

## Research Article

# A Cauchy Point Direction Trust Region Algorithm for Nonlinear Equations

Zhou Sheng<sup>1</sup> and Dan Luo<sup>2</sup> 

<sup>1</sup>Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China

<sup>2</sup>School of Mathematics and Statistics, Baise University, Baise 533000, Guangxi, China

Correspondence should be addressed to Dan Luo; [bsxyldan@163.com](mailto:bsxyldan@163.com)

Received 13 July 2020; Accepted 10 August 2020; Published 24 August 2020

Guest Editor: Maojun Zhang

Copyright © 2020 Zhou Sheng and Dan Luo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a Cauchy point direction trust region algorithm is presented to solve nonlinear equations. The search direction is an optimal convex combination of the trust region direction and the Cauchy point direction with the sufficiently descent property and the automatic trust region property. The global convergence of the proposed algorithm is proven under some conditions. The preliminary numerical results demonstrate that the proposed algorithm is promising and has better convergence behaviors than the other two existing algorithms for solving large-scale nonlinear equations.

## 1. Introduction

The system of nonlinear equations is one of the most important mathematical models, with extensive applications in chemical equilibrium problems [1], problem of hydraulic circuits of power plants [2], L-shaped beam structures [3], and image restoration problems [4]. Some  $\ell_1$ -norm regularized optimization problems in compressive sensing [5] are also obtained by reformulating the systems of nonlinear equations. In this paper, consider the following nonlinear system of equation:

$$F(x) = 0, \quad (1)$$

where  $F: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  is continuously differentiable. Let

$$f(x) = \frac{1}{2} \|F(x)\|^2, \quad (2)$$

where  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$ . It is clear that the nonlinear system of equations (1) is equivalent to the unconstrained optimization problem as follows:

$$\min_{x \in \mathfrak{R}^n} f(x). \quad (3)$$

Many literature studies have studied a variety of numerical algorithms to solve the nonlinear system of

equations, such as the conjugate gradient algorithm [6–9], Levenberg–Marquardt algorithm [10–13], and trust region algorithm [14–21]. The trust region method plays an important role in the area of nonlinear optimization. It was proposed by Powell in [22] to solve the nonlinear optimization problem by using an iterative structure. Recently, some trust region-based methods have been developed to solve nonlinear equations and have shown very promising results. For instance, Fan and Pan [16] discussed a regularized trust region subproblem with an adaptive trust region radius, which possesses quadratic convergence under the local error bound condition. Yuan et al. [17] proposed a trust region-based algorithm, in which the Jacobian is updated by the BFGS formula, for solving nonlinear equations. The algorithm showed global convergence at a quadratic convergence rate. This algorithm does not compute the Jacobian matrix in each interrogation, which significantly reduces the computation burden. Esmaeili and Kimiaei [19] studied an adaptive trust region algorithm with the moderate radius size by employing the nonmonotone technique, and the global and quadratic local convergence was given. However, the maximal dimension of the test problems was only 300. Qi et al. [23] proposed an asymptotic Newton search direction method which is an adequate combination of the projected

gradient direction and the projected trust region direction for solving box-constrained nonsmooth equations and proved the global and quadratic local convergence. Motivated by the search direction by Qi et al., Yuan et al. [24] used the idea of the search direction to solve nonsmooth unconstrained optimization problems and also showed global convergence.

Inspired by the fact that the search direction by Qi et al. [23] speeds up the local convergence rate compared to the gradient direction, we propose an efficient Cauchy point direction trust region algorithm to solve (1) in this paper. The search direction is an optimal convex combination of the trust region direction and the Cauchy point direction with the automatic trust region property and the sufficiently descent property, which can improve the numerical performance. We also show the global convergence of the proposed algorithm. Numerical results indicate that this algorithm has better behavior convergence compared with that of the classical trust region algorithm and the adaptive regularized trust region algorithm. However, the local convergence is not given.

The remainder of this paper is organized as follows. In Section 2, we introduce our motivation and a new search direction. In Section 3, we give a Cauchy point direction trust region algorithm for solving problem (1). The global convergence of the proposed algorithm is proven in Section 4. Preliminary numerical results are reported in Section 5. We conclude our paper in Section 6.

Throughout this paper, unless otherwise specified,  $\|\cdot\|$  denotes the Euclidean norm of vectors or matrices.

## 2. Motivation and Search Direction

In this section, motivated by the definition of the Cauchy point which is parallel to the steepest descent direction, we present a search direction with the Cauchy point. Let  $\Delta_k$  be the trust region radius and  $\Delta_k > 0$ .

**2.1. Cauchy Point Direction.** Noting that  $J_k^T J_k$  is a semi-definite matrix, we define the Cauchy point at the current point  $x_k$ :

$$d_k^C(\Delta_k) := -\frac{c_k \Delta_k}{\|J_k^T F_k\|} J_k^T F_k, \quad (4)$$

where

$$c_k = \begin{cases} 1, & \text{if } F_k^T (J_k J_k^T)^2 F_k = 0, \\ \min \left\{ 1, \frac{\|J_k^T F_k\|^3}{\Delta_k F_k^T (J_k J_k^T)^2 F_k} \right\}, & \text{if } F_k^T (J_k J_k^T)^2 F_k > 0, \end{cases} \quad (5)$$

$F_k = F(x_k)$ , and  $J_k = \nabla F(x_k)$  is the Jacobian of  $F(x)$  or its approximation.

The following lemma gives a nice property of the Cauchy point direction  $d_k^C$ .

**Proposition 1** (automatic trust region property). *Suppose that  $d_k^C$  is defined by (4), then  $\|d_k^C\| \leq \Delta_k$ .*

*Proof.* It is easy to get the result of this proposition from  $0 < c_k \leq 1$ . This completes the proof.

From the definition of  $d_k^C$  and the above proposition, we can determine the descent property and the automatic trust region property of the Cauchy point direction  $d_k^C$ .  $\square$

**2.2. Trust Region Direction.** Let the trial step  $d_k^{\text{TR}}(\Delta_k)$  be a solution of the following trust region subproblem:

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & m_k(d) := \frac{1}{2} \|F_k + J_k d\|^2, \\ \text{s.t.} \quad & \|d\| \leq \Delta_k, \end{aligned} \quad (6)$$

where  $\Delta_k$  is the same as the above definition.

From the famous result of Powell [22], we give the following lemma and omit its proof.

**Lemma 1.** *Let  $d_k^{\text{TR}}(\Delta_k)$  be a solution of the trust region subproblem (6), then*

$$\begin{aligned} f_k - m_k(d_k^{\text{TR}}(\Delta_k)) &= \frac{1}{2} \|F_k\|^2 - \frac{1}{2} \|F_k + J_k d_k^{\text{TR}}(\Delta_k)\|^2 \\ &\geq \frac{1}{2} \|J_k^T F_k\| \min \left\{ \Delta_k, \frac{\|J_k^T F_k\|}{\|J_k^T J_k\|} \right\}. \end{aligned} \quad (7)$$

As mentioned by Qi et al. [23] and Yuan et al. [24], when  $x_k$  is far from the optimization solution, the trust region direction  $d_k^{\text{TR}}(\Delta_k)$  may not possess the descent property. However, it follows from the definition of the Cauchy point direction that  $d_k^C(\Delta_k)$  always possesses the descent property. On the contrary, if the search direction is defined by using a combination of  $d_k^C$  and  $d_k^{\text{TR}}$ , it will be a descent direction. Hence, we give a new search direction by using a convex combination form in the next section.

**2.3. Search Direction.** Let  $\lambda_k$  be a solution of the one-dimensional minimization problem as follows:

$$\min_{\lambda \in [0,1]} h_k(\lambda) := \frac{1}{2} \|F_k + J_k(\lambda d_k^C(\Delta_k) + (1-\lambda)d_k^{\text{TR}}(\Delta_k))\|^2, \quad (8)$$

and the search direction is defined as

$$d_k(\Delta_k) = \lambda_k d_k^C(\Delta_k) + (1-\lambda_k) d_k^{\text{TR}}(\Delta_k). \quad (9)$$

For the above one-dimensional minimization problem, we can use the gold section method, Fibonacci method, parabolic method [25], etc. The idea of the search direction follows from Qi et al. [23]. However, we use the Cauchy point direction which possesses the automatic trust region property to get optimal  $\lambda_k$  by using the gold section method for problem (6) in this paper.

### 3. New Algorithm

In this section, let  $d_k(\Delta_k)$  be the optimal solution of (3). Then, the actual reduction is defined as

$$\text{Ared}_k := f_k - f(x_k + d_k(\Delta_k)), \quad (10)$$

and the predicted reduction is

$$\begin{aligned} \text{Pred}_k &:= m_k(0) - m_k(d_k(\Delta_k)) \\ &= \frac{1}{2} \|F_k\|^2 - \frac{1}{2} \|F_k + J_k d_k(\Delta_k)\|^2. \end{aligned} \quad (11)$$

Thus,  $r_k$  is defined by the ratio between  $\text{Ared}_k$  and  $\text{Pred}_k$ :

$$r_k := \frac{\text{Ared}_k}{\text{Pred}_k}. \quad (12)$$

We now list the detailed steps of the Cauchy point direction trust region algorithm in Algorithm 1.

**Algorithm 1.** Cauchy point direction trust region algorithm for solving (3).

*Step 1.* Given initial point  $x_0 \in \mathfrak{R}^n$  and parameters  $0 \leq \mu_1 < \mu_2 < 1$ ,  $0 < \eta_1 < 1 < \eta_2$ ,  $0 < \delta < 1$ ,  $0 \leq \epsilon \ll 1$ , and  $\Delta_0 > 0$ . Let  $k := 0$ .

*Step 2.* If the termination condition  $\|J_k^T F_k\| \leq \epsilon$  is satisfied at the iteration point  $x_k$ , then stop. Otherwise, go to Step 3.

*Step 3.* Solve the trust region subproblem (6) to obtain  $d_k^{\text{TR}}(\Delta_k)$  and compute  $c_k$  and  $d_k^C(\Delta_k)$  from (4) and (5), respectively.

*Step 4.* To solve the one-dimensional minimization problem (8) giving  $\lambda_k$  by using Algorithm 2, calculate the search direction from (6).

*Step 5.* Compute  $\text{Ared}_k$ ,  $\text{Pred}_k$ , and  $r_k$ , if

$$\begin{cases} f_k - \frac{1}{2} \|F_k + J_k d_k\|^2 \geq -\delta F_k J_k^T d_k^C(\Delta_k), \\ r_k \geq \mu_1, \end{cases} \quad (13)$$

are satisfied; let  $x_{k+1} := x_k + d_k(\Delta_k)$ , and update  $\Delta_k$  by

$$\Delta_{k+1} := \begin{cases} \eta_1 \Delta_k, & \text{if } r_k < \mu_1, \\ \Delta_k, & \text{if } \mu_1 \leq r_k < \mu_2, \\ \eta_2 \Delta_k, & \text{if } r_k \geq \mu_2. \end{cases} \quad (14)$$

*Step 6.* If  $r_k < \mu_1$ , return to Step 3, and let  $x_{k+1} := x_k$ . Otherwise, return to Step 2. Let  $k := k + 1$ .

In Step 4 of Algorithm 1, we use the gold section algorithm as given in Algorithm 2.

**Algorithm 2.** Gold section algorithm [25] for solving problem (8).

*Step 1.* For the given initial parameters  $a_0 = 0, b_0 = 1, \epsilon' > 0$ , compute

$$\begin{aligned} p_0 &:= a_0 + 0.382(b_0 - a_0), \\ q_0 &:= a_0 + 0.618(b_0 - a_0), \end{aligned} \quad (15)$$

and  $h_k(p_0), h_k(q_0)$ . Let  $k := 0$ .

*Step 2.* If  $h_k(p_k) \leq h_k(q_k)$ , go to Step 3. Otherwise, go to Step 4.

*Step 3.* If the termination condition  $|q_k - a_k| \leq \epsilon'$  is satisfied, stop. Otherwise, let

$$\begin{aligned} a_{k+1} &:= a_k, & b_{k+1} &:= q_k, & h_k(q_{k+1}) &:= h_k(p_k), \\ q_{k+1} &:= p_k, & p_{k+1} &:= a_{k+1} + 0.382(b_{k+1} - a_{k+1}), \end{aligned} \quad (16)$$

and compute  $h_k(p_{k+1})$  and go to Step 5.

*Step 4.* If the termination condition  $|b_k - p_k| \leq \epsilon'$  is satisfied, stop. Otherwise, let

$$\begin{aligned} a_{k+1} &:= p_k, & b_{k+1} &:= b_k, & h_k(p_{k+1}) &:= h_k(q_k), \\ p_{k+1} &:= q_k, & q_{k+1} &:= a_{k+1} + 0.618(b_{k+1} - a_{k+1}), \end{aligned} \quad (17)$$

and compute  $h_k(q_{k+1})$  and go to Step 5.

*Step 5.* Let  $k := k + 1$ ; return to Step 2.

### 4. Global Convergence

In this section, the global convergence of Algorithm 1 is proven. Firstly, it is easy to have the following lemma from Lemma 1.

**Lemma 2.** Let  $d_k(\Delta_k)$  be defined by (9), where  $\lambda_k$  is a solution of problem (8). Then,

$$\text{Pred}_k \geq \frac{1}{2} \|J_k^T F_k\| \min \left\{ \Delta_k, \frac{\|J_k^T F_k\|}{\|J_k^T J_k\|} \right\}. \quad (18)$$

*Proof.* From the definition of  $d_k(\Delta_k)$ , we have  $m_k(d_k(\Delta_k)) \leq m_k(d_k^{\text{TR}}(\Delta_k))$ . It follows from the definition of  $\text{Pred}_k$  and Lemma 1 that  $\text{Pred}_k \geq f_k - m_k(d_k^{\text{TR}}(\Delta_k))$  holds. Thus, (16) holds, and therefore, the proof is completed.

The next lemma indicates an important property of the Cauchy point direction  $d_k^C$ , which is the sufficiently descent property.  $\square$

**Lemma 3.**  $(J_k^T F_k)^T d_k^C(\Delta_k) = -c_k \Delta_k \|J_k^T F_k\|$  holds, for any  $\Delta_k > 0$ .

*Proof.* For any  $\Delta_k$ , from (9), we have

$$\begin{aligned} (J_k^T F_k)^T d_k^C(\Delta_k) &= F_k^T J_k d_k^C(\Delta_k) = -c_k \Delta_k F_k^T J_k \frac{J_k^T F_k}{\|J_k^T F_k\|} \\ &= -c_k \Delta_k \|J_k^T F_k\|, \end{aligned} \quad (19)$$

and note that  $0 < c_k \leq 1$ ; thus,

$$(J_k^T F_k)^T d_k^C(\Delta_k) = -c_k \Delta_k \|J_k^T F_k\| < 0. \quad (20)$$

This completes the proof.

To derive the global convergence of Algorithm 1, the following assumption is considered.  $\square$

*Assumption 1.* The Jacobian  $J(x)$  of  $F(x)$  is bounded; i.e., there exists a positive constant  $M > 0$  such that  $\|J(x)\| \leq M$ , for any  $x \in \mathfrak{R}^n$ .

In the sequel, we show that “Steps 3–6” of Algorithm 1 do not cycle in the inner loop infinitely.

**Lemma 4.** *Suppose that Assumption 1 holds, let  $\{x_k\}$  be generated by Algorithm 1. If  $x_k$  is not a stationary point of problem (3), then Algorithm 1 is well defined; i.e., the inner*

*cycle of Algorithm 1 can be left in a finite number of internal iterates.*

*Proof.* It follows from  $x_k$  which is not a stationary point that there exists a positive constant  $\epsilon_0 > 0$  such that

$$\|J_k^T F_k\| \geq \epsilon_0, \quad \forall k. \quad (21)$$

Since  $0 < c_k \leq 1$  and Assumption 1 holds, there exists a positive constant  $\alpha > 0$  such that

$$\|J_k d_k^C\| \leq c_k \Delta_k \|J_k\| \leq \alpha \Delta_k. \quad (22)$$

Letting  $0 < \Delta_k \leq 2/\alpha^2 (1 - \delta)\epsilon_0 c_k$ , then from (4) and (22), we obtain

$$\begin{aligned} f_k - \frac{1}{2} \|F_k + J_k d_k(\Delta_k)\|^2 &\geq f_k - \frac{1}{2} \|F_k + J_k d_k^C(\Delta_k)\|^2 = -F_k^T J_k d_k^C(\Delta_k) - \frac{1}{2} \|J_k d_k^C(\Delta_k)\|^2 \\ &= -\delta F_k^T J_k d_k^C(\Delta_k) - (1 - \delta) F_k^T J_k d_k^C(\Delta_k) - \frac{1}{2} \|J_k d_k^C(\Delta_k)\|^2 \\ &\geq -\delta F_k^T J_k d_k^C(\Delta_k) + (1 - \delta) c_k \Delta_k \|J_k^T F_k\| - \frac{1}{2} \alpha^2 \Delta_k^2, \end{aligned} \quad (23)$$

and noting (21), we have  $(1 - \delta) c_k \Delta_k \|J_k^T F_k\| - 1/2 \alpha^2 \Delta_k^2 \geq (1 - \delta) c_k \Delta_k - 1/2 \alpha^2 \Delta_k^2 \geq 0$ , which implies

$$f_k - \frac{1}{2} \|F_k + J_k d_k(\Delta_k)\|^2 \geq -\delta F_k^T J_k d_k^C(\Delta_k), \quad (24)$$

and therefore, (13) holds. Next, we prove that (14) is also satisfied. By contradiction, we assume that “Step 3–Step 6” of Algorithm 1 do cycle in the inner loop infinitely; then,  $\Delta_k \rightarrow 0$ .

From the definition of  $d_k^C(\Delta_k)$ , we get

$$\begin{aligned} \frac{1}{2} \|F_k + J_k d_k^C(\Delta_k)\|^2 &= \frac{1}{2} \|F_k\|^2 + F_k^T J_k d_k^C(\Delta_k) + \frac{1}{2} \|J_k d_k^C(\Delta_k)\|^2 \\ &= \frac{1}{2} \|F_k\|^2 - c_k \Delta_k \|J_k^T F_k\| + \frac{1}{2} \alpha^2 \Delta_k^2. \end{aligned} \quad (25)$$

It follows from (13) and Lemma 3 that there exists a positive constant  $\beta > 0$  such that

$$\begin{aligned} f_k - \frac{1}{2} \|F_k + J_k d_k(\Delta_k)\|^2 &\geq -\delta F_k^T J_k d_k^C(\Delta_k) \\ &= \delta c_k \Delta_k \|J_k^T F_k\| \geq \beta \Delta_k. \end{aligned} \quad (26)$$

By the definition  $d_k^{\text{TR}}(\Delta_k)$ , we have  $\|d_k^{\text{TR}}\| \leq \Delta_k$ , and noting Proposition 1 and  $0 < c_k \leq 1$ , further, we get

$$\|d_k(\Delta_k)\| \leq \Delta_k. \quad (27)$$

Thus, from (26) and (27), we have

$$\begin{aligned} |r_k - 1| &= \left| \frac{\text{Ared}_k - \text{Pred}_k}{\text{Pred}_k} \right| \\ &= \left| \frac{1/2 \|F_k + J_k d_k(\Delta_k)\|^2 - 1/2 \|F_k + J_k d_k(\Delta_k)\| + O\left(\|d_k(\Delta_k)\|^2\right)}{\text{Pred}_k} \right| \\ &= \frac{\|F_k + J_k d_k(\Delta_k)\| O\left(\|d_k(\Delta_k)\|^2\right) + O\left(\|d_k(\Delta_k)\|^4\right)}{f_k - 1/2 \|F_k + J_k d_k\|^2} \leq \frac{O\left(\|d_k(\Delta_k)\|^2\right)}{\beta \Delta_k} \leq \frac{O\left(\|\Delta_k\|^2\right)}{\beta \Delta_k} \rightarrow 0, \end{aligned} \quad (28)$$

TABLE 1: Test problems.

No.	Problem name	$x_0$
1	Exponential function 1	$(n/(n-1), \dots, n/(n-1))$
2	Exponential function 2	$(1/n^2, \dots, 1/n^2)$
3	Extended Rosenbrock function	$(5, 1, \dots, 5, 1)$
4	Trigonometric function	$(101/100n, \dots, 101/100n)$
5	Singular function	$(1, \dots, 1)$
6	Logarithmic function	$(1, \dots, 1)$
7	Broyden tridiagonal function	$(-1, \dots, -1)$
8	Trigexp function	$(0, \dots, 0)$
9	Function 15	$(-1, \dots, -1)$
10	Strictly convex function 1	$(1/n, \dots, 1/n)$
11	Strictly convex function 2	$(1, \dots, 1)$
12	Zero Jacobian function	$(100(n-100)/n, (n-1000)(n-500)/360n^2, \dots, (n-1000)(n-500)/360n^2)$
13	Linear function-full rank	$(100, \dots, 100)$
14	Penalty function	$(1/3, \dots, 1/3)$
15	Brown almost-linear function	$(1/2, \dots, 1/2)$
16	Extended Powell singular function	$(1.5 \times 10^{-4}, \dots, 1.5 \times 10^{-4})$
17	Tridiagonal system	$(12, \dots, 12)$
18	Extended Freudenstein and Roth function	$(6, 3, \dots, 6, 3)$
19	Discrete boundary value problem	$-n/(n+1)^2, \dots, i-n-1/(n+1)^2, \dots, -1/(n+1)^2$
20	Troesch problem	$(1/2, \dots, 1/2)$

and therefore  $r_k \rightarrow 1$ , which means that  $r_k \geq \mu_2$  when  $\Delta_k$  is sufficiently small. However, the definition of  $\Delta_k$  in Step 5 of Algorithm 1 contradicts the fact that  $\Delta_k \rightarrow 0$ . Hence, the cycling between Step 3 and Step 6 terminates finitely. This completes the proof.

We can get the global convergence of Algorithm 1 based on the above lemmas.  $\square$

**Theorem 1.** *Suppose that Assumption 1 holds and  $\{x_k\}$  be generated by Algorithm 1, then*

$$\liminf_{k \rightarrow \infty} \|J_k^T F_k\| = 0. \quad (29)$$

*Proof.* Suppose that the conclusion is not true, then there exists a positive constant  $\epsilon_1 > 0$  such that for any  $k$ ,

$$\|J_k^T F_k\| \geq \epsilon_1. \quad (30)$$

From Lemma 2, Assumption 1, and the above inequation, we have

$$\text{Pred}_k \geq \frac{1}{2} \|J_k^T F_k\| \min \left\{ \Delta_k, \frac{\|J_k^T F_k\|}{\|J_k^T J_k\|} \right\} \geq \frac{1}{2} \epsilon_1 \min \left\{ \Delta_k, \frac{\epsilon_1}{M^2} \right\}. \quad (31)$$

Noting the updating rule of  $\Delta_k$ ,  $r_k \geq \mu_1$ , (31), and Lemma 4, we deduce that

$$\begin{aligned} +\infty &\geq \sum_{k=1}^{\infty} \left( \frac{1}{2} \|F_k\|^2 - \frac{1}{2} \|F_{k+1}\|^2 \right), \\ &\geq \sum_{k=1}^{\infty} \mu_1 \text{Pred}_k, \\ &\geq \sum_{k=1}^{\infty} \frac{1}{2} \mu_1 \epsilon_1 \min \left\{ \Delta_k, \frac{\epsilon_1}{M^2} \right\}, \\ &\geq \sum_{k=1}^{\infty} \frac{1}{2} \mu_1 \epsilon_1 \min \left\{ \Delta, \frac{\epsilon_1}{M^2} \right\}, \\ &= +\infty, \end{aligned} \quad (32)$$

which yields a contradiction that this means (30) is not true; i.e., the result (29) of the theorem holds, and therefore, the proof is completed.  $\square$

## 5. Numerical Results

In this section, the numerical results of the proposed algorithm for solving problem (1) are reported. We denote the proposed Algorithm 1 by CTR and compare it with some existing algorithms. The adaptive trust region algorithm by Fan and Pan [16] and the classical trust region algorithm [26] are denoted by ATR and TTR, respectively. In numerical experiments, all parameters of Algorithm 1 are chosen as follows:  $\mu_1 = 0.1$ ,  $\mu_2 = 0.9$ ,  $\eta_1 = 0.25$ ,  $\eta_2 = 3$ ,  $\epsilon = 10^{-5}$ , and  $c_0 = \Delta_0 = 1$ ,  $\delta = 0.9$ . We choose  $\epsilon' = 10^{-6}$  in Algorithm 2. Steihaug method [27] is employed to obtain the approximative trial step  $d_k$  for solving the trust region subproblem (6). We will terminate algorithms if the number of iterations is larger than 2000.

All codes were implemented in MATLAB R2010b. The numerical experiments were performed on a PC with an Intel Pentium (R) Dual-Core CPU at 3.20 GHz and 2.00 GB of RAM and using the Windows 7 operating system.

We list the test problems in Table 1 that can also be found in [28]. In Tables 1–3, “No.” denotes the number of the test problem, “Problem name” denotes the name of the test problem, “ $x_0$ ” denotes the initial point, “ $n$ ” denotes the dimension of the problem, “NI” is the total number of iterations, “NF” is the number of the function evaluations, “ $\|F_k\|$ ” refers to the optimum function values, and “ $\lambda_{\text{mean}}$ ” denotes the mean value of  $\lambda$ . It is worth mentioning that “\*” represents the stopping of the algorithms in situations that the number of iterations exceeds the maximal iterations, but the terminal conditions cannot be satisfied yet.

It is shown from Tables 2 and 3 that the proposed algorithm is efficient to solve these test problems and is competitive with the other two algorithms in NI and NF for most problems. For problems 1, 2, 16, 19, and 20,  $\lambda_{\text{mean}}$  is

TABLE 2: Numerical results.

No.	$n$	CTR			ATR		TTR	
		NI/NF	$\ F_k\ $	$\lambda_{\text{mean}}$	NI/NF	$\ F_k\ $	NI/NF	$\ F_k\ $
1	500	6/7	$3.18E-06$	$3.32E-07$	7/8	$6.73E-06$	6/7	$3.18E-06$
	1000	5/6	$8.95E-06$	$3.32E-07$	7/8	$4.74E-06$	5/6	$8.95E-06$
	2000	5/6	$6.32E-06$	$3.32E-07$	7/8	$3.34E-06$	5/6	$6.32E-06$
2	500	2/3	$3.50E-06$	$1.57E-06$	2/3	$3.38E-06$	2/3	$3.38E-06$
	1000	2/3	$6.67E-06$	$3.32E-07$	2/3	$6.69E-06$	2/3	$6.69E-06$
	2000	3/4	$9.13E-06$	$3.32E-07$	3/8	$8.39E-06$	3/5	$8.44E-06$
3	500	12/16	$2.16E-07$	$3.31E-01$	12/20	$3.89E-06$	16/21	$9.44E-10$
	1000	11/15	$5.92E-07$	$2.96E-01$	12/20	$5.51E-06$	16/21	$1.39E-09$
	2000	11/15	$1.78E-07$	$3.50E-01$	12/20	$7.82E-06$	16/21	$1.84E-09$
4	100	9/11	$7.33E-06$	$5.28E-01$	2001/2293	$3.03E-03^*$	2001/2270	$3.48E-03^*$
	500	22/25	$4.39E-06$	$2.96E-01$	18/19	$5.84E-06$	19/20	$8.32E-06$
5	1000	25/27	$6.37E-06$	$4.35E-01$	23/58	$5.68E-06$	23/24	$4.33E-06$
	2000	24/28	$5.24E-06$	$4.50E-01$	19/20	$5.53E-06$	21/23	$5.23E-06$
	500	5/6	$2.12E-06$	$9.81E-04$	5/6	$4.91E-08$	7/8	$2.74E-08$
6	1000	5/6	$1.44E-06$	$5.05E-01$	5/6	$1.91E-07$	7/8	$1.58E-07$
	2000	6/7	$7.69E-07$	$6.08E-01$	5/6	$7.56E-08$	7/8	$5.08E-08$
	500	6/7	$2.57E-09$	$1.59E-01$	6/7	$1.51E-09$	6/7	$4.28E-11$
7	1000	6/7	$2.63E-06$	$1.63E-01$	6/7	$1.03E-08$	6/7	$3.83E-06$
	2000	6/7	$5.50E-06$	$2.26E-01$	6/7	$2.60E-08$	6/7	$5.43E-06$
8	100	14/16	$4.31E-08$	$2.92E-01$	2001/2281	$3.76E+00^*$	2001/2268	$3.76E+00^*$
9	1000	14/18	$5.42E-06$	$1.46E-02$	14/20	$2.76E-07$	16/22	$1.36E-09$
	500	6/7	$5.54E-08$	$4.14E-01$	6/7	$1.33E-07$	6/7	$1.09E-07$
10	1000	6/7	$6.04E-08$	$3.71E-01$	6/7	$2.49E-07$	6/7	$9.91E-07$
	2000	6/7	$1.16E-07$	$4.47E-01$	6/7	$4.33E-07$	7/8	$2.32E-07$
11	100	8/9	$3.22E-07$	$2.21E-01$	8/9	$3.38E-08$	8/9	$1.15E-06$
	500	5/6	$3.62E-12$	$8.36E-01$	16/17	$5.81E-06$	18/19	$5.96E-06$
12	1000	5/6	$4.15E-11$	$8.82E-01$	16/17	$7.38E-06$	18/19	$9.31E-06$
	2000	6/8	$3.42E-07$	$8.29E-01$	16/17	$8.23E-06$	19/20	$2.82E-06$
	500	9/10	$2.41E-07$	$7.19E-01$	4/5	$7.67E-06$	9/10	$1.04E-08$
13	1000	9/10	$3.93E-07$	$7.62E-01$	4/5	$8.32E-06$	9/10	$3.37E-08$
	2000	10/11	$7.01E-06$	$7.30E-01$	5/6	$1.00E-10$	10/11	$1.35E-07$
14	500	4/5	$6.96E-06$	$1.09E-02$	7/10	$5.22E-06$	5/6	$5.34E-06$
	1000	12/13	$1.79E-07$	$1.77E-01$	8/10	$2.39E-06$	6/8	$2.65E-06$
15	2000	10/11	$5.39E-06$	$1.15E-01$	11/16	$1.70E-06$	10/11	$5.32E-07$
	500	3/4	$2.85E-09$	$3.61E-01$	3/4	$2.48E-06$	4/5	$2.49E-06$
	1000	4/5	$1.32E-07$	$5.56E-01$	3/4	$6.34E-07$	6/7	$8.80E-08$
16	2000	4/5	$1.40E-08$	$5.04E-01$	4/5	$3.74E-07$	6/7	$3.63E-07$
	500	1/2	$1.10E-06$	$3.32E-07$	1/2	$1.10E-06$	1/2	$1.10E-06$
17	1000	1/2	$1.55E-06$	$3.32E-07$	1/2	$1.55E-06$	1/2	$1.55E-06$
	2000	1/2	$2.20E-06$	$3.32E-07$	1/2	$2.20E-06$	1/2	$2.20E-06$
17	1000	34/44	$9.22E-09$	$3.58E-01$	90/219	$6.85E-09$	63/75	$4.89E-10$

TABLE 3: Numerical results (continued).

No.	$n$	CTR			ATR		TTR	
		NI/NF	$\ F_k\ $	$\lambda_{\text{mean}}$	NI/NF	$\ F_k\ $	NI/NF	$\ F_k\ $
18	500	6/8	$3.57E-07$	$4.13E-01$	7/8	$5.84E-08$	7/8	$6.12E-09$
	1000	8/10	$1.14E-06$	$3.17E-01$	7/8	$7.87E-08$	8/9	$7.10E-08$
	2000	8/10	$3.08E-06$	$3.17E-01$	7/8	$1.07E-07$	8/9	$2.29E-08$
19	500	2/3	$9.77E-09$	$3.32E-07$	3/4	$6.77E-09$	2/3	$9.77E-09$
	1000	2/3	$9.10E-10$	$3.32E-07$	3/4	$6.17E-10$	2/3	$9.09E-10$
20	2000	2/3	$4.47E-10$	$3.32E-07$	3/4	$3.02E-10$	2/3	$4.47E-10$
	1000	4/5	$1.90E-07$	$3.32E-07$	8/9	$8.94E-08$	4/5	$1.87E-07$

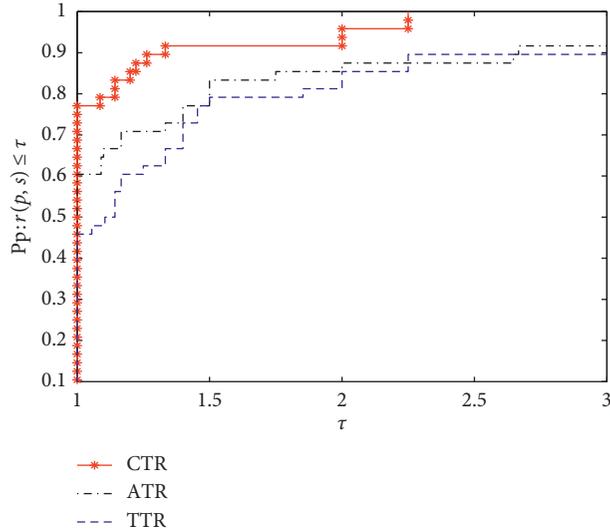


FIGURE 1: Performance profiles of the algorithms for the test problems (NI).

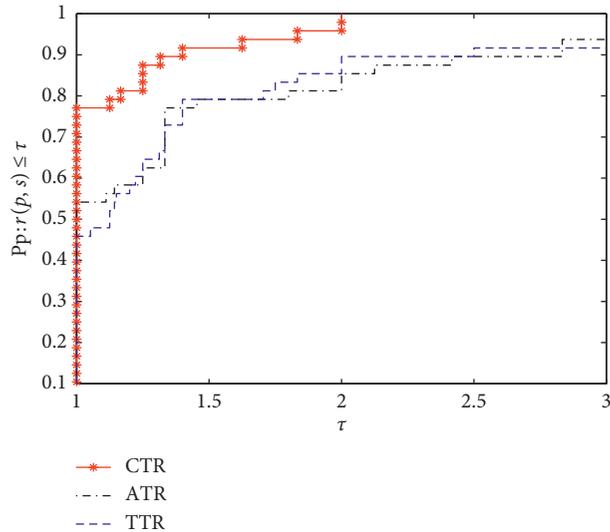


FIGURE 2: Performance profiles of the algorithms for the test problems (NF).

close to zero, which implies that the trust region direction was used many times. Moreover, the number of iteration is the same as that of algorithm TTR, which means that our algorithm speeds up the local convergence of the iterates. It can be seen that  $\lambda_{\text{mean}}$  is not too large for other test problems. However, for problems 4 and 8, algorithms ATR and TTR are not efficient to get the optimal solution when  $\text{NI} > 2000$ . We used the performance profiles by Dolan and Moré [29] to compare these algorithms with respect to NI and NF

again, as shown in Figures 1 and 2. Therefore, it is noted from Tables 2 and 3 and Figures 1 and 2 that the proposed algorithm is well promising and has better performance in test problems in comparison with two existing algorithms. In order to investigate the behavior of the convergence rate of three algorithms, we plot the curve  $0 - k - Lg\|x_k - x^*\|$  on problem 10 with  $n = 1000$  and  $n = 2000$  in Figure 3, where  $x^*$  is the optimal point. Figure 3 indicates that Algorithm 1 converges faster than others.

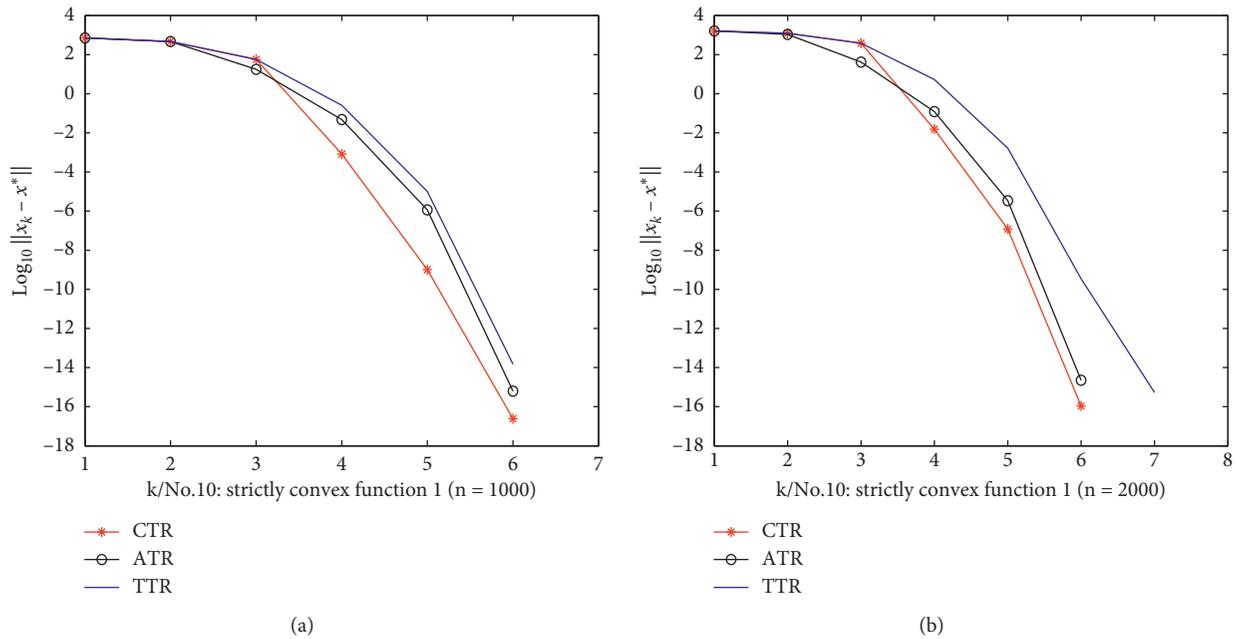


FIGURE 3: The behavior of the convergence: (a)  $n = 1000$  and (b)  $n = 2000$ .

## 6. Conclusions

In this paper, we presented a Cauchy point direction trust region algorithm for solving the nonlinear system of equations. This algorithm combines the trust region direction and the Cauchy point direction with the descent property and the automatic trust region. The optimal convex combination parameters were determined by using the gold section algorithm which is an exact one-dimensional linear search algorithm. The new algorithm was proven to have the descent property and the trust region property. We also showed the global convergence of the proposed algorithm under suitable condition. More importantly, the numerical results demonstrated the faster convergence of the proposed algorithm over two existing methods, which is very promising to solve large-scale nonlinear equations. However, the convergence rate of the proposed algorithm remains unclear. Therefore, the demonstration of the theoretical convergence rate of the proposed algorithm, application in image restoration problems, and compressive sensing will be the topics among others in our future study.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (11261006 and 11661009) and the Guangxi Natural Science Foundation (2020GXNSFAA159069).

## References

- [1] K. Meintjes and A. P. Morgan, "Chemical equilibrium systems as numerical test problems," *ACM Transactions on Mathematical Software*, vol. 16, no. 2, pp. 143–151, 1990.
- [2] A. A. Levin, V. Chistyakov, E. A. Tairov, and V. F. Chistyakov, "On application of the structure of the nonlinear equations system, describing hydraulic circuits of power plants, in computations," *Bulletin of the South Ural State University. Series "Mathematical Modelling, Programming and Computer Software"*, vol. 9, no. 4, pp. 53–62, 2016.
- [3] F. Georgiades, "Nonlinear equations of motion of L-shaped beam structures," *European Journal of Mechanics—A/Solids*, vol. 65, pp. 91–122, 2017.
- [4] G. Yuan, T. Li, and W. Hu, "A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems," *Applied Numerical Mathematics*, vol. 147, pp. 129–141, 2020.
- [5] Y. Xiao and H. Zhu, "A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing," *Journal of Mathematical Analysis and Applications*, vol. 405, no. 1, pp. 310–319, 2013.
- [6] Z. Papp and S. Rapajić, "FR type methods for systems of large-scale nonlinear monotone equations," *Applied Mathematics and Computation*, vol. 269, pp. 816–823, 2015.
- [7] W. Zhou and F. Wang, "A PRP-based residual method for large-scale monotone nonlinear equations," *Applied Mathematics and Computation*, vol. 261, pp. 1–7, 2015.

- [8] G. Yuan and M. Zhang, "A three-terms Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 286, pp. 186–195, 2015.
- [9] G. Yuan, Z. Meng, and Y. Li, "A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations," *Journal of Optimization Theory and Applications*, vol. 168, no. 1, pp. 129–152, 2016.
- [10] J. Fan and J. Pan, "Convergence properties of a self-adaptive Levenberg-Marquardt algorithm under local error bound condition," *Computational Optimization and Applications*, vol. 34, no. 1, pp. 47–62, 2006.
- [11] J. Fan, "Accelerating the modified Levenberg-Marquardt method for nonlinear equations," *Mathematics of Computation*, vol. 83, no. 287, pp. 1173–1187, 2013.
- [12] X. Yang, "A higher-order Levenberg-Marquardt method for nonlinear equations," *Applied Mathematics and Computation*, vol. 219, no. 22, pp. 10682–10694, 2013.
- [13] K. Amini and F. Rostami, "A modified two steps Levenberg-Marquardt method for nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 288, pp. 341–350, 2015.
- [14] J.-l. Zhang and Y. Wang, "A new trust region method for nonlinear equations," *Mathematical Methods of Operations Research (ZOR)*, vol. 58, no. 2, pp. 283–298, 2003.
- [15] J. Fan, "Convergence rate of the trust region method for nonlinear equations under local error bound condition," *Computational Optimization and Applications*, vol. 34, no. 2, pp. 215–227, 2006.
- [16] J. Fan and J. Pan, "An improved trust region algorithm for nonlinear equations," *Computational Optimization and Applications*, vol. 48, no. 1, pp. 59–70, 2011.
- [17] G. Yuan, Z. Wei, and X. Lu, "A BFGS trust-region method for nonlinear equations," *Computing*, vol. 92, no. 4, pp. 317–333, 2011.
- [18] H. Esmaeili and M. Kimiaei, "A new adaptive trust-region method for system of nonlinear equations," *Applied Mathematical Modelling*, vol. 38, no. 11–12, pp. 3003–3015, 2014.
- [19] H. Esmaeili and M. Kimiaei, "An efficient adaptive trust-region method for systems of nonlinear equations," *International Journal of Computer Mathematics*, vol. 92, no. 1, pp. 151–166, 2015.
- [20] M. Kimiaei and H. Esmaeili, "A trust-region approach with novel filter adaptive radius for system of nonlinear equations," *Numerical Algorithms*, vol. 73, no. 4, pp. 999–1016, 2016.
- [21] H. Esmaeili and M. Kimiaei, "A trust-region method with improved adaptive radius for systems of nonlinear equations," *Mathematical Methods of Operations Research*, vol. 83, no. 1, pp. 109–125, 2016.
- [22] M. J. D. Powell, "Convergence properties of a class of minimization algorithms," in *Nonlinear Programming*, pp. 1–27, Elsevier, Amsterdam, Netherlands, 1975.
- [23] L. Qi, X. J. Tong, and D. H. Li, "Active-set projected trust-region algorithm for box-constrained nonsmooth equations," *Journal of Optimization Theory and Applications*, vol. 120, no. 3, pp. 601–625, 2004.
- [24] G. Yuan, Z. Wei, and Z. Wang, "Gradient trust region algorithm with limited memory BFGS update for nonsmooth convex minimization," *Computational Optimization and Applications*, vol. 54, no. 1, pp. 45–64, 2013.
- [25] Y. X. Yuan and W. Sun, *Optimization Theory and Methods*, Science Press, Beijing, China, 1997.
- [26] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-region Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
- [27] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 626–637, 1983.
- [28] W. La Cruz, J. M. Martínez, and M. Raydan, "Spectral residual method without gradient information for solving large-scale nonlinear systems of equations," *Mathematics of Computation*, vol. 75, no. 255, pp. 1429–1449, 2006.
- [29] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.