Turkish Journal of Mathematics

Volume 45 | Number 1 Article 16

1-1-2021

A new Gauss--Newton-like method for nonlinear equations

HAIJUN WANG

QI WANG

Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2021) 45: 264 – 280

doi:10.3906/mat-1912-68

A new Gauss–Newton-like method for nonlinear equations

Haijun WANG[1,*], **Qi WANG**[1,2]
[1]School of Mathematics, China University of Mining and Technology, Xuzhou, P.R. China,
[2]Wuxi Machinery and Electron Higher Professional and Technical School, Wuxi, P.R. China

Received: 18.12.2019 • **Accepted/Published Online:** 23.11.2020 • **Final Version:** 21.01.2021

Abstract: In this paper, a new Gauss–Newton-like method that is based on a rational approximation model with linear numerator is proposed for solving nonlinear equations. The new method revises the $J_k^{\mathrm{T}} J_k$ matrix by a rank-one matrix at each iteration. Furthermore, we design a new iterative algorithm for nonlinear equations and prove that it is locally q-quadratically convergent. The numerical results show that the new proposed method has better performance than the classical Gauss–Newton method.

Key words: Rational approximation model, Gauss–Newton method, nonlinear equations, local convergence

1. Introduction

We consider a system of nonlinear equations

$$F(x) = 0, \tag{1.1}$$

where $F : \mathrm{D} \subset \mathrm{R}^n \to \mathrm{R}^m$ is F-differentiable on an open convex domain neighborhood D, $F = (F_1, \ldots, F_m)^{\mathrm{T}}$, and $x_* \in \mathrm{D}$ exists for which $F(x_*) = 0$. In this paper, we assume all $\|\cdot\|$ refers to the 2-norm in all cases.

Many relationships in nature are inherently nonlinear in that effects are not in direct proportion to their cause. Typically, the behavior of a nonlinear system is described in mathematics by a nonlinear system of equations. Accordingly, solving nonlinear equations occurs frequently in scientific work. Solving systems of nonlinear equations has now become one of the most important problems in numerical analysis. Many robust and efficient methods for solving nonlinear equations are brought forward. A method for the solution of certain nonlinear equations problem in least squares was discussed in [9]. A maximum neighborhood method was developed for least squares estimation in [10]. The quadratic rate of convergence of the Levenberg–Marquardt method for solving a system of nonlinear equations was discussed in [17] and [6]. Based on the line search for the approximate LM step and extending the LM parameter to more general cases, an accelerated version of the modified Levenberg–Marquardt method (AMLM) for nonlinear equations was proposed in [7]. In [2], a modified two-step Levenberg–Marquardt method for nonlinear equations was proposed by introducing an adaptive LM parameter for AMLM algorithm. In[8], a modified quasi-Newton method was proposed for solving the nonlinear equation based on the new quasi-Newton equation. In[13], a new iterative scheme for solving nonlinear equations was proposed based on a rational approximation model. In [14], a Gauss–Newton-like method for nonlinear least

*Correspondence: wanghj@cumt.edu.cn
2010 *AMS Mathematics Subject Classification:* 90C30, 65H05

squares with equality constraints-local convergence and applications was discussed. The authors in [1] focused on locating a solution of a nonlinear least squares problem by using an inexact Gauss–Newton method. The authors in [3] presented a derivative-free version of the Gauss–Newton(DFO-GN) method for solving nonlinear least squares problems. There are many other methods we have not listed here (see [4]).

The Newton method for nonlinear equations is an important and basic method, which converges quadratically. The linear expansion of nonlinear system (1.1) at an iteration point x_k is

$$F(x_k) + J(x_k)(x - x_k) = 0, \tag{1.2}$$

where $J(x_k)$ is the Jacobian matrix of $F(x)$ at x_k.

If $m = n$ and $J(x_k)$ is nonsingular, we can obtain the following classical Newton iteration:

$$x_{k+1} = x_k - J(x_k)^{-1}F(x_k), \quad k = 1, 2, \cdots. \tag{1.3}$$

If $m \neq n$ or $J(x_k)$ is singular, then Newton iteration (1.3) will fail. By applying the least square idea, the Gauss–Newton method can effectively solve the above problems. The Gauss–Newton method can be viewed as a modification of the Newton method with line search.

Considering the following problem

$$\min_{x \in \mathrm{R}^n} \Phi(x) \tag{1.4}$$

where $\Phi(x) = \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}F(x)^{\mathrm{T}}F(x)$ and the first derivative of $\Phi(x)$ is

$$\nabla\Phi(x) = J(x)^{\mathrm{T}}F(x).$$

Similarly, the second derivative (or Hessian matrix of $\Phi(x)$) is

$$\nabla^2\Phi(x) = J(x)^{\mathrm{T}}J(x) + S(x)$$

where

$$S(x) = \sum_{i=1}^{m} F_i(x)\nabla^2 F_i(x)$$

denotes the second-order information in $\nabla^2\Phi(x)$. Thus, the quadratic model of $\Phi(x)$ around x_k is

$$\begin{aligned} m_k(x) &= \Phi(x_k) + \nabla\Phi(x)(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}}\nabla^2\Phi(x_k)(x - x_k) \\ &= \frac{1}{2}F(x_k)^{\mathrm{T}}F(x_k) + J(x_k)^{\mathrm{T}}F(x_k)(x - x_k) \\ &\quad + \frac{1}{2}(x - x_k)^{\mathrm{T}}((J(x_k)^{\mathrm{T}}J(x_k) + S(x_k))(x - x_k), \end{aligned} \tag{1.5}$$

the specialization of the Taylor series quadratic model (1.5) for minimization to objective functions of form (1.4). Thus, the problem (1.4) can be approximated to the following form:

$$\min_{x \in \mathrm{R}^n} m_k(x). \tag{1.6}$$

According to (1.5), the Newton method applied to (1.6) is

$$x_{k+1} = x_k - (J(x_k)^{\mathrm{T}} J(x_k) + S(x_k))^{-1} J(x_k)^{\mathrm{T}} F(x_k). \tag{1.7}$$

Certainly (1.7) would be a fast local method for the problem (1.4), since it is locally q-quadratically convergent under standard assumptions. The problem with the full Newton approach is that $S(x)$ is usually either unavailable or inconvenient to obtain, and it is too expensive to approximate by finite difference. To simplify the calculation and get an efficient algorithm, we ignore $S(x)$ in the quadratic model of the objective function and use the approximation

$$\nabla^2 \Phi(x_k) \approx J(x_k)^{\mathrm{T}} J(x_k),$$

then we obtain the classical Gauss-Newton iteration

$$x_{k+1} = x_k - (J(x_k)^{\mathrm{T}} J(x_k))^{-1} J(x_k)^{\mathrm{T}} F(x_k). \tag{1.8}$$

The classical Gauss–Newton iteration is locally q-quadratically convergent for a zero-residual problem (problem (1.4) for which $F(x_*) = 0$ is called a zero-residual problem, where $x_* \in \mathrm{D}$ is a solution of problem (1.1)), and the sequence generated by the classical Gauss–Newton method (1.8) converges to x_* (see [4] Theorem 10.2.1 and Corollary 10.2.2). However, the iteration (1.8) is not well-defined if $J(x_k)$ does not have full column rank, and the iteration (1.8) is not locally convergent on problems that are very nonlinear.

In this paper, for above reasons, we proposed a new Gauss–Newton-like method for solving nonlinear equations (1.1) to improve the computational efficiency of the classical Gauss–Newton method. Inspired by the work of Wang in [16], Sui in [12], and an improved method by Sayheya in [13], we propose a new Gauss–Newton-like iterative model and design the corresponding algorithm. Compared with the classical Gauss–Newton method with numerical examples, the results show that new Gauss–Newton-like method is more effective.

This paper is organized as follows. In the next section, we give a new Gauss–Newton-like method to solve nonlinear equations (1.1). In Section 3, we will analyze the convergence of the new Gauss–Newton-like method. In Section 4, numerical examples will be further considered to show that the new method may be more effective and practicable than the classical Gauss–Newton method. Some conclusions are given in the last section.

2. Description of the new Gauss–Newton method

In this section, we will suggest a new Gauss–Newton-like method for solving a system of nonlinear equations (1.1). Above all, we consider a class of rational approximate function (see [16]) of $F_i(x)(i = 1, \cdots, m)$ as follows:

$$Q_i(x, a_k) = F_i(x_k) + \frac{F_i'(x_k)^{\mathrm{T}}(x - x_k)}{1 + a_k^{\mathrm{T}}(x - x_k)}, \qquad i = 1, \cdots, m$$

where $a_k \in \mathrm{R}^n$ is the undetermined vector and $x_k \in \mathrm{R}^n$ is the current point. With some calculations, we can deduce that the function $Q_i(x, a_k)$ is a first-order approximation of $F_i(x)$, i.e.

$$Q_i(x_k, a_k) = F_i(x_k), \quad i = 1, \cdots, m,$$

$$Q_i'(x_k, a_k) = F_i'(x_k), \quad i = 1, \cdots, m.$$

The above rational function has many excellent properties. For example, it can reflect more curvature information than the classical linear approximation model. This property may be able to reduce the number of iterations when using an iteration method that was constructed by rational approximate functions to solve the system of nonlinear equations (1.1).

Taking into account the interpolation condition

$$Q_i'(x_{k-1}, a_k) = F_i'(x_{k-1}),$$

then we obtain

$$a_k = \pm\frac{1}{c_0}\left(\sqrt{\frac{c_0}{c_1}}(F_i'(x_k) - F_i'(x_{k-1}))\right),$$

where $c_0 = F_i(x_{k-1})^{\mathrm{T}}(x_{k-1} - x_k)$, $c_1 = F_i(x_k)^{\mathrm{T}}(x_{k-1} - x_k)$, and $x_{k-1} \in \mathrm{R}^n$ is the preceding point of current point x_k.

Thus, we can get the following rational function (see [12]):

$$Q_i(x, a_k) = F_i(x_k) + \frac{F_i'(x_k)^{\mathrm{T}}(x - x_k)}{1 \pm \frac{1}{c_0}\left(\sqrt{\frac{c_0}{c_1}}(F_i'(x_k) - F_i'(x_{k-1}))^{\mathrm{T}}\right)(x - x_k)}, \quad i = 1, \cdots, m.$$

Each approximate function $Q_i(x, a_k)$ of nonlinear function $F_i(x)(i = 1, \cdots, m)$ at the current iteration point x_k contains a different vector $a_k = \pm\frac{1}{c_0}\left(\sqrt{\frac{c_0}{c_1}}(F_i'(x_k) - F_i'(x_{k-1}))\right)(i = 1, \cdots, m)$, which makes it more complex when we use an iteration method constructed by rational approximate $Q_i(x, a_k) = 0 \ \ (i = 1, \cdots, m)$ to solve the system of nonlinear equations (1.1).

In order to avoid the above disadvantage, Saheya (in [13]) chose the same a_k for all nonlinear functions $Q_i(x, a_k)(i = 1, \cdots, m)$ at x_k, let $Q(x) = (Q_1(x, a_k), \cdots, Q_m(x, a_k))^{\mathrm{T}}$ and

$$Q(x) = F(x_k) + \frac{J(x_k)(x - x_k)}{1 + a_k^{\mathrm{T}}(x - x_k)} \tag{2.1}$$

to approximate the nonlinear functions $F(x)$ and take into account

$$F(x) \approx Q(x) = 0.$$

If $m = n$ in (1.1) and the matrix $J(x_k) + F(x_k)a_k^{\mathrm{T}}$ is invertible, in [13], Saheya obtained the following iterative formula:

$$x_{k+1} = x_k - \left(J(x_k) + F(x_k)a_k^{\mathrm{T}}\right)^{-1} F(x_k) \tag{2.2}$$

to estimate the root of nonlinear equations $F(x) = 0$.

However, the iterative formula (2.2) may fail to solve the system of nonlinear equations (1.1) with $m \neq n$. Here, we extend the rational approximation model to the Gauss–Newton method. Let $F(x) = Q(x)$, according to (2.1), we have

$$\begin{aligned} \Phi(x) &= \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}F(x)^{\mathrm{T}}F(x) \\ &= \frac{1}{2}F(x_k)^{\mathrm{T}}F(x_k) + \frac{J(x_k)^{\mathrm{T}}F(x_k)(x - x_k)}{1 + a_k^{\mathrm{T}}(x - x_k)} + \frac{(x - x_k)^{\mathrm{T}}J(x_k)^{\mathrm{T}}J(x_k)(x - x_k)}{2(1 + a_k^{\mathrm{T}}(x - x_k))^2} \end{aligned} \tag{2.3}$$

and the first derivative of $\Phi(x)$ is

$$\nabla\Phi(x) = \frac{1}{1+a_k^{\mathrm{T}}(x-x_k)}\left(I + \frac{a_k(x-x_k)^{\mathrm{T}}}{1+a_k^{\mathrm{T}}(x-x_k)}\right)\left(J(x_k)^{\mathrm{T}}F(x_k) - \frac{J(x_k)^{\mathrm{T}}J(x_k)(x-x_k)}{1+a_k^{\mathrm{T}}(x-x_k)}\right). \tag{2.4}$$

Let $\nabla\Phi(x) = 0$, then we have

$$J(x_k)^{\mathrm{T}}F(x_k) = \frac{J(x_k)^{\mathrm{T}}J(x_k)(x-x_k)}{1+a_k^{\mathrm{T}}(x-x_k)},$$

$$\left(J(x_k)^{T}J(x_k) + J(x_k)^{\mathrm{T}}F(x_k)a_k^{\mathrm{T}}\right)(x-x_k) = -J(x_k)^{\mathrm{T}}F(x_k)$$

and further obtain the following new Gauss–Newton iterative formula

$$x_{k+1} = x_k - \left(J(x_k)^{T}J(x_k) + J(x_k)^{\mathrm{T}}F(x_k)a_k^{\mathrm{T}}\right)^{-1}J(x_k)^{\mathrm{T}}F(x_k). \tag{2.5}$$

It is easy to see that the iteration (2.5) is reduced to the classical Gauss–Newton iteration if $a_k = \mathbf{0}$.

In the following, we will determine the vector a_k in (2.5). Here, we use additional interpolation condition

$$Q(x_{k-1}) = F(x_{k-1}) \tag{2.6}$$

to update a_{k-1} into a_k.

From (2.6), we obtain

$$(1+a_k^{\mathrm{T}}(x_{k-1}-x_k))(F(x_{k-1}-F(x_k))) = J(x_k)(x_{k-1}-x_k). \tag{2.7}$$

Furthermore, there exists vector $z \in \mathrm{R}^m$ and $z \neq \mathbf{0}$ satisfy

$$1 + a_k^{\mathrm{T}}(x_{k-1}-x_k) = \frac{z_k^{\mathrm{T}}J(x_k)(x_{k-1}-x_k)}{z_k^{\mathrm{T}}(F(x_{k-1})-F(x_k))}. \tag{2.8}$$

Let $s_{k-1} = x_k - x_{k-1}, F(x_k) = F_k, J(x_k) = J_k, y_{k-1} = F_k - F_{k-1}$, then

$$\begin{aligned} a_k^{\mathrm{T}}s_{k-1} &= 1 - \frac{z_k^{\mathrm{T}}J_ks_{k-1}}{z_k^{\mathrm{T}}y_{k-1}} \\ &= \frac{z_k^{\mathrm{T}}y_{k-1} - z_k^{\mathrm{T}}J_ks_{k-1}}{z_k^{\mathrm{T}}y_{k-1}}. \end{aligned} \tag{2.9}$$

Therefore, let

$$a_k = \frac{z_k^{\mathrm{T}}(y_{k-1}-J_ks_{k-1})p_k}{s_{k-1}^{\mathrm{T}}p_k \cdot z_k^{\mathrm{T}}y_{k-1}}, \tag{2.10}$$

where $p_k \in \mathrm{R}^n, s_{k-1}^{\mathrm{T}}p_k \neq 0$. It is easy to see that a_k satisfies (2.7).

We can choose different $z_k \in \mathrm{R}^m$ and $p_k \in \mathrm{R}^n$ in (2.10). For simplicity, here we choose $z_k = y_{k-1}, p_k = s_{k-1}$, then we obtain

$$J_k^{\mathrm{T}}F_ka_k^{\mathrm{T}} = \frac{y_{k-1}^{\mathrm{T}}(y_{k-1}-J_ks_{k-1})}{s_{k-1}^{\mathrm{T}}s_{k-1}} \times \frac{J_k^{\mathrm{T}}F_ks_{k-1}^{\mathrm{T}}}{y_{k-1}^{\mathrm{T}}y_{k-1}}. \tag{2.11}$$

Now, based on the above results, we design the following new Gauss–Newton-like algorithm.

Algorithm 2.1 New Gauss–Newton-like method (**NGNL**)

Step 0. Start with an initial point $x_0 \in \mathrm{R}^n$ and set the iteration number as $k = 0$.

Step 1. Let $a_0 = 0$, $\varepsilon = \varepsilon_0, B_0 = 0$ and compute the value of the function $F(x_k)$ at point x_k.

Step 2. If $\|F(x_k)\| \leq \varepsilon$, then stop.

Step 3. Compute the gradient of the function J_k at point x_k, and set

$$\left(J_k^{\mathrm{T}} J_k + B_k\right) s_k = -J_k F_k,$$
$$x_{k+1} = x_k + s_k.$$

Step 4.Update the matrix B_k

$$B_k = \frac{y_{k-1}^{\mathrm{T}}(y_{k-1} - J_k s_{k-1})}{s_{k-1}^{\mathrm{T}} s_{k-1}} \times \frac{J_k^{\mathrm{T}} F_k s_{k-1}^{\mathrm{T}}}{y_{k-1}^{\mathrm{T}} y_{k-1}}.$$

Step 5. Set the new iteration number as $k = k + 1$ and go to step 2.

There are two differences between the new Gauss–Newton-like method (NGNL) and the classical Gauss–Newton method (CGN). Firstly, NGNL uses rational approximate function to obtain new iterative formula. Secondly, NGNL utilizes the function values of the previous iteration point to revise $J_k^{\mathrm{T}} J_k$ in every iteration.

3. Convergence analysis

In this section, we prove the convergence of Algorithm 2.1. Due to the need of proof, we first make the following assumptions.

Aassumption 3.1

(1) *Assume that $J(x)$ is Lipschitz continuous in $\mathrm{D} \subset \mathrm{R}^n$ with $||J(x)|| \leq \alpha$ for all $x \in \mathrm{D}$, and there exists a constant $L > 0$, such that for all $x, y \in \mathrm{D}$*

$$\|J(x)^T J(x) - J(y)^{\mathrm{T}} J(y)\| \leq L\|x - y\|.$$

(2) *The function F is continuously differentiable in the open convex set $\mathrm{D} \subset \mathrm{R}^n$, and there exists a constant $\gamma > 0$, such that for all $x, y \in \mathrm{D}$*

$$\|J(x) - J(y)\| \leq \gamma\|x - y\|.$$

(3) *Let x_* be the local minimal point of the least squares problem (1.4) (or x_* is the root of problem (1.1)), $J(x_*)^{\mathrm{T}} J(x_*)$ is nonsingular, and there exists a constant $\mu > 0$, such that*

$$\|(J(x_*)^{\mathrm{T}} J(x_*))^{-1}\| \leq \mu.$$

In order to facilitate the following proof of convergence theorem, here we introduce the following lemmas.

Lemma 3.1 [3] *Let $F:\mathrm{R}^n\to\mathrm{R}^m$ satisfy the condition (2) of assumption 3.1, then for all $x+s\in\mathrm{D}$,*

$$\|F(x+s)-F(x)-J(x)s\|\le\frac{\gamma}{2}\|s\|^2. \tag{3.1}$$

Lemma 3.2 [3] *Under the condition of Lemma 3.1, then there exist $\varepsilon>0$ and $0<m<M$, such that*

$$m\|v-u\|\le\|F(v)-F(u)\|\le M\|v-u\|, \tag{3.2}$$

for all $v,u\in\mathrm{D}$ for which $\max\{\|v-x_*\|,\|u-x_*\|\}\le\varepsilon$.

Lemma 3.3 [3] *Let $\|\bullet\|$ be any norm on $\mathrm{R}^{n\times n}$ that obeys the consistency condition and $\|I\|=1$ and let $\|E\|<1$, then $(I-E)^{-1}$ exists and*

$$\|(I-E)^{-1}\|\le\frac{1}{1-\|E\|}.$$

If A is nonsingular and $\|A^{-1}(B-A)\|<1$, then B is nonsingular and

$$\|B^{-1}\|\le\frac{\|A^{-1}\|}{1-\|A^{-1}(B-A)\|}.$$

With the help of above lemmas, we proposed the convergence theorem for the new Gauss–Newton-like method. Here, we denote the ε-neighborhood of x_* by

$$N(x_*,\varepsilon)=\{x\mid\|x-x_*\|\le\varepsilon,\forall x\in\mathrm{R^n}\}.$$

Theorem 3.4 *Let $F:\mathrm{R}^n\to\mathrm{R}^m$ be twice continuously differentiable in an open convex set $\mathrm{D}\subset\mathrm{R}^n$, under the previous assumptions, assume that $J(x)$ is Lipschitz continuous in $\mathrm{D}\subset\mathrm{R}^n$ with $||J(x)||\le\alpha$ for all $x\in\mathrm{D}$ and there exist $x_*\in\mathrm{R}^n$, $r>0$, such that $N(x_*,r)\subset\mathrm{D}$ and $F(x_*)=0$. Then there exist $\varepsilon>0$ such that for all $x_0\in N(x_*,\varepsilon)$ the sequence $\{x_k\}_{k=2}^{\infty}$ generated by Algorithm 2.1 is well-defined, converges to x_*, and obeys*

$$\|x_{k+1}-x^*\|\le\mu\alpha\gamma\left(1+\frac{M}{m}\right)\|x_k-x^*\|^2,\quad k=1,2,\ldots. \tag{3.3}$$

and

$$\|x_{k+1}-x_*\|<q^{k+1}\|x_0-x_*\|,\quad k=1,2,\ldots. \tag{3.4}$$

where $0<q<1$

Proof when $a_0=0$, the iterative formula (2.5) is reduced to the classical Gauss–Newton method. It can be seen from the proof of the classical Gauss–Newton method that

$$\|x_1-x^*\|\le q\|x_0-x_*\|, \tag{3.5}$$

where $0<q<1$.

Let

$$\varepsilon=\min\left\{r,\frac{m}{\mu q(2mL+\alpha\gamma M)},\frac{m}{\mu\alpha\gamma(m+M)}\right\},$$

then

$$\begin{aligned}
&\|(J_*^{\mathrm{T}}J_*)^{-1}((J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})-J_*^{\mathrm{T}}J_*)\|\\
&\le \|(J_*^{\mathrm{T}}J_*)^{-1}\|(\|J_1^{\mathrm{T}}J_1-J_*^{\mathrm{T}}J_*\|+\|J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}}\|)\\
&\le \mu(L\|x_1-x_*\|+\|J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}}\|).
\end{aligned}$$

Considering the second part of the above expression $\|J_1^TF_1a_1^T\|$, from (3.1) and (3.2) we can obtain

$$\begin{aligned}
\|J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}}\| &\le \|J_1\|\times\|F_1\|\times\|a_1\|\\
&\le \alpha\|F_1\|\times\left\|\frac{y_0^{\mathrm{T}}(y_0-J_1s_0)s_0^{\mathrm{T}}}{s_0^{\mathrm{T}}s_0y_0^{\mathrm{T}}y_0}\right\|\\
&= \alpha\|F_1-F_*\|\times\left\|\frac{y_0^{\mathrm{T}}}{y_0^{\mathrm{T}}y_0}\times\frac{(F_1-F_0-J_1s_0)s_0^{\mathrm{T}}}{s_0^{\mathrm{T}}s_0}\right\|\\
&\le \alpha M\|x_1-x_*\|\times\frac{1}{\|y_0\|}\times\left\|\frac{(F_1-F_0-J_1s_0)s_0^{\mathrm{T}}}{s_0^{\mathrm{T}}s_0}\right\|\\
&\le \alpha M\|x_1-x_*\|\times\frac{1}{\|F_1-F_0\|}\times\frac{\gamma}{2}\|s_0\|\\
&\le \alpha M\|x_1-x_*\|\times\frac{\gamma}{2m}\\
&\le \frac{\alpha\gamma M}{2m}\|x_1-x_*\|.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|(J_*^{\mathrm{T}}J_*)^{-1}\left((J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})-J_*^{\mathrm{T}}J_*\right)\| &\le \mu\left(L+\frac{\alpha\gamma M}{2m}\right)\|x_1-x_*\|\\
&\le \mu q\left(L+\frac{\alpha\gamma M}{2m}\right)\|x_0-x_*\|\\
&\le \mu q\left(L+\frac{\alpha\gamma M}{2m}\right)\varepsilon\\
&\le \frac{1}{2}.
\end{aligned}$$

According to the perturbation Lemma 3.3, we know that $J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}}$ is reversible, and

$$\begin{aligned}
\|(J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})^{-1}\| &= \frac{\|(J_*^{\mathrm{T}}J_*)^{-1}\|}{1-\|(J_*^{\mathrm{T}}J_*)^{-1}\left((J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})-J_*^{\mathrm{T}}J_*\right)\|}\\
&\le 2\|(J_*^{\mathrm{T}}J_*)^{-1}\|\le 2\mu,
\end{aligned}$$

which shows that x_2 is well defined. From (2.5) we obtain

$$\begin{aligned}
x_2-x_* &= x_1-x_*-(J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})^{-1}J_1^{\mathrm{T}}F_1\\
&= x_1-x_*-(J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})^{-1}(J_1^{\mathrm{T}}F_1-J_1^{\mathrm{T}}F_*)\\
&= (J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})^{-1}\left(J_1^{\mathrm{T}}F_*-J_1^{\mathrm{T}}F_1-(J_1^{\mathrm{T}}J_1+J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})(x_*-x_1)\right).
\end{aligned}$$

Thus,

$$\begin{aligned}\|x_2 - x_*\| &\le \|(J_1^{\mathrm{T}}J_1 + J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})^{-1}\| \times \|J_1^{\mathrm{T}}F_* - J_1^{\mathrm{T}}F_1 - (J_1^{\mathrm{T}}J_1 + J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}})(x_* - x_1)\| \\ &\le 2\mu(\|J_1^{\mathrm{T}}F_* - J_1^{\mathrm{T}}F_1 - J_1^{\mathrm{T}}J_1(x_* - x_1)\| + \|J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}}(x_* - x_1)\|) \\ &\le 2\mu(\|J_1^{\mathrm{T}}\| \times \|F_* - F_1 - J_1(x_* - x_1)\| + \|J_1^{\mathrm{T}}F_1a_1^{\mathrm{T}}(x_* - x_1)\|) \\ &\le 2\mu\left(\frac{\alpha\gamma}{2}\|x_1 - x_*\|^2 + \frac{\alpha\gamma M}{2m}\|x_1 - x_*\|^2\right) \\ &\le \mu\alpha\gamma\left(1 + \frac{M}{m}\right)\|x_1 - x_*\|^2.\end{aligned}$$

Together (3.5) with the above result, we can obtain

$$\begin{aligned}\|x_2 - x_*\| &\le \mu\alpha\gamma\left(1 + \frac{M}{m}\right)\|x_1 - x_*\|^2 \\ &\le \mu\alpha\gamma\left(1 + \frac{M}{m}\right)q^2\|x_0 - x_*\|^2 \\ &\le \frac{\mu\alpha\gamma(m+M)}{m}q^2\|x_0 - x_*\|\varepsilon \\ &< q^2\|x_0 - x_*\|.\end{aligned}$$

Thus, we have $x_2 \in N(x_*, r)$, and complete the proof when $k = 1$.

Now we assume that

$$\|x_{k+1} - x_*\| \le \mu\alpha\gamma\left(1 + \frac{M}{m}\right)\|x_k - x_*\|^2$$

and

$$\|x_{k+1} - x_*\| < q^{k+1}\|x_0 - x_*\|$$

are true statements. From this assumption, we want to deduce the truth of step $k + 2$.

Using the introduction hypothesis, we find that

$$\begin{aligned}
\|x_{k+2}-x_*\| &= \|x_{k+1}-x_*-(J_{k+1}^{\mathrm{T}}J_{k+1}+J_{k+1}^{\mathrm{T}}F_{k+1}a_{k+1}^{\mathrm{T}})^{-1}J_{k+1}^{\mathrm{T}}F_{k+1}\| \\
&= \|x_{k+1}-x_*-(J_{k+1}^{\mathrm{T}}J_{k+1}+J_{k+1}^{\mathrm{T}}F_{k+1}a_{k+1}^{\mathrm{T}})^{-1}(J_{k+1}^{\mathrm{T}}F_{k+1}-J_{k+1}^{\mathrm{T}}F_*)\| \\
&\le \|(J_{k+1}^{\mathrm{T}}J_{k+1}+J_{k+1}^{\mathrm{T}}F_{k+1}a_{k+1}^{\mathrm{T}})^{-1}\|\times\|J_{k+1}^{\mathrm{T}}F_* \\
&\quad -J_{k+1}^{\mathrm{T}}F_{k+1}-(J_{k+1}^{\mathrm{T}}J_{k+1}+J_{k+1}^{\mathrm{T}}F_{k+1}a_{k+1}^{\mathrm{T}})(x_*-x_{k+1})\| \\
&\le 2\mu(\|J_{k+1}^{\mathrm{T}}F_*-J_{k+1}^{\mathrm{T}}F_{k+1}-J_{k+1}^{\mathrm{T}}J_{k+1}(x_*-x_{k+1})\| \\
&\quad +\|J_{k+1}^{\mathrm{T}}F_{k+1}a_{k+1}^{\mathrm{T}}(x_*-x_{k+1})\|) \\
&\le 2\mu(\|J_{k+1}^{\mathrm{T}}\|\|F_*-F_{k+1}-J_{k+1}(x_*-x_{k+1})\| \\
&\quad +\|J_{k+1}^{\mathrm{T}}F_{k+1}a_{k+1}^{\mathrm{T}}(x_*-x_{k+1})\|) \\
&\le 2\mu\left(\frac{\alpha\gamma}{2}\|x_{k+1}-x_*\|^2+\frac{\alpha\gamma M}{2m}\right)\|x_{k+1}-x_*\|^2) \\
&\le \mu\alpha\gamma\left(1+\frac{M}{m}\right)\|x_{k+1}-x_*\|^2
\end{aligned}$$

and

$$\begin{aligned}
\|x_{k+2}-x_*\| &\le \mu\alpha\gamma\left(1+\frac{M}{m}\right)\|x_{k+1}-x_*\|^2 \\
&\le \mu\alpha\gamma\left(1+\frac{M}{m}\right)q^{2(k+1)}\|x_0-x_*\|\varepsilon \\
&< q^{2k+2}\|x_0-x_*\| \\
&< q^{k+2}\|x_0-x_*\|.
\end{aligned}$$

With mathematical induction, we obtain

$$\|x_{k+2}-x^*\| \le \mu\alpha\gamma\left(1+\frac{M}{m}\right)\|x_{k+1}-x^*\|^2, \quad k=1,2,\ldots$$

$$\|x_{k+2}-x_*\| < q^{k+2}\|x_0-x_*\|, \quad k=1,2,\ldots.$$

So the proof is completed. □

Corollary 3.5 *Let the assumptions of Theorem 3.4 be satisfied. If $F(x_*)=0$, then there exists ε such that for all $x_0\in N(x_*,\varepsilon)$, the sequence $\{x_k\}$ generated by the new Gauss–Newton-like method is well-defined and converges q-quadratically to x_*.*

Proof From (3.4), we obtain the sequence $\{x_k\}$ which converges to x_*, and from (3.3) the rate of convergence is q-quadratic. □

4. Numerical experiment

In this section, in order to verify the practical effect of the new Gauss–Newton-like method proposed in this paper, we apply Algorithm 2.1(NGNL), the classical Gauss–Newton method (CGN), and the Newton-like method (INM)([13]) into a series of numerical examples and compare the performance of these methods. All numerical examples are calculated on the on a personal computer (Intel i3, 2.1GHz/2GB) with MATLAB R2014a. In the following tables,

Po: denotes the initial point;

In: denotes the number of iterations;

Ti: denotes the CPU running time (seconds);

Fn: denotes the current function value.

Example 4.1 In this example, we examine the effectiveness of the new Gauss–Newton-like method for solving nonlinear equations with single root. Numerical examples are chosen from [15] and shown in Table 1.

Table 1. Test equations of example 4.1.

Test equation	Initial point range
$f_1(x) = exp(x)sin(x) + ln(1+x^2) = 0$	$x_0 \in [-0.1, 1]$
$f_2(x) = exp(x)sin(x) + cos(x)ln(1+x) = 0$	$x_0 \in [-1, 1]$
$f_3(x) = exp(sin(x)) - x/5 - 1 = 0$	$x_0 \in [-0.5, 1]$
$f_4(x) = (x+1)exp(sin(x)) - x^2exp(cos(x)) - 1 = 0$	$x_0 \in [-1.5, 1]$
$f_5(x) = sin(x) + cos(x) + tan(x) - 1 = 0$	$x_0 \in [-1, 0.5]$
$f_6(x) = exp(-x) - cos(x) = 0$	$x_0 \in [-1, 0.5]$
$f_7(x) = ln(1+x^2) + exp(x^2 - 3x)sin(x) = 0$	$x_0 \in [-0.2, 1]$
$f_8(x) = x^3 + ln(1+x) = 0$	$x_0 \in [-0.5, 1]$
$f_9(x) = sin(x) - x/3 = 0$	$x_0 \in [-0.5, 1]$
$f_{10}(x) = (x-10)^6 - 10^6 = 0$	$x_0 \in [-1, 1]$

We apply Algorithm 2.1 NGNL, INM, and CGN into above examples and choose three different initial points which are randomly generated in the range of initial point. Calculation will be finished when $\|F(x_k)\| \leq 10^{-6}$ or the number of iterations exceeds 100. The numerical results are shown in Table 2.

Example 4.2 To show that Algorithm 2.1 is an effective algorithm for solving a system of nonlinear equations, we select 17 test functions from the test problems set given by Moré et al. in [11], which are shown in Table 3. (Note: In test functions F_1-F_{11}, $m = n$ and in test functions F_{12}-F_{17}, $m > n$, where n represents the number of variables and m represents the number of functions $F_i(x), i = 1, 2, ..., m$.)

In this example, the standard initial point given in [11] is chosen as the initial point for each question. The algorithm is terminated when $\|F(x_k)\| < 10^{-6}$ or the number of iterations exceeds $100(n+1)$. The numerical results are shown in Table 4, where Dim represents the initial dimension and \ indicates that the number of iterations exceeds $100(n+1)$.

To compare the performances of the new Gauss–Newton-like method (NGNL), the classical Gauss–Newton method (CGN), the and Newton-like method (INM), we choose the performance profile proposed by Dolan and Moré in [5] as a means (see Figures 1 and 2).

Table 2. Numerical experiment results of INM, NGNL, and CGN.

Fuction		INM			NGNL			CGN		
	Po	In	Ti	Fn	In	Ti	Fn	In	Ti	Fn
$f_1(x)$	0.7962	4	0.1848	3.35E-07	4	0.0007	3.35E-07	5	0.0017	6.97E-08
	0.2063	3	0.0015	1.94E-07	3	0.0030	1.94E-07	4	0.0016	1.75E-09
	0.5016	4	0.0007	1.02E-09	4	0.0007	1.02E-09	5	0.0008	1.47E-10
$f_2(x)$	0.8119	3	0.0018	2.45E-11	3	0.0015	2.45E-11	4	0.0017	5.54E-11
	0.9150	3	0.0017	4.73E-10	3	0.0017	4.73E-10	4	0.0019	2.95E-10
	0.9298	3	0.0007	6.53E-10	3	0.0006	6.53E-10	4	0.0008	3.66E-10
$f_3(x)$	-0.3095	3	0.0009	2.83E-08	3	0.0012	2.83E-08	4	0.0011	8.04E-11
	-0.2636	3	0.0013	3.00E-09	3	0.0006	3.00E-09	4	0.0013	4.26E-12
	0.9559	3	0.0006	1.66E-11	3	0.0015	1.66E-11	3	0.0006	1.66E-11
$f_4(x)$	0.7834	3	0.0019	5.60E-08	3	0.0015	5.60E-08	4	0.0021	1.80E-10
	0.0809	4	0.0008	2.07E-14	4	0.0007	2.07E-14	4	0.0008	1.20E-12
	-1.2561	4	0.0007	7.02E-08	4	0.0017	7.02E-08	5	0.0008	5.65E-13
$f_5(x)$	0.9143	4	0.0304	1.78E-15	4	0.0012	1.78E-15	4	0.0014	7.68E-10
	-0.0292	2	0.0006	5.27E-11	2	0.0013	5.27E-11	2	0.0007	2.31E-08
	0.6006	3	0.0007	4.59E-13	3	0.0006	4.59E-13	3	0.0006	6.36E-09
$f_6(x)$	-0.7872	4	0.0012	9.34E-11	4	0.0025	9.34E-11	5	0.0012	3.62E-11
	-0.3674	3	0.0005	1.41E-07	3	0.0015	1.41E-07	4	0.0007	1.46E-09
	0.3736	4	0.0007	3.44E-09	4	0.0015	3.44E-09	5	0.0008	2.02E-09
$f_7(x)$	0.7506	4	0.0022	2.48E-10	4	0.0014	2.48E-10	5	0.0019	3.63E-12
	0.9514	4	0.0008	4.35E-09	4	0.0018	4.35E-09	5	0.0009	5.82E-11
	0.5869	4	0.0006	2.35E-13	4	0.0016	2.35E-13	4	0.0008	3.40E-08
$f_8(x)$	-0.4464	4	0.0015	6.22E-15	4	0.0013	6.33E-15	4	0.0013	2.41E-08
	0.7737	4	0.0008	7.01E-08	4	0.0007	7.01E-08	4	0.0007	1.21E-07
	0.9010	4	0.0008	8.16E-08	4	0.0017	8.16E-08	5	0.0009	2.71E-12
$f_9(x)$	0.5181	3	0.0011	1.29E-08	3	0.0011	1.29E-08	3	0.0012	9.06E-12
	0.6366	3	0.0006	7.90E-07	3	0.0010	7.90E-07	3	0.0007	6.95E-09
	0.6146	3	0.0007	3.83E-07	3	0.0006	3.83E-07	3	0.0005	2.14E-09
$f_{10}(x)$	-0.2155	3	0.0011	0	3	0.0010	0	4	0.0012	0
	0.3110	3	0.0006	1.82E-08	3	0.0006	1.82E-08	4	0.0007	1.05E-09
	-0.6576	4	0.0008	0	4	0.0017	0	4	0.0007	1.95E-07

We suppose that n_s represents the number of methods and n_p is the number of test problems in test set p. For each problem p and method s, $f_{p,s}$ is the iterations when the method s is used to solve the problem p. Let the performance ratio

$$r_{p,s} := \frac{f_{p,s}}{\min\{f_{p,s} : s \in S\}},$$

where S is the methods set which includes NGNL, CGS, and INM. We assume that a parameter $r_M \geq r_{p,s}$ is chosen for all p, s, and $r_{p,s} = r_M$ if and only if solver s cannot solve problem p. Aiming to get the composite

Table 3. Test problems of Example 4.2.

Function	Name	Dim
F_1	*Rosenbrock function*	$n = 2,\ m = n$
F_2	*Powell badly scaled function*	$n = 2,\ m = n$
F_3	*Freudenstein and Roth function*	$n = 2,\ m = n$
F_4	*Powell singular function*	$n = 4,\ m = n$
F_5	*Trigonometric function*	n *variable*, $m = n$
F_6	*Trigonometric Exponential function*	n *variable*, $m = n$
F_7	*Broyden tridiagonal function*	n *variable*, $m = n$
F_8	*Extended Powell singular function*	n *variable*, $m = n$
F_9	*Discrete boundary value function*	n *variable*, $m = n$
F_{10}	*Discrete integral equation function*	n *variable*, $m = n$
F_{11}	*Broydenbanded function*	n *variable*, $m = n$
F_{12}	*Brown badly scaled function*	$n = 2,\ m = 3$
F_{13}	*Beale function*	$n = 2,\ m = 3$
F_{14}	*Box three − dimensional function*	$n = 3,\ m \geq n$
F_{15}	*Wood function*	$n = 4,\ m = 6$
F_{16}	*Biggs EXP6 function*	$n = 6,\ m \geq n$
F_{17}	*Variably dimensioned function*	n *variable*, $m = n + 2$

assessment for each solver, we define the performance profile of the number of iteration for solver s as follows:

$$\rho_s(\gamma) := \frac{1}{n_p} size\{p \in P : r_{p,s} \leq \gamma\},$$

where $\gamma \geq r_M$.

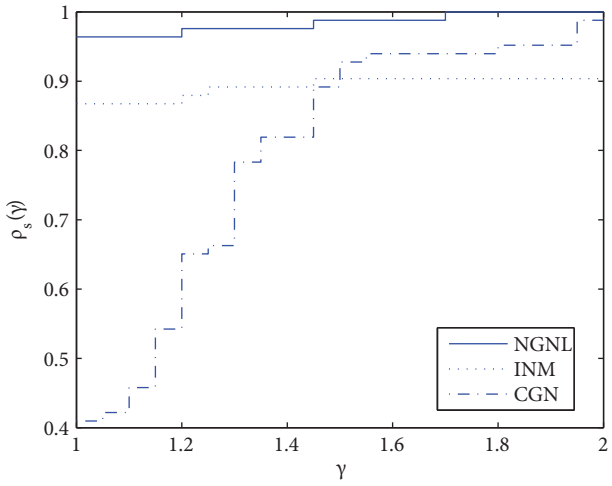

Figure 1. Performance profiles of INM, NGNL, and CGN for iteration numbers.

With the same idea, we let $t_{p,s}$ be the CPU time required to solve problem p by solver s and assume

Table 4. Numerical experiment results of INM, NGNL, and CGN.

Pr		INM			NGNL			CGN		
	Dim	In	Ti	Fn	In	Ti	Fn	In	Ti	Fn
F_1	2	3	0.5065	0.00E+00	3	0.8615	7.38E-13	2	0.3909	1.08E-12
F_2	2	6	1.4631	8.59E-07	6	2.0759	8.59E-07	11	2.7622	7.84E-07
F_3	2	21	6.6486	8.86E-09	21	8.9845	8.86E-09	42	12.8778	1.74E-11
F_4	4	10	0.0065	5.10E-07	10	0.0517	5.05E-07	12	0.0057	7.56E-07
F_5	5	5	0.0037	1.31E-07	5	0.0185	1.87E-07	6	0.0161	6.60E-11
F_6	5	17	0.0148	3.68E-07	12	0.0102	6.41E-07	12	0.0207	9.19E-07
	50	14	0.0458	5.37E-07	13	0.0954	3.04E-07	14	0.0705	2.80E-07
	500	12	3.0313	6.58E-07	12	8.6255	7.27E-07	14	4.7145	2.84E-07
	1000	12	25.7258	8.07E-07	12	81.5752	8.73E-07	14	43.1951	2.89E-07
F_7	5	4	0.0159	2.21E-13	4	0.0075	2.18E-13	4	0.0071	1.16E-09
	50	4	0.0490	1.06E-11	4	0.0632	1.05E-11	4	0.0393	1.06E-09
	200	4	0.9022	6.87E-11	4	2.8000	6.86E-11	4	1.4341	1.06E-09
	1000	4	7.9124	8.87E-11	4	24.8408	8.86E-11	4	12.8777	1.06E-09
F_8	4	10	0.0124	5.10E-07	10	0.0072	5.05E-07	12	0.0114	7.56E-07
	40	11	0.1260	2.77E-07	11	0.0187	2.74E-07	13	0.0706	5.98E-07
	400	11	4.2215	8.76E-07	11	3.4422	8.66E-07	14	8.4967	4.73E-07
	1200	12	184.2385	2.60E-07	12	111.5607	2.57E-07	14	346.1998	8.19E-07
F_9	5	2	3.2737	7.73E-10	2	10.5538	7.73E-10	2	6.6547	7.73E-10
	50	2	3.4177	7.79E-10	2	10.8294	7.79E-10	2	6.73912	7.78E-10
	500	2	3.3989	5.71E-10	2	10.7255	5.71E-10	2	6.9348	5.71E-10
	1000	1	1.0430	2.78E-07	1	3.1195	2.78E-07	1	3.1146	2.78E-07
F_{10}	5	3	0.0191	5.29E-08	3	0.0113	5.26E-08	3	0.0103	5.32E-08
	50	3	0.0384	8.17E-11	3	0.0481	8.18E-11	3	0.0659	9.70E-10
	500	3	0.9051	8.19E-07	2	1.6557	8.19E-07	3	1.7809	1.00E-10
	1000	2	5.5000	4.87E-07	2	12.9404	4.88E-07	3	12.8934	5.03E-11
F_{11}	10	5	0.0109	2.42E-13	5	0.0113	2.38E-13	5	0.0096	1.55E-08
	50	5	0.0304	7.37E-12	5	0.0625	7.29E-12	5	0.0501	1.55E-08
	500	5	0.2895	2.16E-11	5	0.5162	2.16E-11	5	0.3594	1.55E-08
	1000	5	1.5605	2.58E-11	5	2.7303	2.58E-11	5	2.0991	1.55E-08
F_{12}	2,3	\	\	\	6	1.9253	2.28E-07	6	1.2149	3.33E-16
F_{13}	2,3	\	\	\	5	1.8958	7.44E-07	5	1.1648	3.05E-07
F_{14}	3,10	\	\	\	5	6.1755	2.59E-12	5	3.8780	3.37E-10
F_{15}	4,6	\	\	\	63	38.9144	1.08E-07	70	24.7892	2.99E-09
F_{16}	6,10	\	\	\	6	0.0084	1.31E-07	6	0.0099	6.69E-13
F_{17}	10	\	\	\	8	0.0514	3.47E-09	9	0.0075	3.99E-08
	50	\	\	\	12	0.0367	2.23E-11	\	\	\
	500	\	\	\	20	1.3503	0.00E+00	\	\	\

that a parameter $\tau_M \geq r_{p,s}$ is chosen for all p, s, and $r_{p,s} = \tau_M$ if and only if solver s cannot solve problem p. The performance ratio

$$r_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}$$

and the performance profile of CPU time

$$\rho_s(\tau) := \frac{1}{n_p} size\{p \in P : r_{p,s} \leq \tau\},$$

where $\tau \geq \tau_M$.

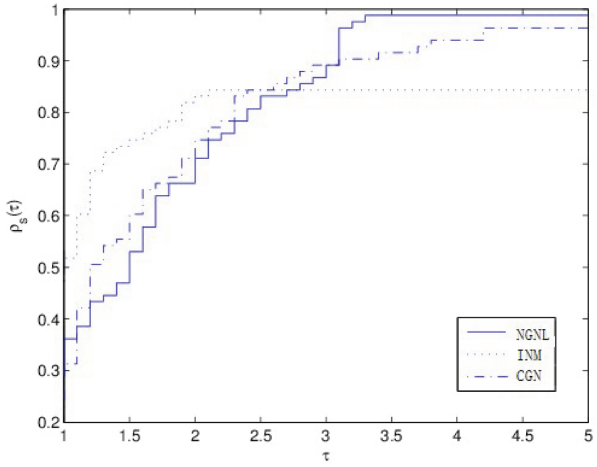

Figure 2. Performance profiles of INM, NGNL, and CGN for CPU time.

5. Results and discussion

Based on the above numerical results in Tables 1–4 and Figures 1 and 2, in this section, we give some conclusions about the performance of the NGNL, CGN, and INM.

1. In terms of the numerical results in Tables 1–4 and Figures 1 and 2, it is obvious that the new proposed method can be used as an alternative to existing methods or in some cases where existing methods are not successful.

2. It can be seen from Table 2 that the iteration number of NGNL is lower than yhose of CGN and INM. The CPU time is nearly the same for NGNL, CGN, and INM. The numerical results indicate that NGNL has nearly the same efficiency compared with CGN and INM for one-dimensional nonlinear equation.

3. In terms of the numerical results in Table 4, the iterative method NGNL needs more CPU time for some test problems. However, we can see that the iterative method INM fails to obtain real solutions for test problems (F12, $\cdots$, F17), the iterative method CGN fails to obtain real solutions for test problem (F17), while our proposed new Gauss–Newton-like method can solve all test problems successfully.

4. The results in Figures 1 and 2 indicate that the numerical performance of the iterative method NGNL is superior to those of the iterative method CGN and iterative method INM for nonlinear equations.

From the above comparison, we see that the new proposed method (NGNL) is effective and the numerical results also lead us to believe that the new method has definite practical utility.

6. Conclusion

In this paper, we apply the rational approximation model to solve the least squares problem, and obtain a new Gauss–Newton-like method for solving nonlinear equations, where the new method revises the $J_k^{\mathrm{T}} J_k$ matrix by a rank-one matrix at each iteration. Besides, we proposed an algorithm for new Gauss–Newton-like method and analyzed its convergence. The numerical results show that the new method is more effective than some existing methods for solving nonlinear equations. Hence, the new proposed method provides a new choice for science and engineering practice.

There are some questions related to the study. For example, how to find the best a_k in the rational approximation model and how to extend the new proposed method for nonsmooth problems are not be well solved in this paper and will be topics for our future work.

Acknowledgments

The authors would like to express their sincere appreciation to the referees for his/her valuable suggestions and comments which improved the original draft of the paper. The authors are supported by the Fundamental Research Funds for the Central Universities (NO.2017XKQY088)

References

[1] Argyros IK, Magreñán AA. Inexact Gauss–Newton-like method for least square problems. In: Iterative Methods and their Dynamics with Applications: A Contemporary Study. Taylor and Francis Group, CRC Press, 2017, pp. 144-160.

[2] Amini K, Rostami F. A modified two steps Levenberg-Marquardt method for nonlinear equations. Journal of Computational and Applied Mathematics 2015; 288: 341-350. doi: 10.1016/j.cam.2015.04.040

[3] Cartis C, Roberts L. A derivative-free Gauss–Newton method. Mathematical Programming Computation 2019; 11(4): 631-674. doi: 10.1007/s12532-019-00161-7

[4] Dennis Jr JE, Schnabel RB. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. NJ, USA: Prentice-Hall, Englewood Cliffs, 1983.

[5] Dolan ED, Moré JJ. Benchmarking optimization software with performance profiles. Mathematical Programming 2002; 91: 201-213. doi: 10.1007/s101070100263

[6] Fan JY, Yuan YX. On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption. Computing 2005; 74: 23-39.

[7] Fan JY. Accelerating the modified Levenberg-Marquardt method for nonlinear equations. Mathematics of Computation 2014; 83: 1173-1187. doi: 10.1090/S0025-5718-2013-02752-4

[8] Fang XW, Ni Q, Zeng ML. A modified quasi-Newton method for nonlinear equations. Journal of Computational and Applied Mathematics 2018; 328: 44-58. doi: 10.1016/j.cam.2017.06.024

[9] Levenberg K. A method for the solution of certain nonlinear problems in least squares. The Quarterly of Applied Mathematics 1944; 2: 164-166. doi: 10.1090/qam/10666

[10] Marquardt DW. An algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics 1963; 11: 431-441. doi: 10.1137/0111030

[11] Moré JJ, Garbow BS, Hillstrom KE. Testing unconstrained optimization software. ACM Transactions on Mathematical Software 1981; 7(1): 17-41. doi: 10.1145/355934.355936

[12] Sui YK, Saheya B, Chen GQ. An improvement for the rational approximation RALND at accumulated two point information. Mathematica Numerica Sinica 2014; 36: 51-64. doi: 10.1002/joc.1509

[13] Saheya B, Chen GQ, Sui YK, Wu CY. A new Newton-like method for solving nonlinear equations. SpringerPlus 2016; 5: 1269. doi: 10.1186/s40064-016-2909-7

[14] Schwetlick H, Tilleb Y, Schellong W. Gauss–Newton-like methods for nonlinear least squares with equality constraints-local convergence and applications. Statistics 1985; 16(2): 167-178. doi: 10.1080/02331888508801839

[15] Thukal R. New modification of Newton method with third-order convergence for solving nonlinear equations of type $f(0) = 0$. American Journal of Computational and Applied Mathematics 2016; 6: 14-18. doi: 10.5923/j.ajcam.20160601.03

[16] Wang HJ. New third-order method for solving systems of nonlinear equations. Numerical Algorithms 2009; 50: 271-282. doi: 10.1007/s11075-008-9227-2

[17] Yamashita N, Fukushima M. On the rate of convergence of the Levenberg-Marquardt method. Computing 2001; 15: 239-249.