x_i| - |\psi_e + \sum_{i=1}^{k} A_{ei}\, y_i| \Big) \Big| \\
&\leq\; \max_{1\leq c\leq k} \sum_{d=1}^{k} 2 \sum_{e=1}^{m} |A_{ec}|\, |A_{ed}| \Big| \sum_{i=1}^{k} A_{ei}\, (x_i - y_i) \Big| \\
&\leq\; 2 \max_{1\leq c\leq k} \sum_{d=1}^{k} \sum_{e=1}^{m} |A_{ec}|\, |A_{ed}| \sum_{i=1}^{k} |A_{ei}| \cdot \|x - y\| \\
&\leq\; 2\,(\ell - k + 1)\, k\, \|x - y\|\,, \qquad (7)
\end{aligned}$$

because $\sum_{i=1}^{k} |A_{ei}| \le k$ and $\max_{1\le c\le k} \sum_{d=1}^{k} \sum_{e=1}^{m} |A_{ec}|\,|A_{ed}| \le \ell - k + 1$, with the last inequality following from the observation that for each cycle $d \ne c$, at least one of its edge is not on c. □

We are now ready to state sufficient conditions for quadratic convergence of Algorithm NR-loop on the loop equation (4). This condition applies to a flow graph $G = (V, E)$, a cycle basis C of G with total length ℓ, a reference flow ψ, and an initial solution $x^{(0)}$ where $F'(x^{(0)})$ is invertible:

Theorem 2 (Convergence conditions of Algorithm NR-loop) *Let* $\beta = \|F'(x^{(0)})^{-1}\|$, $\eta = \|F'(x^{(0)})^{-1}\,F(x^{(0)})\|$, *and* $L = 2k(\ell-k+1)$. *If* $\beta\eta L < \frac{1}{2}$, *then Algorithm NR-loop converges to the unique solution* x^* *of* $F(x) = \bar{0}$ *with quadratic order.*

In what follows, we discuss more specific estimates for L. The estimate of the convergence radius in Theorem 2 critically depends on the choice of the cycle basis. Trivially, for any basis, we have $\ell \le kn$. Therefore, in Theorem 2, ℓ can be replaced by kn if this bound is sufficient. However, in general, using a cycle basis C with small total length ℓ is likely (but not guaranteed) to improve the convergence radius.

The problem of computing cycle bases with small total length has been well studied. In Rizzi (2009), Rizzi introduced an $O(mn)$-time algorithm that generates a cycle basis of total length $\ell = O(m \log n)$, which is within a $2 \log n$ factor of the optimum length. This length bound was further refined to $\ell = O(m \log n / \log(m/n))$ by Kaufmann and Michail, as mentioned in (Kavitha et al. (2009),Theorem 4.5). If the loop equations are built upon the aforementioned cycle basis, the enhanced Lipschitz constant L becomes $2k(\tau m \log n / \log(m/n) - k + 1)$ for some constant τ.

For certain types of graphs, better estimates are possible. In particular, planar graphs emerge naturally when studying flows in pipe networks. If G is planar, the bound on the Lipschitz constant L can be improved by using a cycle basis in which each cycle is a face of G.

Observation 1 *If* G *is planar and* C *is a face-based cycle basis then* $\|F'(x) - F'(y)\| \le L\|x - y\|$, *where* $L = 8n$.

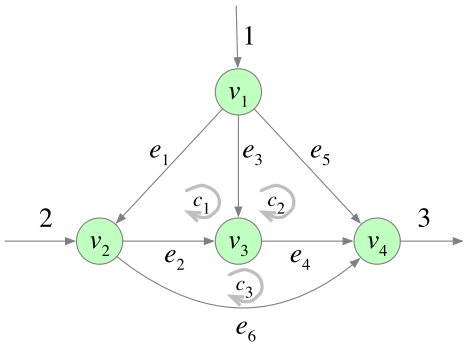
As a function of the graph size, this is at least an order-of magnitude improvement over the bound from Claim 1.

To justify Observation 1, note first that in a face-based cycle basis each edge belongs to at most two cycles, implying the inequality $\sum_{j=1}^{k} |A_{ej}| \le 2$. Then the observation follows from the bound (7) in the proof of Claim 1, by applying first this inequality twice (for $j = i$ and $j = d$) and then inequality $\sum_{e=1}^{m} A_{ec} \le n$ that holds for any simple cycle c.

Example 1 We illustrate the local convergence conditions on the simple flow graph depicted in Figure 1. The graph is planar and has 4 vertices, 6 edges, 2 inlets and 1 outlet. The cycle basis that is used to set up the loop equations is the cycle basis consisting of internal faces. The clockwise directions of the cycles are shown in this figure. The corresponding edge-cycle matrix A is shown on the right-hand side.

Let the reference flow be $\psi = [1, 1, 0, 1, 0, 2]$. The left-hand sides of the loop equations are:

$$\begin{cases} f_1(x) &= -|1 - x_1|(1 - x_1) - |1 - x_1 + x_3|(1 - x_1 + x_3) + |x_1 - x_2|(x_1 - x_2) \\ f_2(x) &= -|x_1 - x_2|(x_1 - x_2) - |1 - x_2 + x_3|(1 - x_2 + x_3) + |x_2|x_2 \\ f_3(x) &= |1 - x_1 + x_3|(1 - x_1 + x_3) + |1 - x_2 + x_3|(1 - x_2 + x_3) - |2 - x_3|(2 - x_3) \end{cases}$$



$$A = \begin{bmatrix} -1 & 0 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Fig. 1 The flow graph in Example 1

and, the Jacobian is: $F'(x) = [\frac{\partial F}{\partial x_1}(x), \frac{\partial F}{\partial x_2}(x), \frac{\partial F}{\partial x_3}(x)]$, where

$$\frac{\partial F}{\partial x_1}(x) = 2\begin{bmatrix} |1-x_1| + |1-x_1+x_3| + |x_1-x_2| \\ -|x_1-x_2| \\ -|1-x_1+x_3| \end{bmatrix}$$

$$\frac{\partial F}{\partial x_2}(x) = 2\begin{bmatrix} -|x_1-x_2| \\ |x_1-x_2| + |1-x_2+x_3| + |x_2| \\ -|1-x_2+x_3| \end{bmatrix}$$

$$\frac{\partial F}{\partial x_3}(x) = 2\begin{bmatrix} -|1-x_1+x_3| \\ -|1-x_2+x_3| \\ |1-x_1+x_3| + |1-x_2+x_3| + |2-x_3| \end{bmatrix}$$

Let the initial guess be $x^{(0)} = [1.38, 1, 0.93]$. It can be verified that $F'(x^{(0)})$ is invertible. The constants in (1) and (2) are $\beta = 0.7 > \|F'(x^{(0)})^{-1}\|$ and $\eta = 0.005 > \|F'(x^{(0)})^{-1}\, F(x^{(0)})\|$. As for the constant in (iii), we can set $L = 8n = 32$ since the face cycle basis is used. This gives $\beta\eta L < 0.12$, so the convergence condition is satisfied, and the system converges with quadratic order starting at the given $x^{(0)}$.

4 Analysis of Hardy Cross algorithm

In order to simplify the calculation of the new solution $x^{(t)}$ in equation (5), a linear operator $H(x)$, typically associated with the Jacobian $F'(x)$, can be used in lieu of it:

$$x^{(t)} = x^{(t-1)} - H(x^{(t-1)})^{-1}\, F(x^{(t-1)}) \tag{8}$$

This variation of the Newton–Raphson method has been thoroughly explored in the literature. In the context of pipe network analysis, Algorithm HC-loop can be viewed as a simplified version of the Newton–Raphson method, using operator $H(x)$ defined as follows:

$$H(x) = \text{diag}\left[\, \frac{\partial f_c}{\partial x_c}(x) \,\right] \text{for } c = 1, 2, ..., k$$

where $\partial f_c/\partial x_c$ are the diagonal entries of the matrix $F'(x)$, defined in equation (6).

As the computation of $H(x)$ is straightforward, Algorithm HC-loop was designed to support manual analysis of small-scale water distribution systems. The process also begins

with a reference flow ψ that satisfies the flow conservation equation (1). It then iteratively produces new flow adjustments using equation (8) until the energy conservation equation (2) is satisfied, within some specified tolerance, for all cycles in the cycle basis used in the algorithm.

It is important to note that although equation (8) may give the impression that the adjustments for all cycles need to be computed simultaneously, they can be performed for each cycle $c \in C$ separately as follows:

$$\begin{aligned} x_c^{(t)} &= x_c^{(t-1)} - \left[\frac{\partial f_c}{\partial x_c}(x^{(t-1)}) \right]^{-1} f_c(x^{(t-1)}) \\ &= x_c^{(t-1)} - \frac{\sum_{e=1}^m A_{ec}\, |q_e^{(t-1)}| q_e^{(t-1)}}{2 \sum_{e=1}^m A_{ec}^2 |q_e^{(t-1)}|} \end{aligned}$$

where $q_e^{(t-1)} = \psi_e + \sum_{i=1}^k A_{ei}\, x_i^{(t-1)}$ is the flow value on edge e at step $t-1$.

The difference between Algorithm HC-loop and Algorithm NR-loop lies in the fact that in Algorithm NR-loop each adjustment takes into account the interaction between overlapping cycles, whereas in Algorithm HC-loop these adjustments are computed independently for each cycle.

Local convergence of variants of the Newton–Raphson algorithm that use a linear operator different from the Jacobian matrix was studied by (Rheinboldt (1968),Theorem 4.3). Below, we state his results, adapting it to functions in Euclidean spaces with a unique root.

Theorem 3 *Let $F : \mathbb{R}^k \mapsto \mathbb{R}^k$ be a differentiable function, $F'(x)$ be the $k \times k$ Jacobian matrix of $F(x)$, $H : \mathbb{R}^k \mapsto \mathbb{R}^k$ be a linear operator, and $x^{(0)} \in \mathbb{R}^k$ be an initial approximation of the iterative process (8) for $F(x) = \bar{0}$. Assume that:*

(i) $H(x^{(0)})$ *is invertible and* $\|H(x^{(0)})^{-1}\| \le \beta$
(ii) $\|H(x^{(0)})^{-1}\, F(x^{(0)})\| \le \eta$
(iii) $\|F'(x) - F'(y)\| \le L\|x - y\|$ *for* $x, y \in \mathbb{R}^k$
(iv) $\|H(x) - H(x^{(0)})\| \le K\|x - x^{(0)}\|$ *for* $x \in \mathbb{R}^k$
(v) $\|F'(x) - H(x)\| \le \delta_0 + \delta_1\|x - x^{(0)}\|$ *for* $x \in \mathbb{R}^k$

for some constants $\beta, \eta, L, K > 0$ and $\delta_0, \delta_1 \ge 0$.

With these assumptions, if $\beta\delta_0 < 1$ and $h = \beta\eta \max(L, K + \delta_1)/(1 - \beta\delta_0)^2 \le \frac{1}{2}$, then the sequence $\{x^{(t)}\}$ generated by equation (8) starting at $x^{(0)}$ is well-defined, contained in the ball $B(x^{(0)}, r)$ for $r = (1 - \sqrt{1 - 2h})(1 - \beta\delta_0)/(\beta \max(L, K + \delta_1))$, and converges to the unique solution x^* of $F(x) = \bar{0}$. If both δ_0, δ_1 are nonzero, then the order of convergence is linear with a factor $(\beta\delta_0 + \beta\delta_1 r)/(1 - \beta K r)$.

We apply Theorem 3 to Algorithm HC-loop in a similar manner to our use of Theorem 1 for Algorithm NR-loop. The bound in (iii) involving L remains consistent with those outlined in Sect. 3. For (i), the computation of the inverse of the diagonal matrix $H(x^{(0)})$ is straightforward:

$$\beta = \|H(x^{(0)})^{-1}\| = \max_{1 \le c \le k} \left[\frac{\partial f_c}{\partial x_c}(x^{(0)}) \right]^{-1}$$

Similarly, the bound in (ii) can be taken as $\eta = \|H(x^{(0)})^{-1}\, F(x^{(0)})\|$. For (iv) and (v), the estimates for K, δ_0, and δ_1 depend on the cycle basis C, similar to L. We show some general estimates of these constants in the following claims. Recall that $k = m - n + 1$ is the size of the cycle basis C and ℓ is its total length.

Claim 2 *The constants K, δ_0 and δ_1 in Theorem 3 can be bounded as follows:*

(a) $\|H(x) - H(x^{(0)})\| \le K\|x - x^{(0)}\|$, *where* $K = 2(\ell - k + 1)$.

(b) $\|F'(x) - H(x)\| \le \delta_0 + \delta_1 \|x - x^{(0)}\|$ *where* $\delta_0 = 2(\ell - k - 2)\,(\psi_{\max} + k\|x^{(0)}\|)$, $\delta_1 = 2k\,(\ell - k - 2)$, *and* $\psi_{\max} = \max\limits_{1\le e\le m} |\psi_e|$.

Proof (a) Here is the derivation of K:

$$\begin{aligned}
\|H(x) - H(x^{(0)})\| &= \max_{1\le c\le k} \Big| \frac{\partial f_c}{\partial x_c}(x) - \frac{\partial f_c}{\partial x_c}(x^{(0)}) \Big| \\
&= \max_{1\le c\le k} 2 \Big| \sum_{e=1}^{m} A_{ec}^2 \Big(|\psi_e + \sum_{i=1}^{k} A_{ei}\, x_i| - |\psi_e + \sum_{i=1}^{k} A_{ei}\, x_i^{(0)}| \Big) \Big| \\
&\le \max_{1\le c\le k} 2 \sum_{e=1}^{m} |A_{ec}| \sum_{i=1}^{k} |A_{ei}|\, \|x - x^{(0)}\| \\
&\le 2(\ell - k + 1)\, \|x - x^{(0)}\| \qquad (9)
\end{aligned}$$

because $\max_{1\le c\le k} \sum_{i=1}^{k} \sum_{e=1}^{m} |A_{ec}|\ |A_{ei}| \le \ell - k + 1$ (see the justification in the proof of Claim 1)

(b) Here is the derivation of δ_0 and δ_1:

$$\begin{aligned}
\|F'(x) - H(x)\| &= \max_{1\le c\le k} \sum_{d\ne c} \Big| \frac{\partial f_c}{\partial x_d}(x) \Big| \\
&= \max_{1\le c\le k} \sum_{d\ne c} 2 \Big| \sum_{e=1}^{m} A_{ec}\, A_{ed}\, |\psi_e + \sum_{i=1}^{k} A_{ei}\, x_i| \Big| \\
&\le \max_{1\le c\le k} 2 \sum_{d\ne c} \sum_{e=1}^{m} |A_{ec}|\, |A_{ed}| \big(\psi_{\max} + k\|x\| \big) \\
&\le 2(\ell - k - 2) \big(\psi_{\max} + k\|x^{(0)}\| + k\|x - x^{(0)}\| \big) \\
&\le 2(\ell - k - 2)\, (\psi_{\max} + k\|x^{(0)}\|\,) + 2k\,(\ell - k - 2)\, \|x - x^{(0)}\| \qquad (10)
\end{aligned}$$

where the right-hand side of (10) is bounded by observing that for each cycle $c \in C$:

$$\sum_{d\ne c} \sum_{e=1}^{m} |A_{ec}|\, |A_{ed}| = \sum_{d=1}^{k} \sum_{e=1}^{m} |A_{ec}|\, |A_{ed}| - \sum_{e=1}^{m} A_{ec}^2 \le \ell - k - 2\,.$$

Above, we use the fact that $\sum_{d=1}^{k} \sum_{e=1}^{m} |A_{ec}|\, |A_{ed}| \le \ell - k + 1$ (see the justification in the proof of Claim 1), and that $\sum_{e=1}^{m} A_{ec}^2 \ge 3$, because c has at least 3 edges. □

The following theorem states the sufficient condition for linear convergence of Algorithm HC-loop for loop equation $F(x) = \bar{0}$, a flow graph $G = (V, E)$ with a given cycle basis C of total length ℓ, a reference flow ψ with the maximum absolute flow value $\psi_{\max}$, and an initial solution $x^{(0)}$ for which $H(x^{(0)})$ is invertible:

Theorem 4 *(Convergence conditions of Algorithm HC-loop) Let* $\beta = \|H(x^{(0)})^{-1}\|$, $\eta = \|H(x^{(0)})^{-1} F(x^{(0)})\|$, $K = 2(\ell - k + 1)$, $L = 2k\,(\ell - k + 1)$, $\delta_0 = 2(\ell - k - 2)\,\big(\psi_{\max} + k\|x^{(0)}\|\big)$, *and* $\delta_1 = 2k\,(\ell - k - 2)$. *If* $\beta\delta_0 < 1$ *and* $\beta\eta \max(L, K + \delta_1)/(1 - \beta\delta_0)^2 \le \frac{1}{2}$, *then Algorithm HC-loop converges to the unique solution* x^* *of* $F(x) = \bar{0}$. *If both* δ_0, δ_1 *are non-zero, then the order of convergence is linear.*

Note that the condition on δ_0, δ_1 being non-zero will be true in non-degenerate scenarios, because $\delta_0 > 0$ as long as ψ is non-zero (which is the case when the total inflows are non-zero), and the only case when $\delta_1 = 0$ is when the network is just one cycle with three edges.

Similar to our analysis of Algorithm NR-loop in Sect. 3, the bounds on the constants in Theorem 3 can be improved for types of graphs that arise in practical networks, in particular for planar graphs.

Observation 2 *If G is planar and C is a face-based cycle basis then*

(a) $\|H(x) - H(x^{(0)})\| \le K\|x - y\|$ *for* $K = 4n$.
(b) $\|F'(x) - H(x)\| \le \delta_0 + \delta_1 \|x - x^{(0)}\|$ *where* $\delta_0 = 2n\big(\psi_{\max} + 2\|x^{(0)}\|\big)$, $\delta_1 = 4n$, *and* $\psi_{\max} = \max_{1\le e\le m} |\psi_e|$.

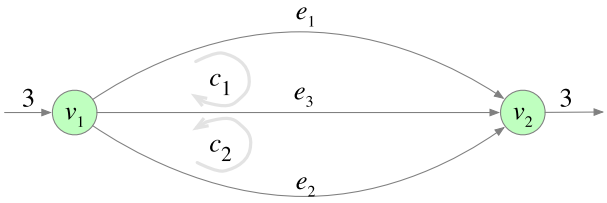
Proof Part (a) follows from the bound (9) in the proof of Claim 2(a), using the fact that $\sum_{i=1}^{k} |A_{ei}| \le 2$ for each edge e (because each edge belongs to at most two cycles in C) and that $\sum_{e=1}^{m} A_{ec}^2 \le n$.

For part (b), we modify the derivation in the proof of Claim 2(a), as follows:

$$\begin{aligned}
\|F'(x) - H(x)\| &\le \max_{1\le c\le k} \sum_{d\ne c} 2 \Big| \sum_{e=1}^{m} A_{ec}\, A_{ed}\, |\psi_e + \sum_{i=1}^{k} A_{ei}\, x_i| \Big| \\
&\le \max_{1\le c\le k} 2 \sum_{e=1}^{m} |A_{ec}| \sum_{d\ne c} |A_{ed}| \big(\psi_{\max} + 2\|x\|\big) \\
&\le 2n(\psi_{\max} + 2\|x\|) \\
&\le 2n\big(\psi_{\max} + 2\|x^{(0)}\|\big) + 4n\|x - x^{(0)}\|
\end{aligned}$$

The first two steps use inequalities $\sum_{i=1}^{k} |A_{ei}| \le 2$ and $\sum_{d\ne c} |A_{ed}| \le 1$ (that hold because every edge e belongs to at most 2 cycles) and $\sum_{e=1}^{m} |A_{ec}| \le n$. □

Lower bound on convergence We now demonstrate that, in general, the guaranteed local convergence rate of Algorithm HC-loop is not better than linear. To illustrate this, consider the flow graph G with 2 vertices, 3 edges, 1 inlet and 1 outlet. The cycle basis consists of two cycles, whose clockwise directions are shown in the figure below.



Let the reference flow be $\psi = [0, 0, 3]$. The loop equations are:

$$\begin{aligned}
f_1(x) &= |x_1|x_1 - |3 - x_1 - x_2|(3 - x_1 - x_2) = 0 \\
f_2(x) &= |x_2|x_2 - |3 - x_1 - x_2|(3 - x_1 - x_2) = 0
\end{aligned}$$

The solution is $x_1^* = x_2^* = 1$. We are interested in the case where the initial adjustment $x^{(0)}$ is sufficiently close to the solution, particularly when $x_1^{(0)} = x_2^{(0)} = 1 + \epsilon$ for some

$0 < |\epsilon| \ll 1$. The first adjustment is:

$$x_1^{(1)} = x_2^{(1)} = 1 + \epsilon - \frac{(1+\epsilon)^2 - (3-2-2\epsilon)^2}{2(1+\epsilon+3-2-2\epsilon)} = 1 + \epsilon - \frac{3\epsilon\,(2\,-\,\epsilon)}{2(2\,-\,\epsilon)} = 1\,+\,\epsilon/2$$

Thus, $\| x^{(1)} - x^* \| = \epsilon/2 = \frac{1}{2} \| x^{(0)} - x^* \|$. The convergence order is linear with convergence rate $\frac{1}{2}$.

5 Conclusion

We expressed the pipe network analysis problem as a root-finding problem for a system of non-linear equations and we discussed the local convergence conditions for Algorithms NR-loop and HC-loop, namely the loop variants of the Newton–Raphson and HardyCross algorithms. We showed that Algorithm NR-loop achieves quadratic local convergence, while HC-loop ensures only linear local convergence. These findings align with experimental results in the literature. Additionally, we demonstrated that in the important case of planar graphs, the convergence conditions for both algorithms can be strengthened by using the face cycle bases.

Appendix A Errors in Altman and Boulos (1995)

Altman and Boulos (1995) attempted to provide local convergence conditions of Algorithm NR-loop and Algorithm NR-flow for solving flow equations. However, their analysis has two errors that make their bounds on convergence invalid. The first one is an algebraic mistake when substituting the variables in (Altman and Boulos (1995), eq. (13)):

$$\begin{aligned} x_1 &= \eta_{11}x_{t+1} + \eta_{12}x_{t+2} + \cdots + \eta_{1t}x_e + a_1 \\ &\vdots \\ x_t &= \eta_{t1}x_{t+1} + \eta_{t2}x_{t+2} + \cdots + \eta_{tt}x_e + a_t \end{aligned}$$

into (Altman and Boulos (1995), eq. (12)):

$$\begin{aligned} \gamma_{11}\xi_1 x_1^2 + \gamma_{12}\xi_2 x_2^2 + \cdots + \gamma_{1e}\xi_e x_e^2 + \Phi_1 &= 0 \\ &\vdots \\ \gamma_{m1}\xi_1 x_1^2 + \gamma_{m2}\xi_2 x_2^2 + \cdots + \gamma_{me}\xi_e x_e^2 + \Phi_m &= 0 \end{aligned}$$

(The indices in the last equation in Altman and Boulos (1995) were also incorrectly given as n instead of m. We corrected these above.) With a change of variables from $x_{t+1}, x_{t+2}, \cdots, x_e$ to $x_1, x_2, \cdots, x_l$, they obtained (Altman and Boulos (1995),eq. (14)):

$$\begin{aligned} f_1 &= p_{11}x_1^2 + p_{12}x_2^2 + \cdots + p_{1l}x_l^2 + b_{11}x_1 + b_{12}x_2 + \cdots + b_{1l}x_l + d_1 = 0 \\ &\vdots \\ f_l &= p_{l1}x_1^2 + p_{l2}x_2^2 + \cdots + p_{1l}x_l^2 + b_{l1}x_1 + b_{l2}x_2 + \cdots + b_{ll}x_l + d_l = 0 \end{aligned}$$

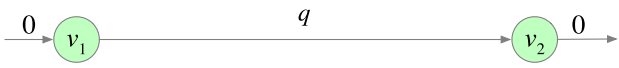
All the terms $x_i x_j$ where $i \neq j$ do not appear in the above equations which affect the computation of the partial derivatives $\partial f_i / \partial x_j$ later on in their analysis.

The other mistake involves the sign issue in the energy equation for each cycle. The authors used a fixed indicator γ_{wu} for the orientation of flow in cycle w in (Altman and Boulos (1995), eq. (6)):

$$\Phi_w + \sum_{u=1}^{e} \gamma_{wu} \xi_u Q_u^2 = 0$$

This would be true if all flow directions remained the same; in other words, if the sign of each Q_u was constant through the whole iterative process. However, it is not guaranteed. Thus, when Q_u flips sign, γ_{wu} needs to be changed accordingly for the energy equation to be valid.

Appendix B An example for non-convergence of algorithm NR-node



Shamir mentioned the issue with non-convergence encountered in certain systems in Shamir (1968). We provide a simple flow graph in which Algorithm NR-node enters an infinite loop. Using a similar setup for the node method as in Shamir (1968), the unknowns are pressures p_v for nodes $v = 1, 2, \cdots, n$, and the system of equations includes flow conservation equations for each node of the form:

$$f_v(p_1, p_2, \cdots, p_n) = w_v + \sum_{e=(u,v)\in E} D_{ev}\, q_e = w_v + \sum_{e=(u,v)\in E} D_{ev} \frac{p_u - p_v}{\sqrt{|p_u - p_v|}} = 0 \tag{B1}$$

The below example demonstrates an oscillation case of Algorithm HR-node for solving the described node method equation. Consider the simple flow graph G with only two vertices v_1 and v_2 connected by a single edge as shown in the figure below. Set the external flow on the two vertices to be 0 and the corresponding pressures to be $p_1 = x \neq 0$ and $p_2 = 0$.

The equation (B1) has the form $f(x) = -x/\sqrt{|x|} = 0$. The iteration step is: $x - f(x)/f'(x) = x - x \cdot 2\sqrt{|x|}/\sqrt{|x|} = -x$. So for any initial value $x \neq 0$, the algorithm will oscillate between x and $-x$.

Acknowledgements Research supported by NSF grant CCF-1536026.

Funding This research was partially supported by NSF grants CCF-1536026 and CCF-2153723.

Declarations

Conflict of interest The authors have no Conflict of interest to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Adams AT (1955) Flow analysis of distribution systems containing elevated reservoirs by the Hardy Cross method. PhD thesis, Georgia Institute of Technology

Altman T, Boulos PF (1995) Convergence of Newton method in nonlinear network analysis. Math Comput Modell 21(4):35–41. https://doi.org/10.1016/0895-7177(95)00004-L

Boulos PF, Lansey KE, Karney BW (2006) Comprehensive water distribution systems analysis handbook for engineers and planners. MWH Soft, Arcadia

Brkić D (2009) An improvement of Hardy Cross method applied on looped spatial natural gas distribution networks. Appl Energy 86:1290–1300. https://doi.org/10.1016/j.apenergy.2008.10.005

Cross H (1936). Analysis of flow in networks of conduits or conductors. Engineering Experiment Station **Bulletin No 286**

Gameiro Lopes A (2004) Implementation of the Hardy-Cross method for the solution of piping networks. Comput Appl Eng Educ 12:117–125. https://doi.org/10.1002/cae.20006

Jay L (2001) A note on q-order of convergence. BIT Numer Math 41:422–429. https://doi.org/10.1023/A:1021902825707

Kantorovich ILV (1964) Functional analysis in normed spaces, Moscow (1959). Translated by D. Brown, Pergamon, New York

Kantorovich ILV (1948) Functional analysis and applied mathematics. Uspehi Mat Nauk 3:89–185

Kavitha T, Liebchen C, Mehlhorn K, Michail D, Rizzi R, Ueckerdt T, Zweig KA (2009) Survey: cycle bases in graphs characterization, algorithms, complexity, and applications. Comput Sci Rev 3(4):199–243. https://doi.org/10.1016/j.cosrev.2009.08.001

Nielsen HB (1989) Methods for analyzing pipe networks. J Hydraul Eng 115(2):139–157. https://doi.org/10.1061/(ASCE)0733-9429(1989)115:2(139)

Ortega JM (1968) The Newton-Kantorovich theorem. Am Math Mon 75(6):658–660

Rheinboldt WC (1968) A unified convergence theory for a class of iterative processes. SIAM J Numer Anal 5(1):42–63

Rizzi R (2009) Minimum weakly fundamental cycle bases are hard to find. Algorithmica 53(3):402–424

Shamir U (1968) Water distribution system analysis. J Hydraulic Div 94

Shamir U, Howard C (1977) Engineering analysis of water-distribution systems. J Am Water Works Assoc 69:510–514. https://doi.org/10.1002/j.1551-8833.1977.tb06802.x

Singh MK, Kekatos V (2021) On the flow problem in water distribution networks: uniqueness and solvers. IEEE Trans Control Netw Syst 8(1):462–474. https://doi.org/10.1109/TCNS.2020.3029150

Srinivasan S, Sundar K, Gyrya V, Zlotnik A (2023) Numerical solution of the steady-state network flow equations for a nonideal gas. IEEE Trans Control Netw Syst 10(3):1449–1461. https://doi.org/10.1109/TCNS.2022.3232524

Stephenson DJ (1984) Pipeflow Analysis. ISSN. Elsevier Science, Amsterdam

Wood DJ (1981) Algorithms for pipe network analysis and their reliability. Technical Report 127, Kentucky Water Resources Research Center

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.