Zhu et al. *Journal of Inequalities and Applications* (2020) 2020:25
https://doi.org/10.1186/s13660-020-2300-7

Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

A simple approximated solution method for solving fractional trust region subproblems of nonlinearly equality constrained optimization

Honglan Zhu[1,2*], Qin Ni[1] and Xuebing Zhang[3]

*Correspondence:
zhuhonglan1981@163.com
[1]College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[2]Business School, Huaiyin Institute of Technology, Huaian, China
Full list of author information is available at the end of the article

Abstract

In this paper, a fractional model is used to solve nonlinearly constrained optimization problems. In order to solve the fractional trust region subproblems simply, we propose an approximated solution method by cyclically fixing the fractional coefficient part of the approximate function. The global convergence of the fractional trust region method is proved, and the numerical results show that the new algorithm is effective and stable.

Keywords: Nonlinearly equality constrained optimization; Fractional model; Trust region method; Approximated solution method; Global convergence

1 Introduction

In this paper, we consider the nonlinear equality constrained optimization problem

$$\min_{x\in R^n} f(x), \tag{1.1}$$

$$\text{s.t.} \quad C(x) = 0, \tag{1.2}$$

where $C(x) = (c_1(x), \dots, c_m(x))^T$ $(m \le n)$ and $f(x)$, $c_i(x)$ $(i = 1, \dots, m)$ are continuously differentiable. The Lagrangian function for problem (1.1)–(1.2) is defined as follows:

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i c_i(x), \tag{1.3}$$

where λ_i for $i = 1, \dots, m$ are Lagrange multipliers. Problem (1.1)–(1.2) has been studied by many researchers, including Han [1], Powell [2], Yuan and Sun [3], Powell and Yuan [4], etc.

There are many efficient methods to solve problem (1.1)–(1.2), and the trust region method is a very effective method (see [4–13]). In addition, the book of Conn, Gould, and Toint [14] is an excellent and comprehensive one on trust region methods. However, most of these methods use the quadratic model to approximate $f(x)$. The sequential quadratic programming method for (1.1)–(1.2) computes a search direction by minimizing a quadratic model of the Lagrangian subject to the linearized constraints. That is, at the kth iteration, the following subproblem

$$\min_{s\in R^n} \varrho_k(s) = g_k^T s + \frac{1}{2} s^T B_k s, \tag{1.4}$$

$$\text{s.t.} \quad A_k^T s + C_k = 0 \tag{1.5}$$

is solved to obtain a search direction s_k, where x_k is the current iterate point, $g_k = \nabla f(x_k)$, B_k is symmetric and an approximation to the Hessian $\nabla_{xx} L(x_k, \lambda_k)$ of the Lagrangian of problem (1.1)–(1.2), $A_k = [\nabla c_1(x_k), \dots, \nabla c_m(x_k)]$ and $C_k = C(x_k)$. The constraint gradients $\nabla c_i(x_k)$ are assumed to be linearly independent for all x_k. However, if the objective function possesses high nonlinear property and the iterative point is far away from the minimum, the quadratic model could not approximate the original problem very well, which may lead to iteration proceeding slowly.

In 1980, Davidon [15] first proposed the collinear scaling of variables and conic model method for unconstrained optimization. The conic model is

$$\tilde{\phi}_k(s) = f_k + \frac{g_k^T s}{1 - a_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - a_k^T s)^2}, \tag{1.6}$$

where $a_k \in R^n$ is a horizontal vector. Then, Sorensen [16] published detailed results on a class of conic model methods and proved that a particular member of this class has the Q-superlinear convergence. Many other scholars have also studied the trust region algorithm of the conic model (see [17, 18]).

The trial step of a trust region algorithm is usually obtained by solving a trust region subproblem. In our trust region subproblem, the trust region bound constraint is

$$\|s\| \le \Delta_k, \tag{1.7}$$

where Δ_k is the trust region radius at the kth iteration. It is easy to see that there is a possibility that the linearized constraints (1.5) may have no solutions in the trust region (1.7). To overcome this difficulty, we use a relaxed version of the linearized constraint, which was proposed by Byrd, Schnabel, and Schultz in [19]. In [20], Sun also used this relaxed version of the linearized constraint and proposed a conic trust region method for nonlinearly constrained optimization. That is, at the kth iteration, the trial step s_k is computed by solving the following conic model trust region subproblem:

$$\min_{s\in R^n} \phi_k(s) = \frac{g_k^T s}{1 - a_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - a_k^T s)^2}, \tag{1.8}$$

$$\text{s.t.} \quad A_k^T s + \theta_k C_k = 0, \tag{1.9}$$

$$\|s\| \le \Delta_k, \tag{1.10}$$

where $\|\cdot\|$ refers to the Euclidean norm and θ_k is a relaxation parameter. $\theta_k \in (0, 1]$ is chosen such that the feasible set of (1.9) and (1.10) is not empty. Geometrically speaking, the role of θ_k is to compress the feasible area of each constraints of (1.5) to the direction of origin (see [8, 21]).

The above conic model (1.8) has only one parameter a_k and less degree of freedom, which affects the effect of the model approaching the objective function. Therefore, we consider selecting a fractional model with more parameters, which can make full use of the function and gradient information in the previous iteration and can approximate the original function well, thus obtaining a new method for solving the optimization problem. In [22], Zhu considered the unconstrained optimization problem and proposed a fractional model

$$\psi_k(s) = h_k(s)g_k^T s + \frac{1}{2}h_k^2(s)s^T B_k s, \tag{1.11}$$

where

$$h_k(s) = \frac{(1 + c_k^T s)}{(1 - a_k^T s)(1 - b_k^T s)}, \tag{1.12}$$

and horizontal vectors $a_k, b_k, c_k \in R^n$ are bounded. The gradient of $\psi_k(s)$ is

$$\nabla \psi_k(s) = \frac{1}{\nu_s^3}\omega(s)\bigl[\nu_s g_k + \bigl(1 + c_k^T s\bigr)B_k s\bigr], \tag{1.13}$$

where

$$\nu_s = \bigl(1 - a_k^T s\bigr)\bigl(1 - b_k^T s\bigr), \tag{1.14}$$

$$\omega(s) = \bigl(1 + c_k^T s\bigr)\nu_s I + \bigl[c_k \nu_s + \bigl(1 + c_k^T s\bigr)\bigl(a_k\bigl(1 - b_k^T s\bigr) + b_k\bigl(1 - a_k^T s\bigr)\bigr)\bigr]s^T. \tag{1.15}$$

If $b_k = c_k = 0$, then $\psi_k(s)$ is reduced to the conic model. If $a_k = b_k = c_k = 0$, then $\psi_k(s)$ is the quadratic model $\varrho_k(s)$. The fractional model is new and it is an extension to conic model $\phi_k(s)$. It has more choice of parameter vectors and can make use of their function and gradient information. Based the new fractional model $\psi_k(s)$, a simplified fractional trust region subproblem

$$\min_{s \in R^n} \psi_k(s), \tag{1.16}$$

$$\text{s.t.} \quad \|s\| \le \tilde{\Delta}_k \tag{1.17}$$

is proposed, where

$$\tilde{\Delta}_k = \min\left\{\Delta_k, \frac{\epsilon_1}{\|a_k\|}, \frac{\epsilon_1}{\|b_k\|}, \frac{\epsilon_1}{\|c_k\|}\right\}, \tag{1.18}$$

$$0 < \epsilon_1 < 1, \tag{1.19}$$

and the parameters a_k, b_k, c_k were chosen such that

$$\|a_k\|\tilde{\Delta}_k \le \epsilon_1, \qquad \|b_k\|\tilde{\Delta}_k \le \epsilon_1, \qquad \|c_k\|\tilde{\Delta}_k \le \epsilon_1. \tag{1.20}$$

Subproblem (1.16)–(1.17) is solved only in quasi-Newton direction in [22]. Based on this, we study in depth the Newton point and the steepest descent point of the fractional trust region subproblem, so as to construct a simple dogleg method to solve the subproblem (see [23]). Numerical experiments show that the fractional model trust region quasi-Newton algorithm seems to be superior to the conic model trust region algorithm in terms of the number of iterations and the running time as the dimension of the optimization problem increases. For the linear equality constrained optimization problem, the null space technique is used to delete the linear equality constraint, and the fractional trust region method for solving the linear equality constrained optimization problem is proposed (see [24]).

Now we use the fractional model to solve nonlinearly constrained optimization problems. In order to solve problem (1.1)–(1.2), we consider the following fractional trust region subproblem. That is, if the current iteration point is x_k, then the trial step s_k is computed by

$$\min_{s\in R^n} \psi_k(s), \tag{1.21}$$

$$\text{s.t.} \quad A_k^T s + \theta_k C_k = 0, \tag{1.22}$$

$$\left|\left(1 - a_k^T s\right)\left(1 - b_k^T s\right)\right| \geq \varepsilon_0, \tag{1.23}$$

$$\|s\| \leq \Delta_k, \tag{1.24}$$

where $\varepsilon_0 \in (0,1)$. The purpose of adding constraint (1.23) is to guarantee that the objective function $\psi_k(s)$ is bounded in the trust region and this also increases the difficulty of calculation.

We know that the exact solution of the trust region subproblem is often difficult to obtain, so many approximate solution methods have been spawned. For example, the trust region method is often combined with dogleg method, conjugate gradient method, inexact line search method, alternate direction search method, and other methods to obtain approximate solutions of trust region subproblems. In this paper, we also consider an approximate solution method for the subproblem of the trust region. First, the null space technique is used to remove the linear equality constraints of the trust region subproblem; then the fractional model of the subproblem is reduced to a quadratic model by cyclically fixing the coefficient part; finally, the problem can be easily solved by searching in the descending direction to obtain an approximate solution to the subproblem.

In the global algorithm, we propose a quasi-Newton trust region method with a fractional model and prove the global convergence of the new algorithm. The numerical experiment shows that the new algorithm is effective and robust, especially for large-scale test problems.

The organization of this paper is as follows. In Sect. 2, the fractional trust region subproblem and an algorithm for solving the subproblem are presented. In Sect. 3, we propose a quasi-Newton method with a fractional model for nonlinearly equality constrained optimization and prove its convergence under some reasonable conditions. Numerical results are given in Sect. 4.

Throughout this paper, we use $\|\cdot\|$ for the 2-norm.

2 The algorithm of fractional trust region subproblem

In order to solve (1.21)–(1.24), firstly we consider removing constraint (1.23) by the same process of simplification as in [22]. Therefore, when the parameters a_k, b_k, c_k satisfy (1.20) where

$$0 < \epsilon_1 \leq 1 - \sqrt{\varepsilon_0} < 1, \tag{2.1}$$

then subproblem (1.21)–(1.24) can be rewritten as the following reduced subproblem:

$$\min_{s\in R^n} \psi_k(s), \tag{2.2}$$

$$\text{s.t.} \quad A_k^T s + \theta_k C_k = 0, \tag{2.3}$$

$$\|s\| \leq \tilde{\Delta}_k, \tag{2.4}$$

where $\tilde{\Delta}_k$ is defined as (1.18).

The null-space technique (see [20, 25, 26]) is an important technique for solving optimization problems with equality constraints. In the following, we show the method to eliminate constraint (1.22). Assume that A_k has full column rank, and there exist an orthogonal matrix Q_k and a nonsingular upper triangular matrix R_k such that

$$A_k = Q_k R_k = \begin{bmatrix} Q_k^{(1)} & Q_k^{(2)} \end{bmatrix} \begin{bmatrix} R_k^{(1)} \\ 0 \end{bmatrix} = Q_k^{(1)} R_k^{(1)}, \tag{2.5}$$

where $Q_k^{(1)} \in R^{n\times m}$, $Q_k^{(2)} \in R^{n\times(n-m)}$, and $R_k^{(1)} \in R^{m\times m}$. Then (1.22) can be rewritten as

$$\bigl(R_k^{(1)}\bigr)^T \bigl(Q_k^{(1)}\bigr)^T s = -\theta_k C_k. \tag{2.6}$$

Therefore the feasible point for (1.22) can be presented by

$$s = \tilde{s}_k + Q_k^{(2)} u \tag{2.7}$$

for any $u \in R^{n-m}$, where

$$\tilde{s}_k = -\theta_k Q_k^{(1)} \bigl(R_k^{(1)}\bigr)^{-T} C_k, \tag{2.8}$$

and $Q_k^{(2)} u$ lies in the null space of A_k. We denote

$$l_k = -Q_k^{(1)} \bigl(R_k^{(1)}\bigr)^{-T} C_k, \tag{2.9}$$

then

$$\tilde{s}_k = \theta_k l_k. \tag{2.10}$$

In order to ensure that s lies in the trust region, we choose θ_k such that the norm of $\tilde{s}_k$ is at most $\iota\tilde{\Delta}_k$, where $\iota \in (0,1)$ is a given constant. That is,

$$\|\tilde{s}_k\| = \theta_k \|l_k\| \leq \iota\tilde{\Delta}_k. \tag{2.11}$$

Define

$$\theta_k = \min\left\{1, \frac{\iota\tilde{\Delta}_k}{\|l_k\|}\right\}, \qquad \widehat{\Delta}_k = \sqrt{\tilde{\Delta}_k^2 - \|\tilde{s}_k\|^2}. \tag{2.12}$$

Then the fractional trust region subproblem (2.2)–(2.4) becomes

$$\min_{u\in R^{n-m}} \tilde{\psi}_k(u) = h_k\big(s(u)\big)g_k^T s(u) + \frac{1}{2}h_k^2\big(s(u)\big)s(u)^T B_k s(u), \tag{2.13}$$

$$\text{s.t.} \quad \|u\| \le \widehat{\Delta}_k, \tag{2.14}$$

where $s(u) = \tilde{s}_k + Q_k^{(2)}u$.

Remark 1 $\|u\| \le \widehat{\Delta}_k$ is equivalent to $\|s\| \le \tilde{\Delta}_k$. It is easy to see that if $\|u\| \le \widehat{\Delta}_k$, then from (2.7), (2.12), and $\|Q_k^{(2)}\| = 1$, we have

$$\|s\| = \sqrt{\|\tilde{s}_k\|^2 + \big\|Q_k^{(2)}u\big\|^2} \le \sqrt{\|\tilde{s}_k\|^2 + \|u\|^2} \le \sqrt{\|\tilde{s}_k\|^2 + \widehat{\Delta}_k^2} = \tilde{\Delta}_k.$$

On the other hand, if $\|s\| \le \tilde{\Delta}_k$, then from (2.5) we have

$$\|u\| = \big\|Q_k^{(2)}u\big\| = \sqrt{\|s\|^2 - \|\tilde{s}_k\|^2} \le \sqrt{\tilde{\Delta}_k^2 - \|\tilde{s}_k\|^2} = \widehat{\Delta}_k.$$

Therefore, if u_* is the solution of (2.13)–(2.14), then $s_* = \tilde{s}_k + Q_k^{(2)}u_*$ is the solution of (2.2)–(2.4), whereas if s_* is the solution of (2.2)–(2.4), then $u_* = (Q_k^{(2)})^T(s_* - \tilde{s}_k)$ is the solution of (2.13)–(2.14).

In order to solve subproblem (2.13)–(2.14) by a simple method, we first fix the fractional part of $\tilde{\psi}_k(u)$ by letting $u = 0$ in $h_k(s(u))$. At the same time, we set $u = -\tau\nabla\tilde{\psi}_k(0)$, where $\tau > 0$, $\nabla\tilde{\psi}_k(0) \neq 0$. By direct computation, we know that $\nabla\tilde{\psi}_k(0) = (Q_k^{(2)})^T\nabla\psi_k(\tilde{s}_k)$, where $\nabla\psi_k(s)$ is defined by (1.13). Then from (2.7) we have that (2.13)–(2.14) reduce to the following simplified subproblem:

$$\min_{\tau\in R} \varphi_k(\tau), \tag{2.15}$$

$$\text{s.t.} \quad 0 \le \tau \le \tau_\Delta, \tag{2.16}$$

where

$$\varphi_k(\tau) = h_k(\tilde{s}_k)g_k^T s(\tau) + \frac{1}{2}h_k^2(\tilde{s}_k)s(\tau)^T B_k s(\tau) \tag{2.17}$$

$$= \frac{1}{2}h_k(\tilde{s}_k)\big(a_\tau^{(1)}\tau^2 + b_\tau^{(1)}\tau + c_\tau^{(1)}\big), \tag{2.18}$$

$$s(\tau) = \tilde{s}_k - \tau Q_k^{(2)}\big(Q_k^{(2)}\big)^T\nabla\psi_k(\tilde{s}_k), \tag{2.19}$$

$$a_\tau^{(1)} = h_k(\tilde{s}_k)\nabla\psi_k(\tilde{s}_k)^T Q_k^{(2)}\big(Q_k^{(2)}\big)^T B_k Q_k^{(2)}\big(Q_k^{(2)}\big)^T\nabla\psi_k(\tilde{s}_k), \tag{2.20}$$

$$b_\tau^{(1)} = -2\nabla\psi_k(\tilde{s}_k)^T Q_k^{(2)}\big(Q_k^{(2)}\big)^T\big(g_k + h_k(\tilde{s}_k)B_k\tilde{s}_k\big), \tag{2.21}$$

$$c_\tau^{(1)} = 2g_k^T\tilde{s}_k + h_k(\tilde{s}_k)\tilde{s}_k^T B_k\tilde{s}_k \tag{2.22}$$

and

$$\tau_\Delta = \frac{\widehat{\Delta}_k}{\|(Q_k^{(2)})^T \nabla \psi(\tilde{s}_k)\|}. \tag{2.23}$$

Combining with (1.17)–(1.20) and (2.11), we can obtain that $\|\tilde{s}_k\| \leq \tilde{\Delta}_k$ and

$$\zeta_1 \leq h_k(\tilde{s}_k) \leq \zeta_2, \tag{2.24}$$

where

$$\zeta_1 = \frac{1-\epsilon_1}{(1+\epsilon_1)^2}, \qquad \zeta_2 = \frac{1+\epsilon_1}{(1-\epsilon_1)^2}. \tag{2.25}$$

From (1.19), it is obvious that $0 < \zeta_1 < 1$ and $\zeta_2 > 1$. From lines 2 and 3 before (2.15), we know that $(Q_k^{(2)})^T \nabla \psi_k(\tilde{s}_k) \neq 0$, where $Q_k^{(2)}$ is defined in (2.5). For $Q_k^{(2)} \in R^{n\times(n-m)}$, $\nabla \psi_k(\tilde{s}_k) \in R^n$, then $Q_k^{(2)}(Q_k^{(2)})^T \nabla \psi_k(\tilde{s}_k) \in R^n$ and $Q_k^{(2)}(Q_k^{(2)})^T \nabla \psi_k(\tilde{s}_k) \neq 0$ can be proved easily. Assume that B_k is a symmetric positive definite matrix, then combining with (2.20), (2.24), and (2.25) we have

$$a_\tau^{(1)} > 0. \tag{2.26}$$

When B_k is only symmetric but not positive definite matrix, then the modified BFGS formula can be used to modify the positive semi-definite matrix to a positive definite matrix (see [25]). Hence, it is easy to see that the solution of (2.15)–(2.16) is

$$\tau^{(1)} = \begin{cases} \min\{\tau_{\text{axis}}^{(1)}, \tau_\Delta\}, & \text{if } b_\tau^{(1)} < 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.27}$$

where

$$\tau_{\text{axis}}^{(1)} = \frac{-b_\tau^{(1)}}{2a_\tau^{(1)}}. \tag{2.28}$$

We know that $u^{(1)} = -\tau^{(1)} \nabla \tilde{\psi}_k(0)$ is an approximate solution of (2.13)–(2.14). In order to better approximate the solution of the subproblem, we are constantly repeating the above process similarly, except for fixing the fractional part. That is, in the ith iteration we substitute $u = -\tau^{(i)} \nabla \tilde{\psi}_k(0)$ into $h_k(s(u))$, where $i = 2, 3, \ldots$. We denote

$$h_k^{(i)} = h_k\bigl(s\bigl(-\tau^{(i)} \nabla \tilde{\psi}_k(0)\bigr)\bigr). \tag{2.29}$$

Then we can obtain a similar trust-region subproblem to (2.15)–(2.16). That is,

$$\min_{\tau \in R} \varphi_k^{(i)}(\tau), \tag{2.30}$$

$$\text{s.t.} \quad 0 \leq \tau \leq \tau_\Delta, \tag{2.31}$$

where

$$\varphi_k^{(i)}(\tau) = h_k^{(i)} g_k^T s(\tau) + \frac{1}{2}\big(h_k^{(i)}\big)^2 s(\tau)^T B_k s(\tau) \tag{2.32}$$

$$= \frac{1}{2} h_k^{(i)}\big(a_\tau^{(i)}\tau^2 + b_\tau^{(i)}\tau + c_\tau^{(i)}\big), \tag{2.33}$$

$$a_\tau^{(i)} = h_k^{(i)} \nabla\psi_k(\tilde{s}_k)^T Q_k^{(2)}\big(Q_k^{(2)}\big)^T B_k Q_k^{(2)}\big(Q_k^{(2)}\big)^T \nabla\psi_k(\tilde{s}_k), \tag{2.34}$$

$$b_\tau^{(i)} = -2\nabla\psi_k(\tilde{s}_k)^T Q_k^{(2)}\big(Q_k^{(2)}\big)^T\big(g_k + h_k^{(i)} B_k \tilde{s}_k\big), \tag{2.35}$$

$$c_\tau^{(i)} = 2 g_k^T \tilde{s}_k + h_k^{(i)} \tilde{s}_k^T B_k \tilde{s}_k. \tag{2.36}$$

By the computation, we have the solution of subproblem (2.30)–(2.31) as follows:

$$\tau^{(i)} = \begin{cases} \min\{\tau_{\text{axis}}^{(i)}, \tau_\Delta\}, & \text{if } b_\tau^{(i)} < 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.37}$$

where

$$\tau_{\text{axis}}^{(i)} = \frac{-b_\tau^{(i)}}{2a_\tau^{(i)}}. \tag{2.38}$$

However, if $\tau^{(i)} = 0$, then subproblem (2.30)–(2.31) has no positive real roots. For this case, we set $b_k = c_k = 0$, and then the fractional model reduces to the conic model. Subproblem (1.21)–(1.24) can be solved by Algorithm 2.4 in [26]. That is, we calculate

$$\tilde{a}_k = \big(Q_k^{(2)}\big)^T a_k, \qquad \bar{a} = \frac{\tilde{a}_k}{1 - a_k^T \tilde{s}_k}, \qquad \bar{g} = \frac{g_k}{1 - a_k^T \tilde{s}_k},$$

$$\bar{B} = \frac{B_k}{(1 - a_k^T \tilde{s}_k)^2}, \qquad \alpha_1 = 1 - \|\bar{a}\|^2 \widehat{\Delta}^2,$$

$$\widehat{\Delta} = \min\left\{\sqrt{\tilde{\Delta}_k^2 - \|\tilde{s}_k\|^2}, \frac{1 - a_k^T \tilde{s}_k - \varepsilon_0}{\|a_k\|}\right\}, \qquad m = \frac{\|\bar{a}\|\widehat{\Delta}^2}{\alpha_1}, \qquad \Delta_1 = \frac{\widehat{\Delta}}{\sqrt{\alpha_1}},$$

$$\hat{g} = \big(Q_k^{(2)}\big)^T \bar{g} + \bar{a}\tilde{s}_k^T \bar{g} + \bar{a}\tilde{s}_k^T \bar{B}\tilde{s}_k + \big(Q_k^{(2)}\big)^T \bar{B}\tilde{s}_k, \qquad V = \operatorname{diag}(\sqrt{\alpha_1}, 1, \ldots, 1),$$

$$\hat{B} = \big(Q_k^{(2)}\big)^T \bar{B} Q_k^{(2)} + \tilde{s}_k^T \bar{B}\tilde{s}_k \bar{a}\bar{a}^T + \bar{a}\tilde{s}_k^T \bar{B} Q_k^{(2)} + \big(Q_k^{(2)}\big)^T \bar{B}\tilde{s}_k \bar{a}^T,$$

$$g_1 = V^{-1} Z\hat{g} + m V^{-1} Z\hat{B} Z^T e_1, \qquad B_1 = V^{-1} Z\hat{B} Z^T V^{-1},$$

where Z is an orthonormal rotation matrix and satisfies $Z\bar{a} = \|\bar{a}e_1\|$, and $e_1 = (1, 0, \ldots, 0)^T$. Then the solution of (2.13)–(2.14) is

$$u_* = \frac{(1 - a_k^T \tilde{s}_k)(Z^T V d^* + m Z^T e_1)}{1 - a_k^T \tilde{s}_k + \tilde{a}_k^T (Z^T V d^* + m Z^T e_1)}, \tag{2.39}$$

where d^* is the solution of the quadratic trust region subproblem

$$\min_{d \in R^n} g_1^T d + \frac{1}{2} d^T B_1 d,$$

$$\text{s.t.} \quad \|d\| \le \Delta_1.$$

Now we give an algorithm for solving the fractional trust region subproblems (2.13)–(2.14).

Subalgorithm 2.1

Step 0. Input the data of the kth iteration, i.e., $0 < \epsilon_1, \iota < 1$, $\varepsilon > 0$, a_k, b_k, c_k, g_k, B_k, A_k, C_k, and $\tilde{\Delta}_k$. Set $i = 1$.

Step 1. Compute $Q_k^{(1)}$, $Q_k^{(2)}$, and $R_k^{(1)}$ as defined in (2.5).

Step 2. Compute l_k, θ_k, $\widehat{\Delta}_k$, and $\tilde{s}_k$ as defined in (2.9)–(2.12). Substitute $\tilde{s}_k$ into formula (1.13) and obtain $\nabla\psi_k(\tilde{s}_k)$.

Step 3. Calculate τ_*.

Step 3.1 Compute $h_k(\tilde{s}_k)$ and $\nabla\psi_k(\tilde{s}_k)$. Compute $a_\tau^{(1)}$, $b_\tau^{(1)}$, $c_\tau^{(1)}$, and τ_Δ from (2.20)–(2.23). Solve (2.15)–(2.16) obtaining $\tau^{(1)}$ as defined in (2.27). If $\tau^{(1)} = 0$, go to Step 5; otherwise, $i = i + 1$, go to Step 3.2.

Step 3.2 Compute $h_k^{(i)}$ and $a_\tau^{(i)}$, $b_\tau^{(i)}$, $c_\tau^{(i)}$ from (2.29) and (2.34)–(2.36). Solve (2.30)–(2.31) obtaining $\tau^{(i)}$ as defined in (2.37). If $\tau^{(i)} = 0$, go to Step 5; otherwise, $i = i + 1$, go to Step 3.3.

Step 3.3 If $|\tau^{(i)} - \tau^{(i-1)}| < \varepsilon$, then $\tau_* = \tau^{(i)}$, and stop. Otherwise, go to Step 3.2.

Step 4. Calculate $u_* = -\tau_*(Q_k^{(2)})^T\nabla\psi_k(\tilde{s}_k)$, then $s_* = \tilde{s}_k + Q_k^{(2)}u_*$.

Step 5. Set $b_k = c_k = 0$. Calculate u_* as defined in (2.39), then $s_* = \tilde{s}_k + Q_k^{(2)}u_*$.

From the above analysis, we know that the steepest descent point u_* is an approximate solution of the fractional trust region subproblem (2.13)–(2.14).

3 Global convergence

In this section, we propose a quasi-Newton method with a fractional model for nonlinearly equality constrained optimization and prove its convergence under some reasonable conditions. In order to solve problem (1.1)–(1.2), we approximate $f(x)$ with a fractional model of the form

$$m_k(s) = f_k + \frac{(1 + c_k^T s)g_k^T s}{(1 - a_k^T s)(1 - b_k^T s)} + \frac{1}{2}\frac{(1 + c_k^T s)^2 s^T B_k s}{(1 - a_k^T s)^2(1 - b_k^T s)^2}, \tag{3.1}$$

where $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, $B_k \in R^{n\times n}$ is a positive definite matrix, and $a_k, b_k, c_k \in R^n$ are parameter vectors. We choose these vectors such that (3.1) satisfies the following conditions:

$$m_k(0) = f_k, \qquad \nabla m_k(0) = g_k, \tag{3.2}$$

$$m_k(-s_{k-1}) = f_{k-1}, \qquad \nabla m_k(-s_{k-1}) = g_{k-1}, \tag{3.3}$$

where $x_k = x_{k-1} + s_{k-1}$. Obviously, (3.2) holds. Then from (3.3) we have

$$\begin{cases} f_{k-1} = f_k - \frac{(1-c_k^T u_{k-1})g_k^T s_{k-1}}{v_{k-1}} + \frac{(1-c_k^T s_{k-1})^2 s_{k-1}^T B_k s_{k-1}}{2v_{k-1}^2}, & (3.4)\\ g_{k-1} = \frac{1}{v_{k-1}^3}Q_{k-1}[v_{k-1}g_k - (1 - c_k^T s_{k-1})B_k s_{k-1}], & (3.5)\end{cases}$$

where $v_{k-1} = (1 - a_k^T s_{k-1})(1 - b_k^T s_{k-1})$ and

$$Q_{k-1} = (1 - c_k^T s_{k-1})v_{k-1}I - [c_k v_{k-1} + (1 - c_k^T s_{k-1})(a_k(1 + B_k^T s_{k-1}) + B_k(1 + a_k^T s_{k-1}))]s_{k-1}^T. \tag{3.6}$$

We choose

$$a_k = k_1 g_{k-1}, \qquad b_k = k_2 B_{k-1} s_{k-1}, \qquad c_k = k_3 g_k, \tag{3.7}$$

where k_1, k_2, k_3 are unknown parameters, and details of the choice of parameters k_1, k_2, k_3 can be found in [24].

The merit function we applied is the L_1 exact penalty function

$$P(x) = f(x) + \sigma \|C(x)\|_1, \tag{3.8}$$

where $\sigma > 0$ is a penalty parameter. We know that, for σ sufficiently large, any strong local minimizer of (1.1)–(1.2) is a local minimizer of $P(x)$ (see [27]). It is found that this function is very convenient to be used as a merit function to force global convergence in line search type algorithms (for example, see [1]). We define the actual reduction in the merit function by

$$\mathrm{Ared}_k = P(x_k) - P(x_k + s_k), \tag{3.9}$$

where s_k is a trial step computed by the algorithm at x_k. The above choice of actual reduction is used to prove the global convergence of the algorithm conveniently. Correspondingly, the predicted reduction is defined as

$$\mathrm{Pred}_k = \psi_k(0) - \psi_k(s_k) + \sigma_k(\|C_k\|_1 - \|C_k + A_k^T s_k\|_1), \tag{3.10}$$

where $\psi_k(s)$ is defined as (1.11). If

$$\psi_k(0) - \psi_k(s_k) \geq -\frac{\sigma_{k-1}}{2}(\|C_k\|_1 - \|C_k + A_k^T s_k\|_1), \tag{3.11}$$

we set the penalty parameter

$$\sigma_k = \sigma_{k-1}; \tag{3.12}$$

otherwise,

$$\sigma_k = \max\left[2\sigma_{k-1}, \frac{2(\psi_k(s_k) - \psi_k(0))}{\|C_k\|_1 - \|C_k + A_k^T s_k\|_1}\right]. \tag{3.13}$$

Then

$$\mathrm{Pred}_k \geq \frac{1}{2}\sigma_k(\|C_k\|_1 - \|C_k + A_k^T s_k\|_1) \tag{3.14}$$

holds for all k.

In the following algorithm, we compute the ratio of the actual reduction and the predicted reduction to choose the next iterate point and update the new trust region. Now we give the new quasi-Newton algorithm based on the fractional model (3.1).

Algorithm 3.1

Step 0. Choose $x_0 \in R^n$, $\varepsilon > 0$, $\varepsilon_0 > 0$, $\epsilon_1 \in (0,1)$, $\Delta_{\max} > 0$, $0 < \iota_1 < \iota_2 < 1$, $0 < \delta_1 < 1 < \delta_2$, $\sigma_0 > 0$, $B_0 = I$, and the initial trust region radius $\Delta_0 \in (0, \Delta_{\max}]$. Set $k = 0$.

Step 1. Stopping criterion. Compute g_k, C_k, and $Q_k^{(2)}$ as defined in (2.5). If $\|(Q_k^{(2)})^T g_k\| \leq \varepsilon$ and $\|C_k\| \leq \varepsilon$, then $x_* = x_k$, stop. If $k = 0$, go to Step 3.

Step 2. Update B_{k+1} by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{z_k z_k^T}{z_k^T s_k}, \tag{3.15}$$

where

$$z_k = \vartheta y_k + (1 - \vartheta) B_k s_k, \quad \vartheta \in [0,1],$$

$$\vartheta = \begin{cases} 1, & \text{if } y_k^T s_k \geq 0.2 s_k^T B_k s_k, \\ \frac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - y_k^T s_k}, & \text{otherwise}, \end{cases}$$

and $y_k = g_{k+1} - g_k$.

Step 3. If $k \leq 1$, then set $a_k = b_k = c_k = 0$ and $d_k = -B_k^{-1} g_k$, compute α_k such that Wolfe–Powell conditions are satisfied, and set $x_{k+1} = x_k + s_k = x_k + \alpha_k d_k$. $k = k + 1$ and go to Step 1.

Step 4. Compute

$$\begin{cases} \alpha = g_{k-1}^T s_{k-1}, & \tilde{\alpha} = g_{k-1}^T \xi_1, \quad \hat{\alpha} = g_{k-1}^T \xi_2, \\ \beta = s_{k-1}^T B_{k-1} s_{k-1}, & \tilde{\beta} = s_{k-1}^T B_{k-1} \xi_1, \quad \hat{\beta} = s_{k-1}^T B_{k-1} \xi_2, \\ \zeta = s_{k-1}^T B_k s_{k-1}, & \tilde{\zeta} = s_{k-1}^T B_k \xi_1, \quad \hat{\zeta} = s_{k-1}^T B_k \xi_2, \\ \gamma = g_k^T s_{k-1}, & \tilde{\gamma} = g_k^T \xi_1, \quad \hat{\gamma} = g_k^T \xi_2, \end{cases} \tag{3.16}$$

where ξ_1 and ξ_2 are chosen such that

$$\tilde{\alpha} = \tilde{\gamma} = \hat{\zeta} = \hat{\gamma} = 0. \tag{3.17}$$

Step 5. If $\gamma = 0$, then go to Step 6; otherwise, compute

$$\eta = \left(\gamma + \sqrt{\gamma^2 + 2\zeta (f_{k-1} - f_k)}\right)/\zeta,$$

$$\ddot{\gamma} = \eta \tilde{\beta} \zeta - \eta \beta \tilde{\zeta} - \tilde{\beta} \gamma,$$

$$\tilde{\iota} = \gamma - \eta \zeta, \qquad i = \tilde{\iota}\left(\hat{\alpha} \tilde{\beta} - \eta^2 \tilde{\zeta} \hat{\beta}\right),$$

$$\dot{\gamma} = \alpha i + \eta \hat{\alpha} \tilde{\beta} \tilde{\iota}^2.$$

If $\dot{\gamma} = 0$ or $\ddot{\gamma} = 0$, then go to Step 6; otherwise, calculate

$$k_1 = -\dot{\iota}/\dot{\gamma}, \qquad k_2 = \eta\tilde{\zeta}/\ddot{\gamma},$$
$$k_3 = \frac{1 - \eta(1 + k_1\alpha)(1 + k_2\beta)}{\gamma},$$

and a_k, b_k, c_k as determined in (3.7), go to Step 7.

Step 6. Let $b_k = c_k = 0$. Calculate

$$\rho_k = (f_{k-1} - f_k)^2 - \alpha\gamma, \tag{3.18}$$

$$\dot{\beta} = \begin{cases} \frac{(f_{k-1}-f_k)+\sqrt{\rho_k}}{-\alpha}, & \text{if } \rho_k \geq 0, \\ 1, & \text{otherwise,} \end{cases} \tag{3.19}$$

and set

$$a_k = \frac{1 - \dot{\beta}}{\alpha} g_{k-1}. \tag{3.20}$$

Step 7. If $\|a_k\| > \frac{\epsilon_1}{\tilde{\Delta}_k}$, then $a_k = \frac{\epsilon_1 a_k}{\tilde{\Delta}_k \|a_k\|}$. Update b_k and c_k in the same way so that (1.20) are satisfied.

Step 8. Solve subproblem (2.2)–(2.4) by Subalgorithm 2.1 to get s_k.

Step 9. If (3.11) holds, then $\sigma_k = \sigma_{k-1}$; otherwise, calculate σ_k as defined in (3.13).

Step 10. Compute

$$\rho_k = \frac{\text{Ared}_k}{\text{Pred}_k}, \tag{3.21}$$

where Ared_k, Pred_k are defined in (3.9) and (3.10).

Step 11. Update the trust region radius

$$\tilde{\Delta}_{k+1} = \begin{cases} \delta_1 \tilde{\Delta}_k, & \text{if } \rho_k \leq \iota_1, \\ \min\{\delta_2 \tilde{\Delta}_k, \Delta_{\max}\}, & \text{if } \rho_k \geq \iota_2 \text{ and } \|s_k\| = \tilde{\Delta}_k, \\ \tilde{\Delta}_k, & \text{otherwise.} \end{cases} \tag{3.22}$$

Step 12. If $\rho_k \geq \iota_1$, then $x_{k+1} = x_k + s_k$. Set $k = k + 1$, and go to Step 1; otherwise $x_{k+1} = x_k$, $k = k + 1$, and go to Step 6.

Remark 2 In order to ensure the positive definite transfer of the Hessian matrix, we use the modified BFGS formula in Step 2 to iterate to get the positive definite matrix B_{k+1} (see [25]).

In the following, we establish the convergence results of Algorithm 3.1. The focus of this paper is to transform the fractional model trust region subproblem of the equality constrained optimization model into a simple one-dimensional quadratic model subproblem by cyclically fixing the fractional coefficient part of the model, so as to obtain a new approximate solution method for solving the subproblem. Therefore, the framework of the global convergence proof is similar to the proof process of the conic model trust region algorithm in [20], with a major difference being the lower bound of reduction in each iteration (see Lemma 3.1), which is an important result required in the proof of convergence.

Lemma 3.1 *Suppose that* (1.20) *holds, where* ϵ_1 *satisfies* (2.1). *If* $\{x_k\}$, $\{B_k\}$, *and* $\{(A_k^T A_k)^{-1}\}$ *are uniformly bounded. Let* s_k *be the solution of subproblem* (2.2)–(2.4), *then there exist positive constants* M_1 *and* M_2 *such that*

$$\begin{aligned}\psi_k(0) - \psi_k(s_k) \geq{}& M_1 \big\| \big(Q_k^{(2)}\big)^T g_k \big\| \min\left\{ \tilde{\Delta}_k, \frac{\|(Q_k^{(2)})^T g_k\|}{\|B_k\|} \right\} \\ &- M_2\big(1 + \|B_k\|\big) \min\big\{ \tilde{\Delta}_k, \|C_k\| \big\} \end{aligned} \tag{3.23}$$

for all k.

Proof Define

$$s_k(t) = \theta_k l_k - tQ_k^{(2)}\big(Q_k^{(2)}\big)^T g_k, \tag{3.24}$$

where θ_k and l_k are defined in (2.12) and (2.9) respectively. Then we can see that $s_k(t)$ is in the feasible region of subproblem (2.2)–(2.4) for all $t \in [0, \widehat{\Delta}_k / \|(Q_k^{(2)})^T g_k\|]$. From the definition of s_k and $s_k(t)$, we have

$$\psi_k(0) - \psi_k(s_k) \geq \psi_k(0) - \psi_k\big(s_k(t)\big) \tag{3.25}$$

for all $t \in [0, \widehat{\Delta}_k / \|(Q_k^{(2)})^T g_k\|]$. From (1.20), (2.24), and $s_k(t) \leq \tilde{\Delta}_k$ it follows

$$\zeta_1 \leq h_k\big(s_k(t)\big) \leq \zeta_2, \tag{3.26}$$

where ζ_1, ζ_2 are defined in (2.25). By $\|Q_k^{(2)}\| = 1$, the Cauchy–Schwarz inequality, and for all $t \in [0, \widehat{\Delta}_k / \|(Q_k^{(2)})^T g_k\|]$, we have

$$\begin{aligned} &\psi_k(0) - \psi_k\big(s_k(t)\big) \\ &\quad= -h_k\big(s_k(t)\big)\theta_k g_k^T l_k + t h_k\big(s_k(t)\big) \big\| \big(Q_k^{(2)}\big)^T g_k \big\|^2 \\ &\qquad- \frac{1}{2}\big(h_k\big(s_k(t)\big)\big)^2 \theta_k^2 l_k^T B_k l_k - \big(h_k\big(s_k(t)\big)\big)^2 \theta_k l_k^T B_k Q_k^{(2)}\big(-t\big(Q_k^{(2)}\big)^T g_k\big) \\ &\qquad- \frac{1}{2}\big(h_k\big(s_k(t)\big)\big)^2 t^2 g_k^T Q_k^{(2)}\big(Q_k^{(2)}\big)^T B_k Q_k^{(2)}\big(Q_k^{(2)}\big)^T g_k \\ &\quad\geq -\zeta_2 \theta_k \|l_k\| \|g_k\| + t\zeta_1 \big\| \big(Q_k^{(2)}\big)^T g_k \big\|^2 \\ &\qquad- \frac{1}{2}\zeta_2^2\big(\theta_k^2 \|l_k\|^2 \|B_k\| + 2\theta_k \|l_k\| \|B_k\| \widehat{\Delta}_k + t^2 \big\| \big(Q_k^{(2)}\big)^T g_k \big\|^2 \|B_k\|\big) \\ &\quad\geq t\zeta_1 \big\| \big(Q_k^{(2)}\big)^T g_k \big\|^2 - \frac{1}{2} t^2 \zeta_2^2 \big\| \big(Q_k^{(2)}\big)^T g_k \big\|^2 \|B_k\| \\ &\qquad- \zeta_2 \theta_k \|l_k\| \|g_k\| - \zeta_2^2 \theta_k \|l_k\| \|B_k\| \big(\theta_k \|l_k\| + \widehat{\Delta}_k\big). \end{aligned} \tag{3.27}$$

By (2.12), we obtain

$$\begin{aligned}
&\max_{t\in[0,\widehat{\Delta}_k/\|(Q_k^{(2)})^T g_k\|]}\left\{t\zeta_1\left\|\left(Q_k^{(2)}\right)^T g_k\right\|^2-\frac{1}{2}t^2\zeta_2^2\left\|\left(Q_k^{(2)}\right)^T g_k\right\|^2\|B_k\|\right\}\\
&\quad\geq\frac{\zeta_1}{2}\left\|\left(Q_k^{(2)}\right)^T g_k\right\|^2\min\left\{\frac{\widehat{\Delta}_k}{\|(Q_k^{(2)})^T g_k\|},\frac{\zeta_1}{\zeta_2^2\|B_k\|}\right\}\\
&\quad\geq\frac{\zeta_1}{2}\left\|\left(Q_k^{(2)}\right)^T g_k\right\|\min\left\{\tilde{\Delta}_k,\frac{\zeta_1\|(Q_k^{(2)})^T g_k\|}{\zeta_2^2\|B_k\|}\right\}-\frac{\zeta_2}{2}\left\|\left(Q_k^{(2)}\right)^T g_k\right\|\left(\theta_k\|l_k\|\right)\\
&\quad\geq M_1\left\|\left(Q_k^{(2)}\right)^T g_k\right\|\min\left\{\tilde{\Delta}_k,\frac{\|(Q_k^{(2)})^T g_k\|}{\|B_k\|}\right\}-\zeta_2^2\theta_k\|l_k\|\|g_k\|,
\end{aligned}\tag{3.28}$$

where $0<\zeta_1<1$, $\zeta_2>1$, $M_1=\zeta_1^2/(2\zeta_2^2)$, and the second inequality is obtained from the triangular inequality properties. Then from (3.25)–(3.28) we have

$$\begin{aligned}
\psi_k(0)-\psi_k(s_k)&\geq\max_{t\in[0,\widehat{\Delta}_k/\|(Q_k^{(2)})^T g_k\|]}\psi_k(0)-\psi_k\left(s_k(t)\right)\\
&\geq M_1\left\|\left(Q_k^{(2)}\right)^T g_k\right\|\min\left\{\tilde{\Delta}_k,\frac{\|(Q_k^{(2)})^T g_k\|}{\|B_k\|}\right\}\\
&\quad-\zeta_2^2\theta_k\|l_k\|\left[\|B_k\|\left(\theta_k\|l_k\|+\widehat{\Delta}_k\right)+2\|g_k\|\right].
\end{aligned}\tag{3.29}$$

Besides, since $\{(A_k^T A_k)^{-1}\}$ are uniformly bounded, then from Lemma 3.1 in [20] we know that there exists a positive constant ζ_3 such that

$$\theta_k\|l_k\|\leq\min\left\{\iota\tilde{\Delta}_k,\zeta_3\|C_k\|\right\}$$

holds where $\iota\in(0,1)$ is defined in (2.11). Then we have

$$\theta_k\|l_k\|\leq\max\{\iota,\zeta_3\}\min\left\{\tilde{\Delta}_k,\|C_k\|\right\}.\tag{3.30}$$

The boundedness of $\{x_k\}$, rule (3.22), and inequality (3.30) imply that $\theta_k\|l_k\|+\widehat{\Delta}_k$ and $\|g_k\|$ are bounded above uniformly. For instance, if $\theta_k\|l_k\|+\widehat{\Delta}_k\leq\kappa_1$ and $\|g_k\|\leq\kappa_2$ for all sufficiently large k, then we have

$$\|B_k\|\left(\theta_k\|l_k\|+\widehat{\Delta}_k\right)+2\|g_k\|\leq\kappa_1\|B_k\|+2\kappa_2\leq\max\{\kappa_1,2\kappa_2\}\left(\|B_k\|+1\right).\tag{3.31}$$

Let

$$M_2=\zeta_2^2\max\{\iota,\zeta_3\}\max\{\kappa_1,2\kappa_2\}.$$

Then from (3.29)–(3.31) we know that (3.23) holds. Hence the theorem is proved. □

The proofs of the following Theorems 3.1–3.2 are similar to those in [20], so we only give the conclusion and omit the proofs.

Theorem 3.1 *Under the conditions of Lemma 3.1, we have*

$$\lim_{k\to\infty} \|C_k\| = 0.$$

Theorem 3.2 *Under the conditions of Lemma 3.1, we have*

$$\lim_{k\to\infty} \inf \left\| \left(Q_k^{(2)}\right)^T g_k \right\| = 0.$$

Theorem 3.3 *Let the conditions of Lemma 3.1 hold and $\lambda_k = (R_k^{(1)})^{-1}(Q_k^{(1)})^T g_k$. Then we have*

$$\lim_{k\to\infty} \inf \|g_k - A_k\lambda_k\| = 0. \tag{3.32}$$

Proof From the definition of λ_k and (2.5), we have

$$\begin{aligned} g_k - A_k\lambda_k &= \left(Q_k^{(1)}\left(Q_k^{(1)}\right)^T + Q_k^{(2)}\left(Q_k^{(2)}\right)^T\right)g_k - Q_k^{(1)}R_k^{(1)}\left(R_k^{(1)}\right)^{-1}\left(Q_k^{(1)}\right)^T g_k \\ &= Q_k^{(2)}\left(Q_k^{(2)}\right)^T g_k. \end{aligned}$$

From Theorem 3.2 it follows that (3.32) holds. □

From Theorems 3.1 and 3.3, we know that there exists a subsequence of $\{x_k\}$ produced by Algorithm 3.1 converging to the KT point of (1.1)–(1.2).

4 Numerical experiment

In this section, Algorithm 3.1 (abbreviated as FTR) is tested with some test problems, where eight problems are directly chosen from [28, 29] and are listed in Table 1.

Moreover, in order to test Algorithm 3.1 more generally, we designed three problems (Problems 9–11) where Problems 10–11 are chosen from [30, 31] and the nonlinear equality constraints are polynomials (see [26]).

9. Conic function

$$\begin{aligned} &f(x) = \sum_{i=1}^{n/2} \frac{x_{2i-1}^2 + x_{2i}^2}{(1 - x_{2i-1})^2} \\ &\text{s.t.} \quad c_i(x) = x_{2i}^2 - x_{2i-1}x_{2i} + 10 = 0, \quad 1 \le i \le n/2. \end{aligned}$$

Table 1 Test functions

Pro	Function Name	Pro	Function Name
1	HS46	2	HS47
3	HS48	4	HS49
5	HS50	6	HS51
7	HS52	8	HS53

10. Generalized Brown function

$$f(x) = \sum_{i=1}^{n/2} \left(\left(x_{2i-1}^2\right)^{(x_{2i}^2+1)} + \left(x_{2i}^2\right)^{(x_{2i-1}^2+1)} \right)$$

$$\text{s.t.} \quad c_i(x) = (3 - 2x_{i+1})x_{i+1} + 1 - x_i - 2x_{i+2} = 0, \quad 1 \le i \le n-2.$$

11. Penalty-I function

$$f(x) = 10^{-5} \sum_{i=1}^{n} (x_i - 1)^2 + \left(\sum_{i=1}^{n} x_i^2 - 0.25 \right)^2$$

$$\text{s.t.} \quad c_i(x) = 3x_i^3 + 4x_{i+1}^2 = 0, \quad 1 \le i \le n-1.$$

Our fractional model is proposed on the basis of the conic model, and it is a generalized form of the conic model. Both of them are suitable for solving the optimization problem that the objective function is non-quadratic and the curvature changes severely. At this time, the quadratic model methods often produce a poor prediction of the minimizer of the function. In order to compare the calculation results of the fractional model and the conic model, we let $b_k = c_k = 0$ in Algorithm 3.1. Then we can obtain the conic model algorithm and call this algorithm CTR. Therefore, we solve these test problems by FTR and CTR respectively and compare their results.

All the computations are carried out in Matlab R2013b on a microcomputer in double precision arithmetic. These tests use the same stopping criterion $\|(Q_k^{(2)})^T g_k\| \le \varepsilon$ and $\|C_k\| \le \varepsilon$. If the stoping criterion is not satisfied with Iter ≤ 5000, then the test is terminated. We mark these by an $*$ in such tables. The columns in the tables have the following meanings: Pro denotes the number of the test problems; n is the dimension of the test problems, Iter is the number of iterations; nf and ng are the numbers of function and gradient evaluations respectively; Qg is the value of Euclidean norm $\|(Q_k^{(2)})^T g_k\|$ at the final iteration; CPU(s) denotes the total iteration time of the algorithm in seconds.

The parameters in these algorithms are as follows:

$$\begin{aligned} &B_0 = I, \quad \epsilon_1 = 0.6, \quad \varepsilon = 10^{-6}, \quad \varepsilon_0 = 10^{-3}, \quad \Delta_0 = 2, \\ &\Delta_{\max} = 10, \quad \iota = \sqrt{0.5}, \quad \sigma_0 = 1, \quad \iota_1 = 0.01, \quad \iota_2 = 0.7, \\ &\delta_1 = 0.25, \quad \delta_2 = 1.5. \end{aligned}$$

The numerical comparison for 11 test problems is listed in Table 2. Because of the difference in the initial iteration point, we actually performed 19 experiments. In terms of the number of iterations, our algorithm FTR may be somewhat superior to CTR for 17 tests, and the two algorithms are the same in efficiency for the other 2 tests. Because FTR needs some extra algebra computation for some parameters, FTR takes more time than CTR.

We know that large-scale problems with 10,000 or more dimensions can be tested for unconstrained optimization problems. But when the test problem contains nonlinear equality constraints, the difficulty of solving the problem is greatly increased. Therefore, the dimensions of the nonlinear equality constraint problems are limited. From Table 3, we

Table 2 The numerical results of Algorithm 3.1 for some test problems

Pro	n	Starting point	Algorithm	Iter	nf/ng	Qg	CPU (s)
1	8	$(2, 1.5, 0.5, \ldots, 2, 1.5, 0.5, 2, 1.5)$	CTR	37	37/32	2.325673(−7)	0.065033
			FTR	30	30/24	8.903926(−7)	0.265808
2	5	$(2, 1.5, -1, 0.5, 2)$	CTR	12	12/12	1.923822(−8)	0.069842
			FTR	12	12/11	1.208933(−7)	0.179698
3	5	$(3, 5, -3, 3, 5)$	CTR	29	29/29	7.867587(−7)	0.104208
			FTR	28	28/27	7.233105(−7)	0.209572
4	5	$(7, 7, 5, -3, 0.2)$	CTR	28	28/24	8.816453(−7)	0.056596
			FTR	25	25/25	1.794643(−7)	0.187379
5	5	$(35, -31, 11, 5, -5)$	CTR	20	20/20	8.080796(−7)	0.054710
			FTR	19	19/19	9.688987(−7)	0.208662
6	5	$(2.5, 0.5, 2, -1, 0.5)$	CTR	10	10/9	2.088608(−7)	0.050139
			FTR	10	10/9	5.040703(−7)	0.170110
7	5	$(3, 3, 3, 3, 3)$	CTR	23	23/20	7.777555(−9)	0.058845
			FTR	20	20/19	1.238526(−9)	0.191213
8	5	$(3, 3, 3, 3, 3)$	CTR	34	34/24	2.452030(−7)	0.085232
			FTR	19	19/17	3.532659(−7)	0.189874
9	4	$(-2, 2, -2, 2)$	CTR	110	110/98	8.314982(−11)	0.163601
			FTR	74	74/67	1.493034(−8)	0.331437
	4	$(-3, -1, -3, -1)$	CTR	113	113/110	8.342504(−7)	0.167646
			FTR	26	26/23	1.415245(−8)	0.209718
	10	$(-3, -1, \ldots, -3, -1)$	CTR	100	100/92	1.493407(−9)	0.203343
			FTR	34	34/31	8.041779(−9)	0.285324
	80	$(-1, 1, \ldots, -1, 1)$	CTR	178	178/164	9.716836(−7)	0.871734
			FTR	128	128/119	1.525727(−7)	3.198384
10	6	$(-1, 1, -1, 1, -1, 1)$	CTR	22	22/19	1.976346(−7)	0.060672
			FTR	14	14/12	2.538846(−7)	0.193613
	4	$(-1.2, 1, -1.2, 1)$	CTR	27	27/24	7.132702(−9)	0.081467
			FTR	24	24/22	6.336906(−9)	0.223280
11	4	$(-1.2, 1, -1.2, 1)$	CTR	118	118/103	1.144310(−7)	0.142733
			FTR	16	16/12	9.425581(−7)	0.191315
	4	$(0.75, 0.5, 0.25, 0)$	CTR	59	59/45	3.391065(−8)	0.080867
			FTR	42	42/32	9.504125(−7)	0.230135
	4	$(-3, -1, -3, -1)$	CTR	15	15/15	1.422248(−9)	0.047370
			FTR	13	13/13	2.558427(−7)	0.180390
	4	$(3, 4, 3, 4)$	CTR	538	538/515	2.226214(−7)	0.653460
			FTR	19	19/17	2.771759(−7)	0.189182
	4	$(-1, 1, -1, 1)$	CTR	160	160/142	1.398612(−8)	0.184020
			FTR	12	12/11	6.116235(−9)	0.190648

can see that as the dimensions of each test problem range from 25 to 400, we have computed nine numerical comparisons experiments for three test questions. It can be found that our algorithm can get the minimum value of the function after a finite number of iterations, but the algorithm CTR can not, and some test problem algorithms failed. The number of iterations of FTR is less than that of CTR. That is, with the increase of the dimension of the problem, our algorithm FTR can be superior to CTR both in iteration number and algorithm stability.

5 Conclusions

The fractional model in Algorithm 3.1 is the extension of a conic model. By using more information of function and gradient from the previous iterations and choosing parameters flexibly, the fractional model can be more approximate to the original problem. When

Table 3 The numerical results of Algorithm 3.1 for some test problems ($\varepsilon = 10^{-4}$)

Pro	n	Starting point	Algorithm	Iter	nf/ng	Qg	CPU (s)
1	50	$(2, 1.5, 0.5, \dots, 2, 1.5, 0.5, 2, 1.5)$	CTR	66	66/59	7.675845(−5)	0.300518
			FTR	29	29/23	5.866993(−5)	0.627469
	101	$(2, 1.5, 0.5, \dots, 2, 1.5, 0.5, 2, 1.5)$	CTR	137	137/128	9.543858(−5)	1.619127
			FTR	37	37/29	2.970168(−5)	1.184047
	200	$(2, 1.5, 0.5, \dots, 2, 1.5, 0.5, 2, 1.5)$	CTR	$*$	$*/*$	$*$	$*$
			FTR	33	33/27	7.135361(−5)	1.942937
5	25	$(35, 11, 5, -5, \dots, 35, 11, 5, -5, 35, 11, 5)$	CTR	76	76/64	7.377510(−5)	0.232441
			FTR	58	58/55	8.181980(−5)	0.546569
	61	$(35, 11, 5, -5, \dots, 35, 11, 5, -5, 35, 11, 5)$	CTR	$*$	$*/*$	$*$	$*$
			FTR	74	74/64	7.910359(−5)	1.934600
10	30	$(-1, 1, \dots, -1, 1)$	CTR	81	81/75	5.527921(−5)	0.202645
			FTR	26	26/21	3.795910(−6)	0.355148
	100	$(-1, 1, \dots, -1, 1)$	CTR	26	26/21	1.689243(−5)	0.197726
			FTR	21	21/16	1.357598(−5)	0.657030
	200	$(-1, 1, \dots, -1, 1)$	CTR	45	45/36	7.697442(−6)	0.713368
			FTR	28	28/20	1.204414(−5)	1.409634
	400	$(-1, 1, \dots, -1, 1)$	CTR	80	80/70	1.815889(−5)	5.648017
			FTR	24	24/18	9.612549(−5)	3.302268

solving the trust region subproblem, we reduce the fractional model of the subproblem to a quadratic model by cycling the fixed coefficient part, and iteratively search in the descending direction to obtain the approximate solution of the subproblem. Solving of the subproblems is creative and the algorithm is simple. The theoretical and numerical results show that the method is effective and robust.

Acknowledgements
We are grateful to the editors and referees for their suggestions and comments.

Funding
Honglan Zhu was supported by the National Natural Science Foundation of China (NSFC) grant 11926357. Qin Ni was supported by the NSFC grant 11771210 and the Natural Science Foundation of Jiangsu Province (No. BK20141409). Xuebing Zhang was supported by the Basic Science Research Program by Natural Science Foundation of Jiangsu Province (No. BK20150420) and the Startup Foundation for Introducing Talent of NUIST.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
All authors worked in cooperation. All authors carried out the proof, read and approved the current version of the manuscript.

Author details
[1]College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, China. [2]Business School, Huaiyin Institute of Technology, Huaian, China. [3]College of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing, China.

Publisher's Note

Received: 29 October 2019 Accepted: 23 January 2020 Published online: 30 January 2020

References

1. Han, S.P.: A globally convergent method for nonlinear programming. J. Optim. Theory Appl. **22**(3), 297–309 (1977)
2. Powell, M.J.D.: Variable Metric Methods for Constrained Optimization. Springer, Berlin Heidelberg (1983)

3. Yuan, Y.X., Sun, W.Y.: Conic Methods for Unconstrained Minimization and Tensor Methods for Nonlinear Equations. Science Press, Beijing (1997)
4. Powell, M.J.D., Yuan, Y.: A trust region algorithm for equality constrained optimization. Math. Program. **49**(1), 189–211 (1990)
5. Vardi, A.: A trust region algorithm for equality constrained minimization: convergence properties and implementation. SIAM J. Numer. Anal. **22**(3), 575–591 (1981)
6. Boggs, P.T., Byrd, R.H., Schnabel, R.B.: A stable and efficient algorithm for nonlinear orthogonal distance regression. SIAM J. Sci. Stat. Comput. **8**(6), 1052–1078 (1987)
7. Toint, P.L.: Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space. IMA J. Numer. Anal. **8**, 231–252 (1988)
8. Zhang, J.Z., Zhu, D.T.: Projected quasi-Newton algorithm with trust region for constrained optimization. J. Optim. Theory Appl. **67**(2), 369–393 (1990)
9. El-Alem, M.: A robust trust-region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization. SIAM J. Optim. **5**(2), 348–378 (1995)
10. Qu, S.J., Zhou, Y.Y., Zhang, Y.L., Wahab, M.I.M., Zhang, G., Ye, Y.Y.: Optimal strategy for a green supply chain considering shipping policy and default risk. Comput. Ind. Eng. **131**, 172–186 (2019)
11. Huang, R.P., Qu, S.J., Yang, X.G., Liu, Z.M.: Multi-stage distributionally robust optimization with risk aversion. J. Ind. Manag. Optim. (2019). https://doi.org/10.3934/jimo.2019109
12. Qin, X., Yao, J.C.: A viscosity iterative method for a split feasibility problem. J. Nonlinear Convex Anal. **20**, 1497–1506 (2019)
13. Dai, Y., Yamashita, N.: Convergence analysis of sparse quasi-Newton updates with positive definite matrix completion for two-dimensional functions. Numer. Algebra Control Optim. **1**, 61–69 (2011)
14. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. Society for Industrial and Applied Mathematics, Philadelphia (2000)
15. Davidon, W.C.: Conic approximations and collinear scalings for optimizers. SIAM J. Numer. Anal. **17**(17), 268–281 (1980)
16. Sorensen, D.C.: The q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization. SIAM J. Numer. Anal. **17**(17), 84–114 (1980)
17. Zhang, X., Wen, X., Ni, Q.: Subspace trust-region algorithm with conic model for unconstrained optimization. Numer. Algebra Control Optim. **3**, 223–234 (2013)
18. Zhu, H., Ni, Q.: A simple alternating direction method for the conic trust region subproblem. Math. Probl. Eng. **2018**, Article ID 5358191 (2018)
19. Byrd, R.H., Shultz, G.A.: A trust region algorithm for nonlinearly constrained optimization. SIAM J. Numer. Anal. **24**(5), 1152–1170 (1987)
20. Sun, W.Y., Yuan, Y.X.: A conic trust-region method for nonlinearly constrained optimization. Ann. Oper. Res. **103**(1), 175–191 (2001)
21. Omojokun, E.O.: Trust region algorithms for optimization with nonlinear equality and inequality constraints. Ph.D. thesis, Boulder, CO, USA (1989)
22. Zhu, H., Ni, Q., Zeng, M.L.: A quasi-Newton trust region method based on a new fractional model. Numer. Algebra Control Optim. **5**(3), 237–249 (2015)
23. Zhu, H., Ni, Q., Dang, C., Zhang, H.: A trust region method based on the fractional model for unconstrained optimization. Sci. Sin., Math. **48**(4), 531–546 (2018)
24. Zhu, H., Ni, Q., Zhang, L., Yang, W.: A fractional trust region method for linear equality constrained optimization. Discrete Dyn. Nat. Soc. **2016**(2), 1–10 (2016)
25. Ni, Q.: Optimization Method and Program Design. Science Press, Beijing (2009)
26. Zhang, L.W., Ni, Q.: Trust region algorithm of new conic model for nonlinearly equality constrained optimization. J. Numer. Methods Comput. Appl. **31**(4), 279–289 (2010)
27. Coleman, T.F., Conn, A.R.: Second-order conditions for an exact penalty function. Math. Program. **19**(1), 178–185 (1980)
28. Schittkowski, K.: More Test Examples for Nonlinear Programming Codes. Springer, New York (1979)
29. Lukšan, L., Vlček, J.: Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems. Numer. Linear Algebra Appl. **5**(3), 219–247 (1998)
30. Zhu, M., Xue, Y., Sheng, Z.F.: A quasi-Newton type trust region method based on the conic model. Numer. Math. J. Chin. Univ. **17**(1), 36–47 (1995)
31. Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**(1), 17–41 (1981)