Numerical Algorithms (2023) 94:1103–1123
https://doi.org/10.1007/s11075-023-01529-6

ORIGINAL PAPER

Inexact free derivative quasi-Newton method for large-scale nonlinear system of equations

C. A. Arias[1] · C. Gómez[2]

Received: 20 May 2022 / Accepted: 27 February 2023 / Published online: 27 June 2023

Abstract

In this work, we propose a free derivative quasi-Newton method for solving large-scale nonlinear systems of equation. We introduce a two-stage linear search direction and develop its global convergence theory. Besides, we prove that the method enjoys superlinear convergence rate. Finally, numerical experiments illustrate that the proposed method is competitive with respect to Newton-Krylov methods and other well-known methods for solving large-scale nonlinear systems of equations.

Keywords Quasi-Newton methods · Free derivative methods · Linear search · Global convergence · Superlinear convergence

Mathematics Subject Classification (2010) 90C53 · 90C56 · 93A15 · 49M15

1 Introduction

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a nonlinear continuously differentiable function, we are interested in solving the problem

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

especially when n is considerably large.

The most popular method for solving (1) is the *Newton method* [1, 2]. This method has very good convergence properties, but its greatest difficulty is having to calculate the Jacobian matrix of F, F', and evaluate it in each iteration which is, computation-

C.A. Arias and C. Gómez contributed equally to this work.

✉ C. A. Arias
carlosarias@unicauca.edu.co

C. Gómez
chcgomezmo@gmail.com

1 Departamento de Matemáticas, Universidad del Cauca, Popayán, Colombia

2 Facultad de Ciencias Básicas e Ingeniería, Universidad de los Llanos, Villavicencio, Colombia

ally, very expensive. One strategy to avoid this calculus and the evaluations is to use an approximation to $F'(\boldsymbol{x})$ which entails to a variety of methods known as *quasi-Newton methods* [3–8].

However, both Newton and quasi-Newton methods need to solve a linear system of equations in their routines; thus, if n is large, even quasi-Newton methods are expensive. It is important to say that although many quasi-Newton methods admit a cheap formula for updating the inverse of B_k, generally these formulas depend on a term that involves a fraction whose denominator could be zero or a very small quantity so in practice this could lead to an ill-conditioned matrix. To counter this, iterative Krylov methods [9–11] were introduced to Newtonian algorithms, which contributed to decrease the computational cost on these algorithms. This strategy consists of finding approximately the search direction, so the mentioned linear system will solve with some tolerable error.

For Newton-Krylov methods, fast convergence properties have been proved whereas, for quasi-Newton-Krylov methods, have been proved fast convergence properties if the Krylov method is the conjugate gradient method [12–15] or employed a Jacobian restart strategy [16]. The greatest difficulty of inexact quasi-Newton methods is that so far it has not been possible to prove that inexact quasi-Newton direction are descent directions for the associated merit function to (1).

Several studies are concerned with solving (1). In [17, 18], the authors proposed a different approach to solve (1). They proposed a nonmonotone spectral free derivative method. This was an ingenious proposal based on spectral gradient method and systematically uses $\pm F(\boldsymbol{x}_k)$ as a search direction. Due to the simplicity of this method, the computational cost is very low and although global convergence has been proved, linear or superlinear convergence has not yet been proved.

In this paper, we establish a global inexact quasi-Newton method, which is of low computational cost to solve (1). We propose a two-stage linear search procedure for obtaining descent direction and we ensure the global convergence without falling into infinite cycles. Also, under reasonable conditions, we prove fast convergence properties for the method.

This paper is organized as follows. In Section 2 we introduce the new algorithm and make some remarks. In Section 3 we develop the convergence theory of the algorithm introduced previously. Besides, we prove global convergence of the method and the linear and superlinear convergence rate. In Section 4 we present some numerical experiments that show the robustness and competitiveness of the new algorithm. Furthermore, we compare the performance of our algorithm with respect to the algorithms proposed in [17, 18]. Finally, we make some remarks in Section 5.

2 Algorithm

In this section, we describe the inexact free derivative quasi-Newton method (IFDQ) that we propose in this work.

Taking into account that one of our interests is to purpose a global algorithm to solve (1), we considered the following minimization problem to this.

$$\underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{x}) \tag{2}$$

where $f(\boldsymbol{x}) = \frac{1}{2}\|F(\boldsymbol{x})\|^2$ is the associated merit function to (1) and $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$.

Below we show the IFDQ algorithm which main innovation is the two-stage linear search procedure

Algorithm 1 IFDQ.

Require: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\lambda \in \left(0, \frac{1}{3}\right)$, $\beta \in (0,1)$, $\theta \in (0,1)$ and a nonsingular $B_0 \in \mathbb{R}^{n\times n}$.

Step 1: Find $\boldsymbol{d}_k$ such that

$$\|B_k\boldsymbol{d}_k + F(\boldsymbol{x}_k)\| \leq \theta_k \|F(\boldsymbol{x}_k)\|, \quad \theta_k \in (0,\theta). \tag{3}$$

Step 2: **Two stages linear search procedure:** Set $\alpha_k = 1$.

1. **if** $\|F(\boldsymbol{x}_k + \alpha_k\boldsymbol{d}_k)\| < (1-\lambda\alpha_k)\|F(\boldsymbol{x}_k)\|$ **then**
2. $\quad\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{d}_k$ and go to Step 3.
3. **else if** $\|F(\boldsymbol{x}_k - \alpha_k\boldsymbol{d}_k)\| < (1-\lambda\alpha_k)\|F(\boldsymbol{x}_k)\|$ **then**
4. $\quad\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\boldsymbol{d}_k$ and go to Step 3.
5. **else**
   , set $\alpha_k = \beta\alpha_k$. If $\alpha_k \geq \lambda$, go to line 1 in the two stages linear search and repeat the procedure, else, break.
6. **end if**

Step 3: Update B_k such that B_{k+1} let be a nonsingular matrix.

Remark 1 *So far it has not been possible to prove, in general, that the inexact quasi-Newton direction, obtained in Step 1 is a descent direction for the merit function f given in (2). For this reason, it is necessary to try both $\boldsymbol{d}_k$ and $-\boldsymbol{d}_k$ directions in the line search procedure.*

Remark 2 *In Step 2 we set two trial directions, $\boldsymbol{d}_k$ and $-\boldsymbol{d}_k$. Note that unless $\nabla f^T\boldsymbol{d}_k = 0$, at least one of the trial directions will be a descent direction, hence, the two-stage linear search procedure will not fall in an infinite cycle.*

Remark 3 *In the linear search procedure we seek a sufficient decrease of the merit function f but if the step length α_k is too small then the algorithm breaks down, avoiding small steps with a poor decrease. Note that the lower bound for the step length is set by the user and can be as small as desired. So to prevent either, many breaks of the algorithm and small steps with a poor decrease we recommend to take $\lambda = 10^{-4}$. Whit this value, the algorithm showed a good performance.*

3 Convergence theory

In this section, we present the main theoretical results obtained for IFDQ algorithm. In the following lemmas and theorems we show that, under reasonable assumptions, the algorithm converges to a solution of the problem (1) and locally enjoys good convergence properties: full inexact quasi-Newton step is accepted and it can even has until superlinear converge.

The hypotheses under which we develop the convergence theory of the proposed algorithm are:

H1. There exist $\mathbf{x}^* \in \mathbb{R}^n$ such that $F(\mathbf{x}^*) = 0$.
H2. There exists $T > 0$ such that $\|B_k^{-1}\| < T$ for all $k \geq 0$.
H3. $F'(\boldsymbol{x})$ is a Lipschitz function, i.e., there exist $L > 0$ such that

$$\|F'(\boldsymbol{x}) - F'(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\| \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

H4. $\{B_k\}$ is a sequence such that

$$\lim_{k\to\infty} \frac{\|(B_k - F'(\boldsymbol{x}_k))\boldsymbol{d}_k\|}{\|\boldsymbol{d}_k\|} = 0. \tag{4}$$

Previous hypotheses are classical hypotheses for quasi-Newton methods. Hypotheses **H1** and **H3** depend on the problem to be solved whereas **H2** and **H4** depends on the approximations to $F'(\boldsymbol{x}_k)$.

An immediate consequence of **H3** is that for all $\boldsymbol{x}, \ \boldsymbol{y} \in \mathbb{R}^n$,

$$\|F(\boldsymbol{x}) - F(\boldsymbol{y}) - F'(\boldsymbol{y})(\boldsymbol{x} - \boldsymbol{y})\| \leq \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{5}$$

It is important to mention that (4) in **H4** is known as Dennis-Moré condition.

The first lemma of our theoretical development ensures that if the IFDQ algorithm does not break down then, it generates a sequence such that its images converge to zero.

Lemma 1 *If* $\{\boldsymbol{x}_k\}$ *is a sequence generated by IFDQ algorithm then* $\lim_{k\to\infty} \|F(\boldsymbol{x}_k)\| = 0$.

Proof Let $\{\boldsymbol{x}_k\}$ be a sequence given by IFDQ algorithm, it follows that

$$\begin{aligned} \|F(\boldsymbol{x}_{k+1})\| &\leq (1 - \alpha_k\lambda)\,\|F(\boldsymbol{x}_k)\| \\ &\leq (1 - \lambda^2)\,\|F(\boldsymbol{x}_k)\| \\ &\leq (1 - \lambda^2)^{k+1}\,\|F(\boldsymbol{x}_0)\|. \end{aligned}$$

By recalling that $1 - \lambda^2 < 1$, the result is established. □

The next result is immediate since F is a continuous function.

Corollary 1 *If* $\{\boldsymbol{x}_k\}$ *is a sequence generated by IFDQ algorithm and* $\boldsymbol{x}^*$ *is a cluster point of* $\{\boldsymbol{x}_k\}$ *then* $\boldsymbol{x}^*$ *is a solution of* (1).

The following theorem ensures the convergence of the sequence generated by IFDQ algorithm under the assumption of non-singularity of the Jacobian matrix F' in the solution of the problem.

Theorem 1 *Assume* ***H2****. Let* $\{\boldsymbol{x}_k\}$ *be a sequence generated by IFDQ algorithm. If* $\boldsymbol{x}^* \in \mathbb{R}^n$ *is a cluster point of* $\{\boldsymbol{x}_k\}$ *such that* $F'(\boldsymbol{x}^*)$ *is nonsingular then* $F(\boldsymbol{x}^*) = 0$ *and* $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}^*$.

Proof Let $\{\boldsymbol{x}_{k_j}\}$ be a subsequence of $\{\boldsymbol{x}_k\}$ such that $\boldsymbol{x}_{k_j} \to \boldsymbol{x}^*$ as $k_j \to \infty$. By recalling that F is a continuous function and Lemma 1, we have that $F(\boldsymbol{x}^*) = 0$.

On the other hand, let $K = \left\| F'(\boldsymbol{x}^*)^{-1} \right\|$ and $\delta > 0$ small enough such that for any $\boldsymbol{y} \in B(\boldsymbol{x}^*, \delta)$ we have that

i $F'(\boldsymbol{y})^{-1}$ there exists.
ii $\left\| F'(\boldsymbol{y})^{-1} \right\| < 2K$.
iii $\left\| F(\boldsymbol{y}) - F(\boldsymbol{x}^*) - F'(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*) \right\| \le \frac{1}{2K} \|\boldsymbol{y} - \boldsymbol{x}^*\|$.

The existence of δ can be guaranteed thanks to Lemmas 1.1 and 1.2 in [15]. Observe that if $\boldsymbol{y} \in B(\boldsymbol{x}^*, \delta)$ then

$$
\begin{aligned}
\|F(\boldsymbol{y})\| &= \|F'(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*) + F(\boldsymbol{y}) - F(\boldsymbol{x}^*) - F'(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*)\| \\
&\ge \|F'(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*)\| - \|F(\boldsymbol{y}) - F(\boldsymbol{x}^*) - F'(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*)\| \\
&\ge \|F'(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*)\| - \frac{1}{2K}\|\boldsymbol{y} - \boldsymbol{x}^*\|
\end{aligned}
$$

and

$$
\left\|\boldsymbol{y} - \boldsymbol{x}^*\right\| = \left\| F'(\boldsymbol{x}^*)^{-1} F'(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*) \right\| \le K \left\| F'(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*) \right\|.
$$

By combining the two last inequalities we can infer that

$$
\left\|\boldsymbol{y} - \boldsymbol{x}^*\right\| \le 2K \|F(\boldsymbol{y})\|, \quad \forall \boldsymbol{y} \in B(\boldsymbol{x}^*, \delta). \tag{6}
$$

On the other hand, let $\epsilon \in (0, \delta/4)$. Since $\boldsymbol{x}^*$ is a cluster point of $\{\boldsymbol{x}_k\}$ and $F(\boldsymbol{x}^*) = 0$ then, there exist k large enough such that

$$
\boldsymbol{x}_k \in S_\epsilon := \left\{ \boldsymbol{y} \in \mathbb{R}^n : \boldsymbol{y} \in B(\boldsymbol{x}^*, \delta/2), \quad M(1+\theta)\|F(\boldsymbol{y})\| < \epsilon \right\},
$$

where $M = \max\{K, T\}$ and T is the constant of hypothesis **H2**. Hence, we obtain

$$
\begin{aligned}
\|\boldsymbol{d}_k\| &= \left\| B_k^{-1}[-F(\boldsymbol{x}_k) + F(\boldsymbol{x}_k) + B_k \boldsymbol{d}_k] \right\| \\
&\le \left\| B_k^{-1} \right\| (\|F(\boldsymbol{x}_k)\| + \|F(\boldsymbol{x}_k) + B_k \boldsymbol{d}_k\|) \\
&\le T\,(\|F(\boldsymbol{x}_k)\| + \theta_k \|F(\boldsymbol{x}_k)\|) \\
&\le T(1+\theta_k)\|F(\boldsymbol{x}_k)\| \\
&< M(1+\theta_k)\|F(\boldsymbol{x}_k)\|.
\end{aligned} \tag{7}
$$

Now, since $\boldsymbol{x}_k \in S_\epsilon$ then $\|\boldsymbol{d}_k\| < \epsilon$. Moreover, we have that

$$\begin{aligned}\left\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\right\| &= \left\|\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k - \boldsymbol{x}^*\right\| \\ &\leq \left\|\boldsymbol{x}_k - \boldsymbol{x}^*\right\| + |\alpha_k| \, \|\boldsymbol{d}_k\| \\ &< \delta.\end{aligned}$$

Thus, we conclude $\boldsymbol{x}_{k+1} \in B(\boldsymbol{x}^*, \delta)$. On the other hand, since IFDQ attempts for a monotone decrease and $\boldsymbol{x}_k \in S_\varepsilon$ then

$$\|F(\boldsymbol{x}_{k+1})\| \leq \|F(\boldsymbol{x}_k)\| \leq \frac{\epsilon}{M(1+\theta)}. \tag{8}$$

So, from (6) and using the last inequality we can infer that

$$\begin{aligned}\left\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\right\| &\leq \frac{2K\epsilon}{M(1+\theta)} \\ &\leq \frac{2\epsilon}{(1+\theta)} \\ &\leq 2\epsilon.\end{aligned} \tag{9}$$

Finally, from (8) and (9) we have that $\boldsymbol{x}_{k+1} \in S_\epsilon$, with which we prove that $\boldsymbol{x}_k \in S_\epsilon$ for all k large enough, and since $\|F(\boldsymbol{x}_k)\| \to 0$ then, from (6), $\boldsymbol{x}_k \to \boldsymbol{x}^*$ as $k \to \infty$. □

In the next lemma, we show that the trial directions in Step 1 remain bounded for all k.

Lemma 2 *Assume* ***H2***. *Let* $\{\boldsymbol{x}_k\}$ *be a sequence generated by IFDQ algorithm then,* $\|\boldsymbol{d}_k\| \leq 2T \|F(\boldsymbol{x}_k)\|$ *and* $\lim_{k\to\infty} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| = 0$.

Proof The first part of this lemma follows from (7) and the fact that $\theta_k \in (0, 1)$.

On the other side, observe that

$$\begin{aligned}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| &= \|\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k - \boldsymbol{x}_k\| \\ &= \alpha_k \|\boldsymbol{d}_k\| \\ &\leq 2T \|F(\boldsymbol{x}_k)\|.\end{aligned}$$

Thus, by Lemma 1 and the last inequality, we have the desired result. □

The next theorem ensures convergence of the sequence generated by IFDQ algorithm without non-singularity condition of $F'(\boldsymbol{x}^*)$.

Theorem 2 *Let* $\{\boldsymbol{x}_k\}$ *be a sequence generated by the algorithm. If* $\boldsymbol{x}^*$ *is an isolated cluster point of the sequence then,* $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}^*$.

Proof Taking into account Lemmas 1 and 2, this proof follows the same ideas of the proof of theorem 3 in [18] □

The next theorem ensures that, at least locally, the IFDQ algorithm shows good performance in the sense that the full quasi-Newton step will be accepted in the two-stage linear search procedure. In the proof we assume a weaker hypothesis than the Dennis-Moré condition. This hypothesis is related with the bounded deterioration property and ensures that at least, for all k large enough, the approximation B_k to the jacobian matrix $F'(\boldsymbol{x}_k)$ is bounded above by a constant.

As in the proof of Theorem 1, let $M = \max\{K, T\}$ where T is the constant in **H2**.

Theorem 3 *Assume hypotheses **H1**, **H2** and **H3**. Let $\{\boldsymbol{x}_k\}$ be a sequence generated by IFDQ algorithm and $\boldsymbol{x}^*$ be a cluster point of the sequence. If $F'(\boldsymbol{x}^*)$ is nonsingular and for all k large enough $\theta_k < \frac{1}{3} - \lambda$ and $\|F'(\boldsymbol{x}_k) - B_k\| < \frac{1}{24M^3}$ then $\boldsymbol{d}_k$ and $\alpha_k = 1$ will be accepted, in the two-stage linear search procedure.*

Proof Observe that

$$\begin{aligned} F(\boldsymbol{x}_k + \boldsymbol{d}_k) &= F(\boldsymbol{x}_k) + \int_0^1 F'(\boldsymbol{x}_k + t\boldsymbol{d}_k)\boldsymbol{d}_k \, dt \\ &= F(\boldsymbol{x}_k) + F'(\boldsymbol{x}_k)\boldsymbol{d}_k + B_k\boldsymbol{d}_k - B_k\boldsymbol{d}_k \\ &\quad + \int_0^1 \left[F'(\boldsymbol{x}_k + t\boldsymbol{d}_k)\boldsymbol{d}_k - F'(\boldsymbol{x}_k)\boldsymbol{d}_k\right] dt. \end{aligned}$$

Thus,

$$\begin{aligned} \|F(\boldsymbol{x}_k + \boldsymbol{d}_k)\| &\leq \|F(\boldsymbol{x}_k) + B_k\boldsymbol{d}_k\| + \left\|(F'(\boldsymbol{x}_k) - B_k)\boldsymbol{d}_k\right\| \\ &\quad + \int_0^1 \left\|F'(\boldsymbol{x}_k + t\boldsymbol{d}_k) - F'(\boldsymbol{x}_k)\right\| \|\boldsymbol{d}_k\| \, dt \\ &\leq \theta_k \|F(\boldsymbol{x}_k)\| + \left\|F'(\boldsymbol{x}_k) - B_k\right\| \|\boldsymbol{d}_k\| + \int_0^1 L \|t\boldsymbol{d}_k\| \|\boldsymbol{d}_k\| \, dt \end{aligned}$$

so, from Lemma 2 and taking into account that for all k large enough

$$\|F'(\boldsymbol{x}_k) - B_k\| < \frac{1}{24M^3} < \frac{1}{6T}$$

we have that

$$\begin{aligned} \|F(\boldsymbol{x}_k + \boldsymbol{d}_k)\| &\leq \theta_k \|F(\boldsymbol{x}_k)\| + \frac{1}{3} \|F(\boldsymbol{x}_k)\| + \frac{L}{2} \|\boldsymbol{d}_k\|^2 \\ &\leq \theta_k \|F(\boldsymbol{x}_k)\| + \frac{1}{3} \|F(\boldsymbol{x}_k)\| + 2T^2 L \|F(\boldsymbol{x}_k)\|^2 \\ &= \left(\theta_k + \frac{1}{3} + 2T^2 L \|F(\boldsymbol{x}_k)\|\right) \|F(\boldsymbol{x}_k)\|. \end{aligned} \tag{10}$$

Now, since $\|F(\boldsymbol{x}_k)\|$ converges to zero then, for all k large enough

$$\|F(\boldsymbol{x}_k)\| < \frac{1}{6T^2L}. \tag{11}$$

Hence, by (10), (11) and since $\theta_k < \frac{1}{3} - \lambda$ then, for all k large enough

$$\|F(\boldsymbol{x}_k + \boldsymbol{d}_k)\| \leq (1-\lambda)\,\|F(\boldsymbol{x}_k)\|$$

thus $\alpha_k = 1$ and $\boldsymbol{d}_k$ will be accepted. □

The following theorem is the first theorem in which we show good convergence properties of the IFDQ algorithm. For this purpose, we assume that $F'(\boldsymbol{x}^*)$ is nonsingular, that $\|F'(\boldsymbol{x}^*)^{-1}\| = K$ and that $M = \max\{K, T\}$ where T is the constant in **H2**.

Theorem 4 *Under the same hypotheses of the previous theorem. If in addition*

$$\theta_k < \min\left\{\frac{1}{12M^2}, \frac{1}{3} - \lambda\right\}$$

then $\boldsymbol{x}_k \to \boldsymbol{x}^*$ *linearly.*

Proof By using Theorem 3, we can ensure that the full quasi-Newton step will be accepted for all k large enough and by Theorem 1, $\boldsymbol{x}_k \to \boldsymbol{x}^*$, so for all k large enough, $F'(\boldsymbol{x}_k)^{-1}$ there exist and $\|F'(\boldsymbol{x}_k)^{-1}\| \leq 2M$ hence,

$$\begin{aligned}
\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| &= \|\boldsymbol{x}_k + \boldsymbol{d}_k - \boldsymbol{x}^* + F'(\boldsymbol{x}_k)^{-1}F(\boldsymbol{x}_k) - F'(\boldsymbol{x}_k)^{-1}F(\boldsymbol{x}_k)\| \\
&\leq \|\boldsymbol{x}_k - \boldsymbol{x}^* - F'(\boldsymbol{x}_k)^{-1}F(\boldsymbol{x}_k)\| + \|F'(\boldsymbol{x}_k)^{-1}(F(\boldsymbol{x}_k) + F'(\boldsymbol{x}_k)\boldsymbol{d}_k)\| \\
&\leq \|F'(\boldsymbol{x}_k)^{-1}[F(\boldsymbol{x}^*) - F(\boldsymbol{x}_k) - F'(\boldsymbol{x}_k)(\boldsymbol{x}^* - \boldsymbol{x}_k)]\| + \\
&\qquad \|F'(\boldsymbol{x}_k)^{-1}\|\|F(\boldsymbol{x}_k) + F'(\boldsymbol{x}_k)\boldsymbol{d}_k + B_k\boldsymbol{d}_k - B_k\boldsymbol{d}_k\| \\
&\leq 2M\|F(\boldsymbol{x}^*) - F(\boldsymbol{x}_k) - F'(\boldsymbol{x}_k)(\boldsymbol{x}^* - \boldsymbol{x}_k)\| + 2M[\|F(\boldsymbol{x}_k) + B_k\boldsymbol{d}_k\| \\
&\qquad + \|F'(\boldsymbol{x}_k)\boldsymbol{d}_k - B_k\boldsymbol{d}_k\|]
\end{aligned}$$

by (5) and Step 1 in the algorithm we have that

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq ML\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + 2M[\theta_k\|F(\boldsymbol{x}_k)\| + \|F'(\boldsymbol{x}_k) - B_k\|\|\boldsymbol{d}_k\|]. \tag{12}$$

Thus, by Lemma 2,

$$\begin{aligned}
\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| &\leq ML\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + 2M\left[\theta_k + \frac{1}{12M^2}\right]\|F(\boldsymbol{x}_k)\| \\
&= ML\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + 2M\left[\theta_k + \frac{1}{12M^2}\right]\|F(\boldsymbol{x}_k) - F(\boldsymbol{x}^*)\|.
\end{aligned}$$

So, by the mean value theorem,

$$\begin{aligned}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\| &\leq ML\|\boldsymbol{x}_k-\boldsymbol{x}^*\|^2+4M^2\left[\theta_k+\frac{1}{12M^2}\right]\|\boldsymbol{x}_k-\boldsymbol{x}^*\| \\ &= \left[ML\|\boldsymbol{x}_k-\boldsymbol{x}^*\|+4M^2\left(\theta_k+\frac{1}{12M^2}\right)\right]\|\boldsymbol{x}_k-\boldsymbol{x}^*\| \qquad (13)\end{aligned}$$

thereby, for k large enough such that

$$\|\boldsymbol{x}_k-\boldsymbol{x}^*\| < \frac{1}{3ML},$$

since $\theta_k < \frac{1}{12M^2}$, we can conclude that

$$\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\| < R\|\boldsymbol{x}_k-\boldsymbol{x}^*\|$$

where $0 < R < 1$, which completes the proof. □

Finally, to complete our theoretical development, the next theorem ensures, under reasonable assumptions, superlinear convergence of the IFDQ algorithm.

Theorem 5 *Assume **H1**, **H2**, **H3** and **H4**. Let $\{\boldsymbol{x}_k\}$ be a sequence generated by IFDQ algorithm and $\boldsymbol{x}^*$ be a cluster point of the sequence. If $F'(\boldsymbol{x}^*)$ is nonsingular and $\theta_k \to 0$ then $\boldsymbol{x}_k \to \boldsymbol{x}^*$ superlinearly.*

Proof From (12) we can infer that

$$\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\| \leq ML\|\boldsymbol{x}_k-\boldsymbol{x}^*\|^2+2M\left[\theta_k\|F(\boldsymbol{x}_k)\|+\frac{\|F'(\boldsymbol{x}_k)-B_k\|\|\boldsymbol{d}_k\|}{\|\boldsymbol{d}_k\|}\right].$$

By Lemma 2 and the Mean Value Theorem,

$$\begin{aligned}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\| &\leq ML\|\boldsymbol{x}_k-\boldsymbol{x}^*\|^2+2M\left[\theta_k+2M\frac{\|F'(\boldsymbol{x}_k)-B_k\|}{\|\boldsymbol{d}_k\|}\right]\|F(\boldsymbol{x}_k)\| \\ &= ML\|\boldsymbol{x}_k-\boldsymbol{x}^*\|^2+2M\left[\theta_k+2M\frac{\|F'(\boldsymbol{x}_k)-B_k\|}{\|\boldsymbol{d}_k\|}\right]\|F(\boldsymbol{x}_k)-F(\boldsymbol{x}^*)\| \\ &\leq ML\|\boldsymbol{x}_k-\boldsymbol{x}^*\|^2+2M\left[\theta_k+2M\frac{\|F'(\boldsymbol{x}_k)-B_k\|}{\|\boldsymbol{d}_k\|}\right]\|\boldsymbol{x}_k-\boldsymbol{x}^*\| \\ &= \left[ML\|\boldsymbol{x}_k-\boldsymbol{x}^*\|+2M\left(\theta_k+2M\frac{\|F'(\boldsymbol{x}_k)-B_k\|}{\|\boldsymbol{d}_k\|}\right)\right]\|\boldsymbol{x}_k-\boldsymbol{x}^*\|\end{aligned}$$

The desired result follows from hypothesis **H4** and the fact that $\boldsymbol{x}_k \to \boldsymbol{x}^*$ and $\theta_k \to 0$ as $k \to \infty$. □

4 Numerical experiments

In this section we report the numerical results of the IFDQ algorithm when solving twenty problems. Sixteen of the problems were taken from [17] and references therein, the rest of the problems were taken from [19–21]. It is important to say that we did not take into account problems 13, 14, 15 and 18 of [17]. First, problems 13 and 14 include many random parameters that difficult the reproduction of the experiments. Second, the poor performance of the algorithms with problems 15 and 18 did not allow us to draw relevant conclusions.

The experiments were carried out in MATLAB® using an Intel Core2^{TM} laptop with a RAM of 4GB. To evaluate the performance of the IFDQ algorithm, we ran experiments and compared the results with four more algorithms: SANE [17], DF-SANE [18], Ac-DFSANE [22] and NITSOL [23].

SANE and DF-SANE are spectral free derivative algorithms. The descent trial direction at each iteration of these methods is $\pm F(\boldsymbol{x}_k)$ and the main difference between them is the linear search.

Ac-DFSANE is an accelerated version, proposed recently, for the DF-SANE algorithm. This chooses in a very ingenious way the new iterate improving in many times the descent achieved in the linear search.

Finally, NITSOL is a practical and efficient implementation of the classical inexact Newton method with GMRES procedure to find the descent direction. This algorithm approximates derivatives by finite differences when these are not available.

For all algorithms we used $\|F(\boldsymbol{x}_k)\| < 10^{-6}$ and $k < 300$ as stop criteria. For IFDQ algorithm we took $B_0 = I_n$ as initial approximation to $F'(\boldsymbol{x}_0)$; $\theta_k = \frac{1}{k+2}$ as inexact parameter in Step 1 and $\lambda = 10^{-4}$ and $\beta = 0.5$ for the two-stage linear search procedure in Step 2. To find $\boldsymbol{d}_k$ in Step 1, we used GMRES procedure.

In the same way, in Step 3 we used the "good" Broyden update [24]. It is important to say that this update is a least change secant update, thus satisfying the well-known property of bounded deterioration [2, 25] with which Dennis and Schnabel in [2] showed that the sequence of matrices $\{B_k\}$ satisfies the Dennis-Moré condition (4), i.e., the sequence of matrices $\{B_k\}$ satisfies hypothesis **H4**.

On the other hand, SANE, DF-SANE, Ac-DFSANE and NITSOL algorithms were carried out with the same parameters as in respective references given above.

In Table 1, we show the complete list of problems with which we ran our experiments. Starting points were the same as in the respective references.

In Tables 2 and 3 we report the results obtained using the following conventions:

F : functions of Table 1.
Method: algorithm used to solve the problems.
n : size of the problem to solve.
k : number of iterations required for the algorithms to solve each problem.
$Feval$: number of evaluation of function at each problem.
t : cpu time, given in seconds.
$**$: means nonconverge of the algorithm because it infringes the stop criteria.

The results in Tables 2 and 3 showed good performance of the IFDQ algorithm. First because IFDQ algorithm required the same or fewer iterations than its counterparts

Table 1 List of problems

Problem	Reference
1. Exponential function 1	[17]
2. Exponential function 2	[17]
3. Exponential function 3	[17]
4. Diagonal function premultiplied by a quasi-orthogonal matrix	[17, 26]
5. Extended Rosenbrock function	[17, 26]
6. Chandrasekhar's H-equation	[17]
7. Badly scaled augmented Powell's function	[17, 26]
8. Trigonometric function	[17]
9. Singular function	[17]
10. Logarithmic function	[17]
11. Broyden tridiagonal function	[17, 27]
12. Trigexp function	[17, 27]
13. Strictly function 1	[17, 28]
14. Strictly function 2	[17, 28]
15. Zero Jacobian function	[17]
16. Geometric programming function	[17]
17. Extended Wood function	[19]
18. Brent function	[19]
19. Yamamura function	[20]
20. Zhou function	[21], Problem 2.

in 17.5% of the problems. Second, our algorithm converged on thirty-eight of the problems, that is, IFDQ algorithm converged on 95% of the experiments carried out.

It is important to mention that although in general when SANE, DF-SANE, Ac-DFSANE and NITSOL converged, they were faster, in terms of CPU time, than IFDQ, but in most cases, that difference was only for a few seconds. This behavior is due to the fact that SANE, DF-SANE and Ac-DFSANE used $\pm F(\boldsymbol{x}_k)$ as trial directions, so they do not have to solve a linear system of equation like (3) or make matrix vector products as products made for IFDQ to update B_k . On the other side, NITSOL makes a single linear search, so generally requires less evaluations of function than IFDQ. In the same way, NITSOL approximate derivatives by finite differences when these are not available nevertheless we think that this could be affecting the convergence of the method since this was the one with the lowest success rate. For the above, IFDQ seems to be a competitive algorithm.

In the second and third columns of Table 4 we show the percentage of experiments in which each algorithm won in terms of CPU time and number of iterations. In last column of this table we show the percentage of success of each algorithm with the experiments. The results show the IFDQ algorithm as an equilibrated method since it was the most successful without requiring a large number of iterations or CPU time compared to the other algorithms.

Table 2 Problems 1 to 10

F	n	Method	k	Feval	t	F	n	Method	k	Feval	t
		IFDQ	14	29	4.842			IFDQ	7	15	0.412
		SANE	14	30	0.084			SANE	8	18	0.316
	5000	DF-SANE	14	29	0.050		1000	DF-SANE	8	17	0.190
		Ac-DFSANE	19	28	0.100			Ac-DFSANE	4	13	0.209
1		NITSOL	**	**	**	6		NITSOL	3	9	0.547
		IFDQ	13	27	17.353			IFDQ	7	15	27.547
		SANE	13	28	0.234			SANE	8	18	30.138
	10000	DF-SANE	13	27	0.116		10000	DF-SANE	8	17	20.103
		Ac-DFSANE	9	28	0.218			Ac-DFSANE	4	13	20.200
		NITSOL	**	**	**			NITSOL	3	9	52.094
		IFDQ	30	61	0.415			IFDQ	9	19	0.083
		SANE	14	36	0.006			SANE	14	72	0.009
	500	DF-SANE	15	31	0.009		399	DF-SANE	14	29	0.022
		Ac-DFSANE	10	31	0.006			Ac-DFSANE	4	13	0.013
2		NITSOL	2	46	0.031	7		NITSOL	**	**	**
		IFDQ	15	31	0.500			IFDQ	9	19	0.268
		SANE	10	29	0.003			SANE	14	72	0.016
	1000	DF-SANE	10	21	0.003		999	DF-SANE	14	29	0.028
		Ac-DFSANE	7	15	0.018			Ac-DFSANE	4	13	0.016
		NITSOL	2	46	0.046			NITSOL	**	**	**
		IFDQ	3	7	0.003			IFDQ	6	13	2.321
		SANE	3	8	0.003			SANE	5	15	0.041
	100	DF-SANE	3	7	0.003		5000	DF-SANE	5	11	0.025
		Ac-DFSANE	12	37	0.022			Ac-DFSANE	**	**	**

Table 2 continued

F	n	Method	k	Feval	t	F	n	Method	k	Feval	t
3		NITSOL	**	**	**	8		NITSOL	**	**	**
		IFDQ	2	5	0.001			IFDQ	6	13	8.631
		SANE	2	6	0.001			SANE	19	53	0.462
	200	DF-SANE	2	5	0.001		10000	DF-SANE	5	11	0.041
		Ac-DFSANE	2	5	0.001			Ac-DFSANE	**	**	**
		NITSOL	**	**	**			NITSOL	**	**	**
		IFDQ	33	67	0.294			IFDQ	144	289	88.522
		SANE	**	**	**			SANE	16	44	0.122
	399	DF-SANE	**	**	**		5000	DF-SANE	16	33	0.087
		Ac-DFSANE	**	**	**			Ac-DFSANE	11	34	0.146
4		NITSOL	**	**	**	9		NITSOL	12	276	12.453
		IFDQ	32	65	0.909			IFDQ	167	335	381.885
		SANE	97	1940	0.634			SANE	16	45	0.288
	999	DF-SANE	**	**	**		10000	DF-SANE	16	33	0.284
		Ac-DFSANE	**	**	**			Ac-DFSANE	10	31	0.347
		NITSOL	**	**	**			NITSOL	12	276	21.641
		IFDQ	44	89	1.331			IFDQ	6	13	7.900
		SANE	**	**	**			SANE	6	14	0.094
	1000	DF-SANE	**	**	**		10000	DF-SANE	6	13	0.066
		Ac-DFSANE	6	19	0.009			Ac-DFSANE	8	25	0.256

Table 2 continued

F	n	Method	k	Feval	t	F	n	Method	k	Feval	t
5		NITSOL	2	6	0.047	10		NITSOL	5	15	0.234
		IFDQ	44	89	62.331			IFDQ	6	13	17.547
		SANE	**	**	**			SANE	6	14	0.113
	10000	DF-SANE	**	**	**		15000	DF-SANE	6	13	0.075
		Ac-DFSANE	6	19	0.194			Ac-DFSANE	8	25	0.203
		NITSOL	2	6	0.063			NITSOL	5	15	0.422

Table 3 Problems 11 to 20

F	n	Method	k	Feval	t	F	n	Method	k	Feval	t
		IFDQ	40	81	0.953			IFDQ	8	17	0.044
		SANE	22	47	0.003			SANE	8	19	0.075
	1000	DF-SANE	22	45	0.001		50	DF-SANE	8	17	0.066
		Ac-DFSANE	17	52	0.006			Ac-DFSANE	16	49	0.169
11		NITSOL	4	22	0.001	16		NITSOL	14	42	14.172
		IFDQ	30	61	2.165			IFDQ	1	3	0.440
		SANE	22	46	0.001			SANE	1	5	0.471
	2000	DF-SANE	22	45	0.001		500	DF-SANE	1	3	0.463
		Ac-DFSANE	17	52	0.006			Ac-DFSANE	1	4	0.600
		NITSOL	4	22	0.063			NITSOL	42	9	14.172
		IFDQ	23	47	0.172			IFDQ	**	**	**
		SANE	12	29	0.002			SANE	**	**	**
	500	DF-SANE	12	25	0.001		2000	DF-SANE	**	**	**
		Ac-DFSANE	8	25	0.012			Ac-DFSANE	43	130	0.138
12		NITSOL	6	20	0.031	17		NITSOL	20	60	0.329
		IFDQ	22	45	0.471			IFDQ	**	**	**
		SANE	12	29	0.091			SANE	**	**	**
	1000	DF-SANE	12	25	0.068		4000	DF-SANE	**	**	**
		Ac-DFSANE	7	22	0.109			Ac-DFSANE	35	106	0.381
		NITSOL	6	20	0.093			NITSOL	17	51	0.469
		IFDQ	6	13	0.134			IFDQ	13	27	0.325
		SANE	6	14	0.009			SANE	**	**	**
	1000	DF-SANE	6	13	0.003		1000	DF-SANE	**	**	**
		Ac-DFSANE	4	13	0.003			Ac-DFSANE	16	49	0.019

Table 3 continued

F	n	Method	k	Feval	t	F	n	Method	k	Feval	t
13		NITSOL	4	14	0.031	18		NITSOL	**	**	**
		IFDQ	6	13	7.859			IFDQ	13	27	8.946
		SANE	6	14	0.088			SANE	**	**	**
	10000	DF-SANE	6	13	0.059		10000	DF-SANE	**	**	**
		Ac-DFSANE	4	13	0.100			Ac-DFSANE	16	49	0.125
		NITSOL	4	14	0.375			NITSOL	**	**	**
		IFDQ	151	303	0.643			IFDQ	4	9	0.109
		SANE	60	137	0.009			SANE	4	10	0.025
	100	DF-SANE	**	**	**		1000	DF-SANE	4	9	0.013
		Ac-DFSANE	**	**	**			Ac-DFSANE	3	10	0.009
14		NITSOL	4	44	0.067	19		NITSOL	3	9	0.031
		IFDQ	291	583	1.650			IFDQ	4	9	0.134
		SANE	101	251	0.015			SANE	4	10	0.066
	200	DF-SANE	**	**	**		5000	DF-SANE	4	9	0.038
		Ac-DFSANE	**	**	**			Ac-DFSANE	3	10	0.038
		NITSOL	4	52	0.076			NITSOL	3	9	0.109
		IFDQ	13	27	0.091			IFDQ	4	9	1.453
		SANE	13	34	0.015			SANE	4	10	0.009
	500	DF-SANE	14	29	0.006		5000	DF-SANE	4	9	0.028
		Ac-DFSANE	26	79	0.047			Ac-DFSANE	6	19	0.062

Table 3 continued

F	n	Method	k	Feval	t	F	n	Method	k	Feval	t
15		NITSOL	10	30	0.031	20		NITSOL	3	9	0.063
		IFDQ	13	27	0.325			IFDQ	4	9	5.331
		SANE	13	34	0.009			SANE	4	10	0.053
	1000	DF-SANE	15	31	0.006		10000	DF-SANE	4	9	0.046
		Ac-DFSANE	28	85	0.066			Ac-DFSANE	6	19	0.147
		NITSOL	10	30	0.273			NITSOL	3	9	0.047

Table 4 Global convergence

Method	CPU time	Iterations	Success
IFDQ	12.5%	17.5%	95%
SANE	22.5%	12.5%	82.5%
DF-SANE	57.5%	15%	75%
Ac-DFSANE	27.5%	32.5%	85%
NITSOL	5%	60%	70%

To test the global convergence of the IFDQ algorithm, we experimented with some of the problems by randomly varying the starting point. In all cases, we ran the algorithm with 500 starting points, whose components were uniformly distributed in the interval $[-100,\ 100]$. In Table 5 we show, for each experiment, the problem, the size of the problem and the success rate of the method. The success rate of the algorithm on selected problems shows us a robust method; thus, it would be a good option for solving large-scale nonlinear system of equations.

To finish our experiments, we want to show the inner behavior of the IFDQ algorithm when solving the problem given by the Extended Rosenbrock function.

In Table 6, we show the behavior of the most important parameters in the algorithm when solving the above-mentioned problem. In this table,

$$RelRes = \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|}$$

it helps us to analyze the rate convergence of the algorithm. As we can see in Table 6, $RelRes$ converge to zero when θ_k converge to zero, which suppose a superlinear convergence of the algorithm, as we proved in Theorem 5. On the other hand, $\alpha_k = 1$ for $k > 18$ just as we proved in Theorem 3.

5 Final remarks

In this work we proposed a new free derivative method for solving, especially, large-scale nonlinear systems of equations. This method takes inexact quasi-Newton

Table 5 Global convergence

F	n	Success rate
6	1000	100%
7	399	98%
13	1000	97%
15	500	100%
16	50	94%
17	2000	98.4%
19	1000	97.8%

Table 6 Inner behavior

k	α_k	θ_k	$RelRes$
1	0.2500	0.3334	0.5590
2	0.2500	0.2500	1.3503
3	0.0313	0.2000	0.9887
4	0.1250	0.1667	0.9713
5	0.0625	0.1429	0.9841
6	0.0625	0.1250	0.9807
7	0.0625	0.1111	0.9782
8	0.0313	0.1000	0.9878
9	0.0625	0.0909	0.9732
10	0.0625	0.0833	0.9717
11	0.0625	0.0769	0.9697
12	0.0625	0.0714	0.9675
13	0.0625	0.0667	0.9652
14	0.1250	0.0625	0.9263
15	0.1250	0.0588	0.9225
16	0.1250	0.0556	0.9158
17	0.2500	0.0526	0.8185
18	0.5000	0.0500	0.6142
19	1.0000	0.0476	0.1404
20	1.0000	0.0455	0.1022
21	1.0000	0.0435	0.0801
22	1.0000	0.0417	0.0748
23	1.0000	0.0400	0

directions to build the new iterate and, taking into account that so far it has not been possible to demonstrate that this is a descent direction and seeking to establish global convergence we proposed a two-stage linear search procedure.

For this new method, we show that it enjoys good convergence properties, this is, IFDQ method locally performs very well and, under reasonable hypotheses, has until superlinear convergence.

Numerical experiments showed that IFDQ had a performance according to expected and that this is a competitive method for solving large-scale nonlinear systems of equations.

Acknowledgements We are grateful with University of Cauca for the time allowed to this research through VRI ID 5352 project.

Author Contributions All authors contributed equally to this work.

Funding Open Access funding provided by Colombia Consortium. This work was supported by Universidad del Cauca given the time for the researcher through VRI ID 5352 project.

Code Availability This work introduce an algorithm completely available for the readers.

Declarations

Ethics approval The authors declare that they followed all the rules of a good scientific practice.

Consent to participate All authors approve their participation in this work.

Consent for publication The authors approve the publication of this research.

Competing interests The authors declare no competing interests.

References

1. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research, New York (2000)
2. Dennis, J.E., Schnabel. R.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM, Philadelphia (1996)
3. Pérez, R., Rocha, V.L.: Recent Applications and Numerical Implementation of Quasi-Newton Methods for Solving Nonlinear Systems of Equations. Numer Algorithms. **35**, 261–285 (2004). https://doi.org/10.1023/B:NUMA.0000021762.83420.40
4. Martínez, J.M.: Practical quasi-Newton methods for solving nonlinear systems. J Comput Math. **124**(1–2), 97–121 (2000). https://doi.org/10.1016/S0377-0427(00)00434-9
5. Gomes-Ruggiero, M.A., Martínez, J.M., Santos, S.A.: Solving nonsmooth equations by means of Quasi-Newton methods with globalization. Recent Advances in Noonsmooth Optimization, pp. 121 – 140. World Scientific Publisher, Singapore (1995). https://doi.org/10.1142/9789812812827_0008
6. Martínez, J.M.: A family of quasi-Newton methods for nonlinear equations with direct secant updates of matrix factorizations. SIAM J Numer Anal. **27**(4), 1034–1049 (1990). https://doi.org/10.1137/0727061
7. Dennis, J.E., More, J.J.: A Characterization of Superlinear Convergence and its Application to Quasi-Newton Methods. Math Comput. **28**(126), 549–560 (1974). https://doi.org/10.2307/2005926
8. Schubert, L.: Modification of quasi-Newton method for nonlinear equations with a sparse Jacobian. Math Comp. **24**(109), 27–30 (1970). https://doi.org/10.2307/2004874
9. Dembo, R., Eisenstat, S., Trond, S.: Inexact Newton Methods. SIAM J Numer Anal. **19**(2), 400–408 (1982). https://doi.org/10.1137/0719025
10. Deuflhard, P.: Global Inexact Newton Mehods for Very Large Scale Nonlinear Problems. Impact of Comptuning Science and engineering. **3**(4), 366–393 (1991). https://doi.org/10.1016/0899-8248(91)90004-E
11. Martinez, J.M., Qi, L.: Inexact Newton Methods for solving nonsmooth equations. J Comput Appl Math. **60**(1–2), 127–145 (1995). https://doi.org/10.1016/0377-0427(94)00088-I
12. Steighaug, T.: Quasi Newton Methods for Large Scale Nonlinear Problems. Yale University (1980)
13. Steihaug, T.: Damped Inexact Quasi Newton Methods. Technical Report TR81-03, Departament of Mathematical Science, Rice University (1984). https://hdl.handle.net/1911/101545
14. Eisenstat, S., Steihaug, T.: Local Analysis of Inexact Quasi Newton Methods. Technical Report TR82-07, Departament of Mathematical Science, Rice University (1982). https://hdl.handle.net/1911/101548
15. Eisenstat, S., Walker, H.: Globally Convergent Inexact Newton Methods. SIAM J Optim. **4**(2), 393–422 (1994). https://doi.org/10.1137/0804022

16. Birgin, E., Krejić, N., Martínez, J.M.: Globally convergent inexact quasi-Newton methods for solving nonlinear systems. Numer Algorithms. **32**, 249–260 (2003). https://doi.org/10.1023/A:1024013824524
17. La Cruz, W., Raydan, M.: Nonmonotone Spectral Methods for Large- Scale Nonlinear Systems. Optim Method Softw. **18**(5), 583–599 (2003). https://doi.org/10.1080/10556780310001610493
18. La Cruz, W., Martínez, J.M., Raydan, M.: Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. Math Comput. **75**(255), 1429–1448 (2006). https://doi.org/10.1090/S0025-5718-06-01840-0
19. Lukšan, L., Vlček, J.: Sparse and Partially Separable Test Problems for Unconstrained and Equality Constrained Optimization. Technical Report 767, Institute of Computer Science, Academy of Sciences of the Czech Republic (1999). http://hdl.handle.net/11104/0123965
20. Tsoulos, I., Athanassios, S.: On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods. Nonlinear Anal-Real. **11**(4), 2465–2471 (2010). https://doi.org/10.1016/j.nonrwa.2009.08.003
21. Zhou, W., Li, D.: Limited memory BFGS method for nonlinear monotone equations. J Comput Math. 25(1), 89 – 96 (2007). https://www.jstor.org/stable/43693347
22. Birgin, E., Martínez, J.M.: Secant acceleration of sequential residual methods for solving large-scale nonlinear systems of equations. SIAM J Numer Anal. **60**(6), 3145–3180 (2022). https://doi.org/10.1137/20M1388024
23. Pernice, M., Walker, H.F.: NITSOL: a newton iterative solver for nonlinear systems. SIAM J Sci Comput. **19**(1), 302–318 (1998). https://doi.org/10.1137/S1064827596303843
24. Broyden, C.G.: A Class of Methods for Solving Nonlinear Simultaneous Equations. Math Comput. **19**(92), 577–593 (1965). https://doi.org/10.2307/2003941
25. Broyden, C.G., Dennis, J.E., Moré, J.J.: On the Local and Superlinear Convergence of Quasi-Newton Methods. IMA J Appl Math. **12**(3), 223–245 (1973). https://doi.org/10.1093/imamat/12.3.223
26. Gasparo, M.: A nonmonotone hybrid method for nonlinear systems. Optim Method Softw. **13**(2), 79–94 (2000). https://doi.org/10.1080/10556780008805776
27. Gomez-Ruggiero, M., Martínez, J.M., Moretti, A.C.: Comparing algorithms for solving sparse nonlinear systems of equations. SIAM J Sci and Stat Comput. **13**(2), 459–483 (1992). https://doi.org/10.1137/0913025
28. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J opt. **7**(1), 2–33 (1997). https://doi.org/10.1137/S1052623494266365