Saheya *et al. SpringerPlus (2016) 5:1269*
DOI 10.1186/s40064-016-2909-7

SpringerPlus

RESEARCH Open Access

A new Newton-like method for solving nonlinear equations

B. Saheya[1,2], Guo-qing Chen[1], Yun-kang Sui[3*] and Cai-ying Wu[1]

*Correspondence: yksui@sina.com
[3] College of Mechanical Engineering and Applied Electronics Technology, Beijing University of Technology, Beijing 100124, China
Full list of author information is available at the end of the article

Abstract

This paper presents an iterative scheme for solving nonline ar equations. We establish a new rational approximation model with linear numerator and denominator which has generalizes the local linear model. We then employ the new approximation for nonlinear equations and propose an improved Newton's method to solve it. The new method revises the Jacobian matrix by a rank one matrix each iteration and obtains the quadratic convergence property. The numerical performance and comparison show that the proposed method is efficient.

Keywords: Rational approximate function, Improved Newton's method, Local convergence

Mathematics Subject Classification: 90C25, 90C30

Background

We consider the system of nonlinear equations

$$F(x) = 0, \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable function. All practical algorithms for solving (1) are iterative. Newton's method is the most widely used method in applications (see Traub 1964; Ortega and Rheinboldt 1970; Dennis and Schnabel 1993; Kelley 2003; Petković et al. 2013a).

The linearization of Eq. (1) at an iteration point x_k is

$$F(x_k) + J(x_k)s = 0, \tag{2}$$

where $s = x - x_k$ and $J(x_k)$ is the Jacobian matrix of $F(x)$ at x_k. For notation purposes, let $F_k = F(x_k)$ and $J_k = J(x_k)$. If $m = n$ and $J(x_k)$ is nonsingular, then the linear approximation (2) gives the Newton–Raphson iteration

$$x_{k+1} = x_k - J_k^{-1}F_k. \tag{3}$$

In 1669, Newton first used the Newton iteration (2) to solve a cubic equation. In 1690 Raphson first employed the formula (3) to solve a general cubic equations. Then Fourier (1890), Cauchy (1829), and Fine (1916) established the convergence theorem of Newton's method for different cases. In 1948, Kantorovich (1948) established the convergence

Springer Open

theorem referred to the Newton–Kantorovich theorem. This theorem is the main tool for proving the convergence of various Newton-type methods.

There are various Newton-Type methods for solving nonlinear equations. Dembo et al. (1982) proposed an inexact Newton method. This method approximately solves the linear equation (2). Another most efficient approach is approximating the Jacobian or inverse of the Jacobian in some way. In this way, the approximation of the Jacobian satisfies the secant equation

$$B_k s_{k-1} = F(x_k) - F(x_{k-1}), \tag{4}$$

where B_k is an approximation for the Jacobian and $s_{k-1} = x_k - x_{k-1}$. For this kind of method, the secant equation (4) plays a vital role; therefore a wide variety of methods that satisfy the secant equation have been designed (Dennis and Schnabel 1993; Kelley 2003). Qi and Sun (1993) extended Newton's method for solving a nonlinear equation of several variables to a nonsmooth case by using the generalized Jacobian instead of the derivative. This extension includes the B-derivative version of Newton's method as a special case. In order to improve the convergence order of Newton-type methods, many higher order approaches have been proposed in past years. In particular, there is much literature focused on the nonlinear scalar function. Petković et al. (2013b) provide a survey, many of which are presented in the book (Petković et al. 2013a). For the nonlinear vector function $F(x)$ in (1), there are still a lot of higher order methods. For instance, Grau-Sánchez et al. (2011), Noor and Waseem (2009), Homeier (2004), and Frontini and Sormani (2004) have proposed a third order method using one function value, two Jacobian matrices and two matrix inversions per iteration. In Darvishi and Barati (2007a), a third order method has been proposed with two function values, one Jacobian and one matrix inversion per iteration. Darvishi and Barati (2007b), and Sharma et al. (2013) developed a fourth order method. In pursuit of a higher order algorithm, researchers have also proposed fifth and sixth order methods in Grau-Sánchez et al. (2011). In summary, these higher order methods need more function values, Jacobians or matrix inversions per iteration.

In this paper, we are interested in a Newton-type method with high computational efficiency for solving the system of nonlinear equations (1). Motivated by the approach in Sui et al. (2014), we provide a new rational model $R : \mathbb{R}^n \to \mathbb{R}^m$. Although our approximation function is similar to the real valued function RALND studied in Sui et al. (2014), the proposed function is different from the RALND function. Based on this model, we linearize the nonlinear function $F(x)$ and obtain a linear equation that is different from the first order Taylor polynomial. We then propose an improved Newton's algorithm to solve nonlinear equations (1). In the new algorithm, in order to reflect more curvature information of nonlinear functions, the Jacobian matrix is updated by rank one matrix in each iteration. This method possesses high computational efficiency , and therefore does not increase calculation of function value, Jacobian or inverse Jacobian. Applying Newton's method's validation criteria, we prove that the algorithm is well-defined and the convergence rate is quadratic under some suitable conditions. The preliminary numerical experiment results and comparison are reported, showing the effectiveness of the algorithm.

This paper is organized as follows. We give a new rational approximation and improved Newton's method in the next section. In section "Convergence analysis", converge analysis is discussed and some numerical experiment results are reported in section "Numerical experiments". The last section is a brief conclusion.

Rational approximation and improved Newton's method

Based on the information of the last two points, Sui proposed a RALND function (Sui et al. 2014) $r : \mathbb{R}^n \to \mathbb{R}$ with linear numerator and denominator that is defined by

$$r(x) = a_0 + \frac{a_k^{\mathrm{T}}(x - x_k)}{1 + b_k^{\mathrm{T}}(x - x_k)}, \tag{5}$$

where $a_k, b_k \in \mathbb{R}^n$ are the undetermined coefficient vectors and $x_k \in \mathbb{R}^n$ is the current point. Let

$$c_0 = \nabla^{\mathrm{T}} f(x_{k-1})(x_{k-1} - x_k), \quad c_1 = \nabla^{\mathrm{T}} f(x_k)(x_{k-1} - x_k).$$

Under the following interpolation conditions

$$r(x_k) = f(x_k), \nabla r(x_k) = \nabla f(x_k), \quad \nabla r(x_{k-1}) = \nabla f(x_{k-1}),$$

we obtain the RALND function

$$r(x) = f(x_k) + \frac{\nabla^{\mathrm{T}} f(x_k)(x - x_k)}{1 + \frac{1}{c_0}\left(\sqrt{\frac{c_0}{c_1}}\nabla^{\mathrm{T}} f(x_k) - \nabla^{\mathrm{T}} f(x_{k-1})\right)(x - x_k)}, \tag{6}$$

where $x_k \in \mathbb{R}^n$, $x_{k-1} \in \mathbb{R}^n$ are the current point and the preceding point. The RALND function has many good properties (Sui et al. 2014). For example, it is monotone with any direction and has more curvature information of the nonlinear function $F(x)$ than the linear approximation model. These properties may be able to reduce the number of iterations when using an iteration method that was constructed by RALND to solve (1). Although the RALND function possesses some nice properties, the function $r : \mathbb{R}^n \to \mathbb{R}$ defined by (6) is a real valued function with each function having a different vector b_k. This make it more complex for nonlinear equations.

Next, we employ the RALND function with the same horizon vector b_k for all nonlinear functions $F_i(x), i = 1, \ldots, n$ at x_k, and approximate the nonlinear equations (1) by

$$F(x_k + s) \approx R(x_k + s) = F_k + \frac{J_k s}{1 + b_k^{\mathrm{T}} s} = 0. \tag{7}$$

When $b_k = 0$, the rational function (7) reduces to the linear expansion (2). There is a well-known analogy between the rational function (7) and RALND (6), but the function (7) is different from (6). For the RALND function (6), each function $F_i(x), i = 1, \ldots, m$ has a different vector $b_i^{(k)}, i = 1, \ldots, m$ at current iteration point x_k, but the new approximation function (7) has the same vector b_k for all functions $F_i(x), i = 1, \ldots, m$ at the same iteration point x_k. This is the main difference between the two functions (7) and (6). Because of this difference, the function (7) is more suitable for nonlinear equations.

Similar to the linearization approach in (2), from approximate equations (7) we can obtain a new iterative formula

$$(J_k + F_k b_k^{\mathrm{T}})s_k = -F_k. \tag{8}$$

If the matrix $J_k + F_k b_k^T$ is invertible, it follows that

$$x_{k+1} = x_k - (J_k + F_k b_k^T)^{-1} F_k. \tag{9}$$

when $b_k = 0$, the iterative scheme (8) and (9) reduce to the linear equations (2) and Newton–Raphson iteration (3), respectively.

Moreover, Davidon proposed the conic model (Davidon 1980; Sorensen 1980) and many researchers have studied the conic model and collinear scaling algorithms (Ariyawansa 1990; Ariyawansa and Lau 1992; Deng and Li 1995; Gourgeon and Nocedal 1985). Near the current iteration point x_k, the conic function $c(x)$ is defined by

$$f(x) \approx c_k(x) = f(x_k) + \frac{\nabla f(x_k)^{\mathrm{T}} s}{1 + b_k^{\mathrm{T}} s} + \frac{s^{\mathrm{T}} B_k s}{2\left(1 + b_k^{\mathrm{T}} s\right)^2}. \tag{10}$$

In the conic model (10), the horizon vector b_k is a parameter. This parameter gives the conic model more freedom. Many researchers have given more attention to b_k. As a result, some methods of choosing the horizon vector have been developed (Davidon 1980; Deng and Li 1995; Sheng 1995). Interestingly, the function (7) is the first two terms of conic model (10). In what follows we use these methods to determine the vector b_k in (7).

After a step from x_{k-1} to x_k, we update b_{k-1} to b_k by requiring the following extra interpolation condition

$$R(x_{k-1}) = F(x_{k-1}). \tag{11}$$

This causes the search direction in (9) to depend on the Jacobian of the current point and the function values of the preceding point as well as the current point. In Newton's method the search direction is determined by the Jacobian and function value of the current point. Compared with Newton's method, more flexibility and more accurate approximation of the nonlinear function may be expected for the rational model (7).

From (11) we have

$$F_{k-1} = F_k - \frac{J_k s_{k-1}}{1 - b_k^{\mathrm{T}} s_{k-1}}, \tag{12}$$

where $s_{k-1} = x_k - x_{k-1}$. Let

$$\beta_k = 1 - b_k^{\mathrm{T}} s_{k-1}, \tag{13}$$

$$y_{k-1} = F_k - F_{k-1}. \tag{14}$$

Considering (12), we get

$$\beta_k y_{k-1} = J_k s_{k-1},$$

thus

$$\beta_k = \frac{y_{k-1}^{\mathrm{T}} J_k s_{k-1}}{y_{k-1}^{\mathrm{T}} y_{k-1}}. \tag{15}$$

Note that

$$b_k = \frac{(1-\beta_k)a_k}{a_k^{\mathrm{T}} s_{k-1}} \tag{16}$$

for any $a_k \in \mathbb{R}^n$ with $a_k^{\mathrm{T}} s_{k-1} \neq 0$, will satisfy (13). Considering the special choice $a_k = s_{k-1}$, we have

$$b_k = \frac{(1-\beta_k)s_{k-1}}{s_{k-1}^{\mathrm{T}} s_{k-1}} = \frac{y_{k-1}^{\mathrm{T}}(y_{k-1} - J_k s_{k-1}) s_{k-1}^{\mathrm{T}}}{\left(y_{k-1}^{\mathrm{T}} y_{k-1}\right)\left(s_{k-1}^{\mathrm{T}} s_{k-1}\right)}. \tag{17}$$

Analogously, we can consider another method (Sheng 1995) for constructing horizon vectors. Using (17) and (15), we see that

$$F_k b_k^{\mathrm{T}} = \frac{y_{k-1}^{\mathrm{T}}(y_{k-1} - J_k s_{k-1})}{y_{k-1}^{\mathrm{T}} y_{k-1}} \frac{F_k s_{k-1}^{\mathrm{T}}}{s_{k-1}^{\mathrm{T}} s_{k-1}}. \tag{18}$$

Next, we give the improved Newton's method for system of nonlinear equations.

Algorithm 1. (Improved Newton's Method (INM))

Let x_0 be a given an initial approximation to the solution of (1) such that J_0 is nonsingular and choose ε_0. The step for obtaining x_{k+1}, $k = 1, 2, \cdots$ are:

Step 1. *Let $b_0 = 0$, $\varepsilon = \varepsilon_0$, and set $k = 0$.*

Step 2. *If $\|F_k\| \le \varepsilon$, stop.*

Step 3. *Compute $F_{k+1}b_{k+1}^T$ and x_{k+1} by*

$$\begin{aligned} F_k b_k^{\mathrm{T}} &= \frac{y_{k-1}^{\mathrm{T}}(y_{k-1} - J_k s_{k-1})}{y_{k-1}^{\mathrm{T}} y_{k-1}} \frac{F_k s_{k-1}^{\mathrm{T}}}{s_{k-1}^{\mathrm{T}} s_{k-1}}, \\ x_{k+1} &= x_k - (J_k + F_k b_k^{\mathrm{T}})^{-1} F_k. \end{aligned}$$

Step 4. *Set $k = k + 1$. Go to* Step 2.

There are two differences between Algorithm 1 and Newton's method. First, INM uses the rank one technique to revise the Jacobian in every iteration. Second, INM utilises the function values of the previous iteration point.

For the one dimensional nonlinear equation $f(x) = 0$, where $f : \mathbb{R} \to \mathbb{R}$ is continuously differentiable on $D \subset \mathbb{R}$, the nonlinear function of $f(x)$ is approximated by

$$f(x) \approx f(x_k) + \frac{f'(x_k)(x - x_k)}{1 + b_k(x - x_k)}.$$

Then, we have

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k) + f(x_k)b_k}. \tag{19}$$

We also use the interpolation method to determined the parameter b_k by

$$b_k = \frac{f(x_k) - f(x_{k-1}) - f'(x_k)(x_k - x_{k-1})}{(f(x_k) - f(x_{k-1}))(x_k - x_{k-1})}. \tag{20}$$

Then (20) together with (19) gives the following iteration scheme

$$x_{k+1} = x_k - \frac{f(x_k)}{\frac{f'(x_k)f(x_{k-1})}{f(x_{k-1})-f(x_k)} + \frac{f(x_k)}{x_k - x_{k-1}}}. \tag{21}$$

This is a new modified Newton formula.

Convergence analysis

In this section, we prove the local quadratic convergence of Algorithm 1 for system of nonlinear equations. The techniques of the proof are similar to Newton's method for nonlinear equations. In the rest of this paper, we make the following assumptions:

Assumption 1 (i) $J(x^*)$ is nonsingular and there exist a constant $\mu > 0$, such that $\|J(x^*)\| \le \mu$.

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

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

For proving the convergence theorem we need the following Lemmas.

Lemma 1 *Let $F : \mathbb{R}^n \to \mathbb{R}^m$ satisfy the* (ii) *of Assumption 1. Then for any $x + s \in D$,*

$$\|F(x+s) - F(x) - J(x)s\| \le \frac{\gamma}{2}\|s\|^2. \tag{22}$$

Proof Please refer to Lemma 4.1.12 in Dennis and Schnabel (1993). □

Lemma 2 *Let F, J satisfy the conditions of Lemma 1, and assume that $J(x^*)$ exists. Then there exist $\varepsilon > 0$ and $0 < m < M$, such that*

$$m\|v - u\| \le \|F(v) - F(u)\| \le M\|v - u\|, \tag{23}$$

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

Proof Please refer to Lemma 4.1.16 in Dennis and Schnabel (1993). □

With the help of the preceding two lemmas we can prove the following Theorem of convergence. We denote the epsilon neighborhood of x_* by $N(x_*, \varepsilon)$, i.e.,

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

Theorem 1 *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ satisfy Assumption 1 and suppose that there exist $x_* \in \mathbb{R}^n$, $m > 0$ and $r > 0$, such that $N(x_*, r) \subset D$, $F(x_*) = 0$. Then there exist $\varepsilon > 0$ such that for all $x_0 \in N(x_*, \varepsilon)$ the sequence $\{x_2, x_3, \cdots\}$ generated by Algorithm 1 is well defined, converges to x_*, and obeys*

$$\|x_{k+1} - x_*\| \le \frac{\mu\gamma(m+\gamma)}{m}\|x_k - x_*\|^2, \quad k = 1, 2, \ldots \tag{24}$$

Proof Since $b_0 = 0$, we obtain the following inequality from the proof of Newton's method (Dennis and Schnabel 1993),

$$\|x_1 - x_*\| \le \frac{1}{2}\|x_0 - x_*\|. \tag{25}$$

Let

$$\varepsilon = \min\left\{r, \frac{2m}{\mu\gamma(2m+\gamma)}\right\}.$$

By a routine computation,

$$\begin{aligned}\left\|J(x_*)^{-1}\left[\left(J_1 + F_1 b_1^T\right) - J(x_*)\right]\right\| &\le \|J(x_*)^{-1}\|\left(\|J_1 - J(x_*)\| + \left\|F_1 b_1^T\right\|\right)\\ &\le \mu\left(\|J_1 - J(x_*)\| + \left\|F_1 b_1^T\right\|\right)\\ &\le \mu\left(\gamma\|x_1 - x_*\| + \left\|F_1 b_1^T\right\|\right).\end{aligned}$$

Considering the second term of the above expression, it follows from (22) and (23) that

$$\begin{aligned}\left\|F_1 b_1^T\right\| = \|F_1\|\|b_1\| \le \|F_1\|&\left\|\frac{y_0^{\mathrm{T}}(F_1 - F_0 - J_1 s_0)}{y_0^{\mathrm{T}} y_0}\right\|\frac{1}{\|s_0\|}\\ &\le \|F_1\|\frac{\|(F_1 - F_0 - J_1 s_0)\|}{\|y_0\|\|s_0\|} \le \|F_1\|\frac{\gamma\|s_0\|}{2\|y_0\|}\\ &\le \|F_1\|\frac{\gamma}{2m} = \frac{\gamma}{2m}\|F_1 - F_*\|\\ &\le \frac{\gamma^2}{2m}\|x_1 - x_*\|.\end{aligned}$$

Then,

$$\begin{aligned}\left\|J(x_*)^{-1}\left[\left(J_1 + F_1 b_1^T\right) - J(x_*)\right]\right\| &\le \mu\gamma\left(1 + \frac{\gamma}{2m}\right)\|x_1 - x_*\|\\ &\le \frac{\mu\gamma}{2}\left(1 + \frac{\gamma}{2m}\right)\|x_0 - x_*\|\\ &\le \frac{\mu\gamma}{2}\left(1 + \frac{\gamma}{2m}\right)\varepsilon \le \frac{1}{2}.\end{aligned} \tag{26}$$

Therefore, by the perturbation theorem, $J_1 + F_1 b_1^T$ is nonsingular and

$$\begin{aligned}\left\|\left(J_1 + F_1 b_1^T\right)^{-1}\right\| &= \frac{\left\|J(x_*)^{-1}\right\|}{1 - \left\|J(x_*)^{-1}\left[\left(J_1 + F_1 b_1^T\right) - J_*\right)\right]\right\|}\\ &\le 2\|J(x_*)^{-1}\|\\ &\le 2\mu.\end{aligned} \tag{27}$$

Thus x_2 is well defined. From our method, we get

$$\begin{aligned}x_2 - x_* &= x_1 - x_* - \left(J_1 + F_1 b_1^{\mathrm{T}}\right)^{-1} F_1\\ &= x_1 - x_* - \left(J_1 + F_1 b_1^{\mathrm{T}}\right)^{-1}(F_1 - F_*)\\ &= \left(J_1 + F_1 b_1^{\mathrm{T}}\right)^{-1}\left[F_* - F_1 - \left(J_1 + F_1 b_1^{\mathrm{T}}\right)(x_* - x_1)\right].\end{aligned}$$

Furthermore,

$$\begin{aligned}\|x_2 - x_*\| &\le \left\|\left(J_1 + F_1 b_1^{\mathrm{T}}\right)^{-1}\right\| \left\|F_* - F_1 - \left(J_1 + F_1 b_1^{\mathrm{T}}\right)(x_* - x_1)\right\| \\ &\le 2\mu \left\|F_* - F_1 - \left(J_1 + F_1 b_1^{\mathrm{T}}\right)(x_* - x_1)\right\| \\ &\le 2\mu \left(\frac{\gamma}{2}\|(x_* - x_1)\|^2 + \left\|F_1 b_1^{\mathrm{T}}(x_* - x_1)\right\|\right) \\ &\le 2\mu \left(\frac{\gamma}{2}\|(x_* - x_1)\|^2 + \frac{\gamma^2}{2m}\|x_* - x_1\|^2\right) \\ &= \frac{\mu\gamma(m+\gamma)}{m}\|x_* - x_1\|^2.\end{aligned}$$

This proves (24). Taking (25) into consideration leads to

$$\begin{aligned}\|x_2 - x_*\| &\le \frac{\mu\gamma(m+\gamma)}{4m}\|x_0 - x_*\|^2 \\ &\le \frac{\mu\gamma(m+\gamma)}{4m}\varepsilon\|x_0 - x_*\| \\ &< \frac{1}{2}\|x_0 - x_*\|.\end{aligned}$$

Then $x_2 \in N(x_*, r)$ and completes the case $k = 1$. The proof of the induction step proceeds identically. □

Numerical experiments

This section is devoted to the numerical results. First, we show the numerical comparison between Algorithm 1, Newton's method and a third order Newton's method for finding a root of real function. This provides the numerical evidence that Algorithm 1 is better then Newton's method. Secondly, we demonstrate the performance of Algorithm 1 for solving system of nonlinear equations. Algorithm 1 has been applied to some popular test problems and compared with Newton's method and a third order method. All codes were written in Mathematica10.0 and run on a PC with an Intel i7 3.6GHz CPU processor, 4GB memory and 64-bit Windows 7 operating system.

Finding roots of real function

In this subsection we demonstrate the performance of our improved Newton's method for finding the root of real functions $f : \mathbb{R} \to \mathbb{R}$. In other words, we show the efficiency of the new iteration formula (21) in solving a root of the nonlinear equation. Specifically, we chose ten particular nonlinear equations from the literature (Thukral 2016) which are listed in Table 1.

In our tests, the stopping criteria used are $\|F(x_k)\| < 10^{-6}$ or the number of iterations exceeds 100. We compute these 10 problems by using the iteration formula (21), Newton's Method and a third order Newton's Method introduced in Darvishi and Barati (2007a). In our experiments, the initial point for each problem is randomly generated ten times in the range of the initial point, and the average numerical results are listed in Table 2, where

INM denotes the iteration formula (21),

Table 1 Test equations and range of initial point

Equation	Range of initial
$f_1(x) = \exp(x)\sin(x) + \ln(1 + x^2) = 0$	$x_0 \in [-0.1, 1]$
$f_2(x) = \exp(x)\sin(x) + \cos(x)\ln(1 + x) = 0$	$x_0 \in [-1, 1]$
$f_3(x) = \exp(\sin(x)) - x/5 - 1 = 0$	$x_0 \in [-0.5, 1]$
$f_4(x) = (x + 1)\exp(\sin(x)) - x^2\exp(\cos(x)) = 0$	$x_0 \in [-1.5, 1]$
$f_5(x) = \sin(x) + \cos(x) + \tan(x) - 1 = 0$	$x_0 \in [-1, 1]$
$f_6(x) = \exp(-x) - \cos(x) = 0$	$x_0 \in [-1, 0.5]$
$f_7(x) = \ln(1 + x^2) + \exp(x^2 - 3x)\sin(x) = 0$	$x_0 \in [-0.2, 1]$
$f_8(x) = x^3 + \ln(1 + x) = 0$	$x_0 \in [-0.5, 1]$
$f_9(x) = \sin(x) - x/3 = 0$	$x_0 \in [-0.5, 1]$
$f_5(x) = (x - 10)^6 - 10^6 = 0$	$x_0 \in [-1, 1]$

NM	denotes Newton's method,
3NM	denotes the third order Newton's method (Darvishi and Barati 2007a),
It	denotes the average number of iterations,
Re	denotes the average value of $|f(x_k)|$ when the iteration stop,
Fa	denotes the number of failures in solving equations.

From Table 2, in terms of the number of iterations, the efficiency of the improved Newton formula (21) is better than Newton's method, but not as good as the third order method.

To compare the performance of the iteration formula (21), Newton's method and the third order method (Darvishi and Barati 2007a), we consider the performance profile introduced in Dolan and More (2002) as a means. We assume that there are n_s solvers and n_p test problems from the test set $\mathcal{P}$ which is chosen from Table 1. The initial point is selected randomly from the range of the initial point. We are interested in using the iteration number as a measure of performance for the iteration formula (21), NM and 3NM. For each problem p and solver s, let

$$f_{p,s} = \text{iteration number required to solve problem } p \text{ by solver } s.$$

Table 2 Numerical experiment results of INM, NM and 3NM

Equation	INM			NM			3NM		
	It	Re	Fa	It	Re	Fa	It	Re	Fa
f_1	3.3	9.1830E−8	0	4.5	7.6091E−8	0	3.2	1.1458E−7	0
f_2	3.0	2.5350E−9	0	4.0	9.4571E−10	0	3.0	1.3982E−13	0
f_3	2.8	1.5251E−8	0	3.1	1.7424E−7	0	2.3	1.9551E−7	0
f_4	3.1	4.0420E−8	0	3.5	2.1129E−9	0	2.7	4.9111E−8	0
f_5	3.0	1.1038E−8	0	3.6	5.9695E−9	0	2.6	4.9111E−8	0
f_6	3.7	2.4752E−8	0	4.3	1.1354E−7	0	2.9	9.2102E−8	0
f_7	3.3	1.400E−8	0	3.9	1.2565E−7	0	2.6	4.6376E−8	0
f_8	3.2	1.3959E−7	0	3.7	1.2625E−9	0	2.6	2.2066E−9	0
f_9	3.7	2.7669E−9	0	5.8	5.1190E−8	0	2.6	4.3207E−8	0
f_{10}	3.2	5.4249E−9	0	4.3	1.2257E−7	0	2.9	1.3057E−7	0

We employ the performance ratio

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

where $\mathcal{S}$ is the three solvers set. 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 does not solve problem p. In order to obtain an overall assessment for each solver, we define

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

which is called the performance profile of the number of iterations for solver s. Then, $\rho_s(\tau)$ is the probability for solver $s \in \mathcal{S}$ that a performance ratio $f_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio.

Figure 1 shows the performance profile of iteration numbers in the range of $\tau \in [1, 2]$ for three solvers on 200 test problem which were selected from Table 1 with random initial points. From this figure, we see that the numerical performance of solver INM is between 3NM and NM. In summary, from the viewpoint of iteration numbers, we conclude that

$$\text{3NM} > \text{INM} > \text{NM},$$

where "$>$" means "better performance".

Solving system of nonlinear equations

In this subsection we show the numerical efficiency of Algorithm 1 for solving system of nonlinear equations. Listed in Table 3 are the 12 multivariable test problems that were chosen from the test problems set (Dennis and Schnabel 1993; Moré et al. 1981; Andrei 2008). The starting points for each problem are the standard starting points. Illustrative examples further demonstrate the superiority of our proposed algorithm. The numerical results are listed in Table 4, where

INM	denotes Algorithm 1,
NM	denotes Newton's method,
3NM	denotes the third order Newton method (Darvishi and Barati 2007a),
Dim	denotes the size of problem,

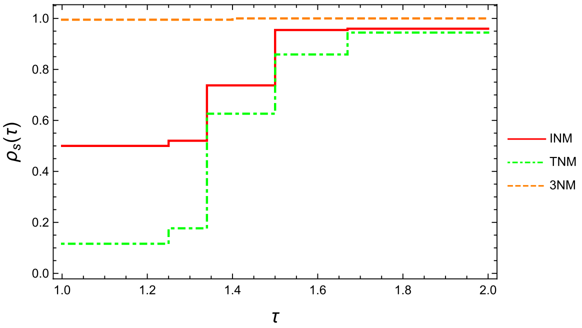

Fig. 1 Performance profile of iteration numbers of INM, NM and 3NM

Table 3 Test problems

Function	Name	Function	Name
F0	Rosenbrock	F1	Powell badly scaled
F2	Freudenstein and Roth	F3	Powell singular
F4	Trigonometric	F5	Trigonometric exponential
F6	Trigexp	F7	Broyden tridiagonal
F8	Extend Power singular	F9	Discrete boundary
F10	Discrete integral equation	F11	Broyden banded

Table 4 Numerical experiment results of INM, NM and 3NM

Pr	Dim	INM		NM		3NM		Pr	Dim	INM		NM		3NM	
		It	Ti	It	Ti	It	Ti			It	Ti	It	Ti	It	Ti
F0	2	2	0.1E−8	2	0.1E−8	1	0.1E−8	F1	2	7	0.001	11	0.001	8	0.1E−8
F2	2	27	0.005	42	0.005	–	–	F3	2	9	0.001	11	0.001	8	0.001
F4	10	6	0.008	7	0.007	–	–	F5	10	8	0.007	10	0.007	19	0.021
F4	50	5	0.349	9	0.641	–	–	F5	50	8	0.121	10	0.187	19	2.589
F4	100	5	2.193	9	3.811	–	–	F5	100	8	0.858	10	0.998	19	3.661
F4	500	6	0.008	7	0.007	–	–	F5	500	8	61.71	10	75.82	19	141.9
F6	10	5	0.015	5	0.001	4	0.015	F7	10	6	0.1E−8	7	0.1E−8	–	–
F6	50	5	0.140	5	0.125	4	0.125	F7	50	8	0.078	10	0.094	–	–
F6	100	5	0.702	5	0.764	4	0.733	F7	100	9	0.533	11	0.633	–	–
F6	500	5	37.39	5	35.820	4	32.00	F7	500	11	59.18	14	75.53	–	–
F8	8	11	0.003	13	0.003	9	0.004	F9	10	2	0.002	2	0.002	2	0.002
F8	60	11	0.212	13	0.239	10	0.209	F9	50	2	0.020	2	0.020	1	0.015
F8	100	11	0.854	13	0.953	10	0.837	F9	100	2	0.140	2	0.136	1	0.082
F8	500	12	71.61	14	83.23	10	62.05	F9	500	1	6.515	1	6.038	1	5.966
F10	10	2	0.004	2	0.004	2	0.005	F11	10	5	0.005	5	0.004	4	0.005
F10	50	2	0.291	2	0.278	2	0.005	F11	50	5	0.114	5	0.123	4	0.143
F10	100	2	2.183	3	3.216	2	3.087	F11	100	5	0.567	5	0.623	4	0.630
F10	500	2	284.2	2	355.3	2	352.8	F11	500	5	33.84	5	32.76	4	29.70

It denotes the number of iterations,
Ti denotes the value of the CPU time in seconds,
– denotes that the number of iterations exceeded 100.

It is observed from Table 4 that in terms of the number of iterations and computation time, the efficiency of Algorithm 1 is better than Newton's method for most of the testing problems, and the efficiency of Algorithm 1 is close to the third order convergence method 3NM (Darvishi and Barati 2007a).

The above experiments were conducted on the standard initial point. We then also need to test the three methods for test problems (Table 3) at random starting points. In particular, starting points for each problem are randomly chosen 10 times from a box surrounding the standard starting points. In order to obtain an overall assessment for the three methods, we are also interested in using the number of iterations as a performance measure for Algorithm 1, Newton's method and the third order method (Darvishi and Barati 2007a). The performance plot based on iteration number is presented

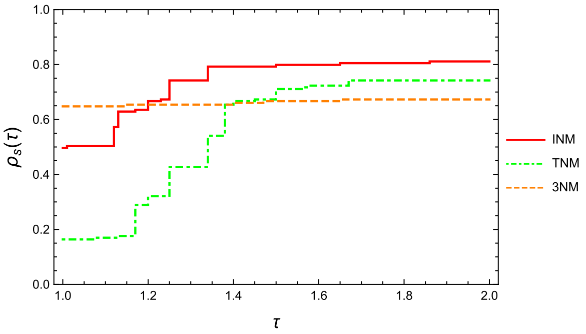

Fig. 2 Performance profile of INM, NM and 3NM

in Fig. 2. From this figure, we can see that Algorithm 1 has the best performance for $\tau > 1.3$. Again, from the viewpoint of large test problems with a perturbed initial point, we conclude that Algorithm 1 is better than Newton's method or the third order method (Darvishi and Barati 2007a).

Conclusion

In this paper, we present an improved Newton's method for system of nonlinear equations by re-use of the previous iteration information. In the novel method, the function value of the previous iteration point was utilized for correcting the Newton direction. The proposed new method also has the quadratic convergence property. From the numerical results obtained for a set of standard test problems, it appears that the rank one revised implementation scheme described, where the Jacobian matrix is updated by a rank one matrix, may allow considerable computational savings for iteration number and computing time. Moreover, two kinds of numerical comparisons are presented in this paper. The first one is the numerical comparison between the new Newton formula, Newton's method and a third order Newton method for finding roots of scalar functions. From this comparison we see that the proposed algorithm is efficient for one dimensional real function. The second comparison is for multivariate vector equations. From this comparison we see that the numerical performance of the proposed algorithm in the case of multidimensional is better than the one-dimensional case. This is a very interesting discovery which may be helpful in other contexts.

Authors' contributions

BS and GC conceived and designed the study; YS organized the manuscript; BS and CW performed the numerical experiments. All authors read and approved the final manuscript.

Author details

[1] College of Mathematics Science, Inner Mongolia University, Hohhot 010021, China. [2] College of Mathematical Science, Inner Mongolia Normal University, Hohhot 010022, China. [3] College of Mechanical Engineering and Applied Electronics Technology, Beijing University of Technology, Beijing 100124, China.

Acknowledgements

This paper was partly supported by the Natural Science Foundation of Inner Mongolia (Award Number: 2014MS0119, 2014MS0102) and the China National Funds for Distinguished Young Scientists (Award Number: 11401326).

Competing interests

The authors declare that they have no competing interests.

Received: 27 April 2016 Accepted: 25 July 2016
Published online: 05 August 2016

References

Andrei N (2008) An unconstrained optimization test functions collection. Adv Model Optim 10:147–161

Ariyawansa KA (1990) Deriving collinear scaling algorithms as extensions of quasi-Newton methods and the local convergence of DFP-and BFGS-related collinear scaling algorithms. Math Program 49:23–48

Ariyawansa KA, Lau DTM (1992) Local and Q-superlinear convergence of a class of collinear scaling algorithms that extends quasi-newton methods with broyden's bounded class of updates. Optimization 23(4):323–339

Cauchy AL (1829) *Sur la détermination approximative des racines d'une équation algébrique ou transcendante*. In: Lecons sur le calcul differentiel, Buré fréres, Paris, pp 575–600

Darvishi MT, Barati A (2007) A third-order Newton-type method to solve system of nonlinear equations. Appl Math Comput 187:630–635

Darvishi MT, Barati A (2007) Afourth-order method from quadrature formulae to solve systems of nonlinear equations. Appl Math Comput 188:257–261

Davidon WC (1980) Conic approximation and collinear Horizontal for optimizer. SIAM J Numer Anal 17:268–281

Dembo RS, Eisenstat SC, Steihaug T (1982) Inexact newton methods. SIAM J Numer Anal 19(2):400–408

Deng NY, Li ZF (1995) Some global convergence properties of a conic-variable metric algorithm for minimization with inexact line searches. Optim Methods Softw 5(1):105–122

Dennis JE, Schnabel RB (1993) Numerical methods for unconstrained optimization and nonlinear equations. SIAM, Philadelphia

Dolan ED, More JJ (2002) Benchmarking optimization software with performance profiles. Math Program 91:201–213

Fine HB (1916) On Newton's method of approximation. Proc Natl Acad Sci USA 2(9):546–552

Fourier JBJ (1890) *Question d'analyse algébrique*. In: Oeuvres complétes(2), Gauthier-Villars, Paris, pp 243–253

Frontini M, Sormani E (2004) Third-order methods from quadrature formulae for solving systems of nonlinear equations. Appl Math Comput 149:771–782

Gourgeon H, Nocedal J (1985) A conic algorithm for optimization. SIAM J Sci Stat Comput 6(2):253–267

Grau-Sánchez M, Grau A, Noguera M (2011) On the computational efficiency index and some iterative methods for solving systems of nonlinear equations. J Comput Appl Math 236:1259–1266

Homeier HHH (2004) Amodified Newton method with cubic convergence: the multivariable case. J Comput Appl Math 169:161–169

Kantorovich LL (1948) On Newton's method for functional equations. Dokl Akad Nauk SSSR 59:1237–1240

Kelley CT (2003) Solving nonlinear equations with Newton's method. SIAM, Philadelphia

Moré JJ, Garbow BS, Hillstrom KE (1981) Testing unconstrained optimization software. ACM Trans Math Softw 7:17–41

Noor MA, Waseem M (2009) Some iterative methods for solving a system of nonlinear equations. Comput Math Appl 57:101–106

Ortega JM, Rheinboldt WC (1970) Iterative solution of nonlinear equations in several variables. Academic Press, New York

Petković MS, Neta B, Petković LD, Džunić J (2013) Multipoint methods for solving nonlinear equations. Elsevier, Amsterdam

Petković MS, Neta B, Petković LD, Džunić J (2013) Multipoint methods for solving nonlinear equations: a survy. Appl Math Comput 226:635–660

Qi L, Sun J (1993) A nonsmooth version of Newton's method. Math Program 58(1–3):353–367

Sharma JR, Guha RK, Sharma R (2013) An efficient fourth order weighted-Newton method for systems of nonlinear equations. Numer Algorithms 62:307–323

Sheng S (1995) Interpolation by conic model for unconstrained optimization. Computing 54:83–98

Sorensen DC (1980) The q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization. SIAM J Numer Anal 17(1):84–114

Sui Y, Saheya, Chen, G (2014) An improvement for the rational approximation RALND at accumulated two-point information. Math Numer Sinica 36(1):51–64

Thukral R (2016) New modification of Newton method with third-order convergence for solving nonlinear equations of type $f(0) = 0$. Am J Comput Appl Math 69(1):14–18

Traub JF (1964) Iterative method for the solution of equations. Prentice-Hall, Englewood Cliffs