Adaptive Conic Trust-Region Method for Nonlinear Least Squares Problems

Yang Yang[1], Sun Wenyu[2]

(1. School of Mathematics and Physics Science, Xuzhou Institute of Technology, Xuzhou 221008, China)
(2. School of Mathematics and Computer Science, Nanjing Normal University, Nanjing 210097, China)

Abstract In this paper, a new method for nonlinear least-squares problems is presented. The method uses the quasi-Newton update of the Gauss-Newton Hessian based on a conic model. A method with adaptive trust region strategy is constructed. The method needs to solve the trust region subproblem with a conic model, which can be transformed to the trust region subproblem with a quadratic model. So the algorithm is easily implemented. The new algorithm is analyzed and its global and local superlinear convergence results is established. Numerical tests are presented that confirm the efficiency of the new algorithm.

Key words: nonlinear least squares problems, trust region method, conic model, global convergence, superlinear convergence

CLC number: O221.2 **Document code**: A **Article ID**: 1001-4616(2007)01-0013-09

求解非线性最小二乘问题的自适应锥模型信赖域算法

杨 扬[1], 孙文瑜[2]

(1. 徐州工程学院数学与物理科学学院, 江苏 徐州 221008)
(2. 南京师范大学数学与计算机科学学院, 江苏 南京 210097)

[摘要] 针对非线性最小二乘问题, 利用锥模型算法思想, 给出了海赛矩阵中二阶信息项的割线近似的不同校正公式, 并利用自适应信赖域技术给出了求解非线性最小二乘问题的自适应锥模型信赖域算法. 算法中我们允许使用非精确方法近似求解信赖域子问题. 文中给出了新算法的全局收敛性和超线性收敛性分析以及数值试验结果.

[关键词] 非线性最小二乘问题, 信赖域方法, 锥模型, 自适应, 总体收敛性, 超线性收敛性

0 Introduction

We consider the following nonlinear least squares problem

$$\min_{x\in R^n} f(\boldsymbol{x}), \tag{1}$$

where the objective function f has the following special form

$$f(\boldsymbol{x}) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(\boldsymbol{x}), \tag{2}$$

where each $r_j(\boldsymbol{x})$ is a smooth function from R^n to **R**. We refer to each r_j as a residual, and we assume throughout this paper that $m \geqslant n$.

Received date: 2005-09-28. **Revised date**: 2006-01-20.

Foundation item: Supported by the National Natural Science Foundation of China (10231060), the Special Research Found of Doctoral Program of Higher Education of China (20040319003), the Research Project of Xuzhou Institute of Technology (XKY200622).

Biography: Yang Yang, born in 1981, female, teaching assistant, majored in numerical mathematics. E-mail: yangyoung600@sina.com.

Corresponding author: Sun Wenyu, born in 1949, professor, majored in numerical mathematics. E-mail: wysun@njnu.edu.cn

Nonlinear least squares problems have been a fruitful study area; mainly because of their applications to various practical problems; for instance; the solution of nonlinear equations and data fitting We can see that the nonlinear least squares problem (1) ~ (2) is a kind of unconstrained optimization problems However; it has its special structure So it is intelligent to exploit the special structure in f and its derivatives form inimizing problem (1) ~ (2).

We first define $r: R^n \to R^m$ by

$$\boldsymbol{r}(\boldsymbol{x}) = (r_1(\boldsymbol{x}),\ r_2(\boldsymbol{x}),\ \cdots,\ r_m(\boldsymbol{x}))^{\mathrm{T}}. \tag{3}$$

Using this notation, we have

$$\boldsymbol{g}(\boldsymbol{x}) = \nabla f(\boldsymbol{x}) = J(\boldsymbol{x})^{\mathrm{T}} r(\boldsymbol{x}), \tag{4}$$

$$\boldsymbol{H}(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x}) = J(\boldsymbol{x})^{\mathrm{T}} J(\boldsymbol{x}) + \sum_{j=1}^{m} r_j(\boldsymbol{x}) \nabla^2 r_j(\boldsymbol{x}), \tag{5}$$

where $J(x)_{j,i} = \frac{\partial r_j}{\partial x_i}$. We denote $\boldsymbol{S}(\boldsymbol{x}) = \sum_{j=1}^{m} r_j(\boldsymbol{x}) \nabla^2 r_j(\boldsymbol{x})$. In many applications; it is possible to calculate $J(\boldsymbol{x})$ explicitly; then the gradient $\boldsymbol{g}(\boldsymbol{x})$ and the first part of Hessian $\boldsymbol{H}(\boldsymbol{x})$ can be easily obtained However; the computation of the second part $\boldsymbol{S}(\boldsymbol{x})$ may cost expensively

In the following paper; we denote $f_k = f(\boldsymbol{x}_k)$, $\boldsymbol{J}_k = \boldsymbol{J}(\boldsymbol{x}_k)$, $S_k = \boldsymbol{S}(\boldsymbol{x}_k)$, $g_k = \boldsymbol{g}(\boldsymbol{x}_k)$ and $\boldsymbol{H}_k = \boldsymbol{H}(\boldsymbol{x}_k)$.

At the current iterate $\boldsymbol{x}_k$, the Gauss– Newton method uses the approximation $\boldsymbol{H}_k \approx \boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{J}_k$, and employs the following quadratic model at $\boldsymbol{x}_k$

$$q(\boldsymbol{d}) = f_k + \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{d} + \frac{1}{2} \boldsymbol{d}^{\mathrm{T}} \boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{J}_k \boldsymbol{d}. \tag{6}$$

A lot of numerical experiments show that this model is effective for zero-residual and small-residual case However; for large-residual case; the model (6) is inadequate; because $\boldsymbol{S}_k$ is too significant to be ignored Thus; we can get a secant approximation of $\boldsymbol{S}_k$ by using the quasi-Newton updating formula and then we get a better approximation of $\boldsymbol{H}_k$. We assume that at $\boldsymbol{x}_k$, symmetric matrix $\boldsymbol{A}_k$ is the approximation of $\boldsymbol{S}_k$, then we can employ such a quadratic model

$$q(\boldsymbol{d}) = f_k + \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{d} + \frac{1}{2} \boldsymbol{d}^{\mathrm{T}} (\boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{J}_k + \boldsymbol{A}_k) \boldsymbol{d}. \tag{7}$$

In Dennis et al[1, 2], the update formula for $\boldsymbol{A}_k$ is given as follows

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \frac{(\hat{\boldsymbol{y}}_k - \boldsymbol{A}_k \boldsymbol{d}_k) \boldsymbol{y}_k^{\mathrm{T}} + \boldsymbol{y}_k (\hat{\boldsymbol{y}}_k - \boldsymbol{A}_k \boldsymbol{d}_k)^{\mathrm{T}}}{\boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{d}_k} - \frac{\boldsymbol{d}_k^{\mathrm{T}} (\hat{\boldsymbol{y}}_k - \boldsymbol{A}_k \boldsymbol{d}_k) \boldsymbol{y}_k \boldsymbol{y}_k^{\mathrm{T}}}{(\boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{d}_k)^2}, \tag{8}$$

where $\boldsymbol{A}_0 = 0$, $\hat{\boldsymbol{y}}_k = (\boldsymbol{J}_{k+1} - \boldsymbol{J}_k)^{\mathrm{T}} \boldsymbol{r}(\boldsymbol{x}_{k+1})$, $\boldsymbol{y}_k = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$, $\boldsymbol{d}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$.

Since Davidon[3] proposed the conic method for the unconstrained optimization, various scholars have studied this topic The conic model method with more degree of freedom can incorporate more information in iterations and it is more effective Moreover; the conic model has the rapid convergence property near the minimizer point as the quadratic model

In the next section, we describe our new algorithm. We discuss its global and local superlinear convergence results in section 2 In section 3; the numerical results are reported The final remarks are given in section 4

1 Our Algorithm

In this section we describe our new algorithm for nonlinear least squares problems At the kth iteration, the local conic model for (1) ~ (2) is as follows

$$\phi_k(\boldsymbol{d}) = f_k + \frac{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{d}}{1 + \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{d}} + \frac{1}{2} \frac{\boldsymbol{d}^{\mathrm{T}} \boldsymbol{B}_k \boldsymbol{d}}{(1 + \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{d})^2}, \tag{9}$$

where $\boldsymbol{h}_k \in R^n$ is a horizon vector satisfying $1 + \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{d} > 0$ It is easy to get

$$\nabla \phi_k(\boldsymbol{d}) = \frac{1}{1 + \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{d}} \left(I - \frac{\boldsymbol{h}_k \boldsymbol{d}^{\mathrm{T}}}{1 + \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{d}}\right) \left(\boldsymbol{g}_k + \frac{\boldsymbol{B}_k \boldsymbol{d}}{1 + \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{d}}\right), \tag{10}$$

$$\nabla^2\phi_k(\boldsymbol{d}) = \frac{\boldsymbol{B}_k - \boldsymbol{h}_k\boldsymbol{g}_k^{\mathrm{T}} - \boldsymbol{g}_k\boldsymbol{h}_k^{\mathrm{T}}}{(1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d})^2} - \frac{2\boldsymbol{B}_k\boldsymbol{d}\boldsymbol{h}_k^{\mathrm{T}} + 2\boldsymbol{h}_k\boldsymbol{d}^{\mathrm{T}}\boldsymbol{B}_k}{(1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d})^3} - \frac{\boldsymbol{h}_k\boldsymbol{h}_k^{\mathrm{T}}}{(1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d})^3}\left(2\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d} + \frac{3\boldsymbol{d}^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{d}}{1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d}}\right). \tag{11}$$

Obviously, we have that $\phi_k(\boldsymbol{0})=f_k$, $\nabla\phi_k(\boldsymbol{0})=\boldsymbol{g}_k$, $\nabla^2\phi_k(\boldsymbol{0})=\boldsymbol{B}_k-\boldsymbol{h}_k\boldsymbol{g}_k^{\mathrm{T}}-\boldsymbol{g}_k\boldsymbol{h}_k^{\mathrm{T}}$.

Assume that we get the next iteration point $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\boldsymbol{d}_k$ (how to get $\boldsymbol{d}_k$ will be given later), and the conic model at $\boldsymbol{x}_{k+1}$ is

$$\phi_{k+1}(\boldsymbol{d}) = f_{k+1} + \frac{\boldsymbol{g}_{k+1}^{\mathrm{T}}\boldsymbol{d}}{1+\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{d}} + \frac{1}{2}\frac{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{B}_{k+1}\boldsymbol{d}}{(1+\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{d})^2}. \tag{12}$$

As the analysis in [4], we want the above conic model to satisfy the following conditions

$$\phi_{k+1}(\boldsymbol{0}) = f_{k+1},\quad \nabla\phi_{k+1}(\boldsymbol{0}) = \boldsymbol{g}_{k+1}, \tag{13}$$

$$\phi_{k+1}(-\boldsymbol{d}_k) = f_k,\quad \nabla\phi_{k+1}(-\boldsymbol{d}_k) = \boldsymbol{g}_k, \tag{14}$$

$$\nabla^2\phi_{k+1}(\boldsymbol{0}) = \boldsymbol{J}_{k+1}^{\mathrm{T}}\boldsymbol{J}_{k+1} + \boldsymbol{A}_{k+1}. \tag{15}$$

We can find that (13) obviously holds. It is easy to show that (14) are satisfied by choosing $\gamma_k>0$, $\boldsymbol{h}_{k+1}$ and $\boldsymbol{B}_{k+1}$ to satisfy

$$\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{d}_k = 1-\gamma_k, \tag{16}$$

$$(\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k)\gamma_k^2 + 2(f_k-f_{k+1})\gamma_k + \boldsymbol{g}_{k+1}^{\mathrm{T}}\boldsymbol{d}_k = 0, \tag{17}$$

$$\boldsymbol{B}_{k+1}\boldsymbol{d}_k = \gamma_k\boldsymbol{g}_{k+1} - \gamma_k^2\boldsymbol{g}_k + \gamma_k^2\boldsymbol{h}_{k+1}\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_k. \tag{18}$$

Let

$$D_k = (f_k-f_{k+1})^2 - \boldsymbol{g}_{k+1}^{\mathrm{T}}\boldsymbol{d}_k\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k, \tag{19}$$

if $D_k\geqslant 0$, set $\rho_k=\sqrt{D_k}$ and $\gamma_k=\dfrac{f_k-f_{k+1}+\rho_k}{-\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k}$. Once γ_k is determined, note that

$$\boldsymbol{h}_{k+1} = \frac{(1-\gamma_k)\boldsymbol{b}_k}{\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{b}_k} \tag{20}$$

for any $\boldsymbol{b}_k\in R^n$ such that $\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{b}_k\neq 0$ will satisfy (16). In particular, since $\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k<0$, we can use $b_k=g_k$, so that

$$\boldsymbol{h}_{k+1} = \frac{(1-\gamma_k)\boldsymbol{g}_k}{\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k}. \tag{21}$$

Since $\nabla^2\phi_{k+1}(\boldsymbol{0})=\boldsymbol{B}_{k+1}-\boldsymbol{h}_{k+1}\boldsymbol{g}_{k+1}^{\mathrm{T}}-\boldsymbol{g}_{k+1}\boldsymbol{h}_{k+1}^{\mathrm{T}}$, from (15) we have that

$$(\boldsymbol{J}_{k+1}^{\mathrm{T}}\boldsymbol{J}_{k+1}+\boldsymbol{A}_{k+1})\boldsymbol{d}_k = \boldsymbol{B}_{k+1}\boldsymbol{d}_k - \boldsymbol{h}_{k+1}\boldsymbol{g}_{k+1}^{\mathrm{T}}\boldsymbol{d}_k - \boldsymbol{g}_{k+1}\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{d}_k.$$

Then, according to (16) and (18), we get that

$$(\boldsymbol{J}_{k+1}^{\mathrm{T}}\boldsymbol{J}_{k+1}+\boldsymbol{A}_{k+1})\boldsymbol{d}_k = (2\gamma_k-1)\boldsymbol{g}_{k+1} - \gamma_k^2\boldsymbol{g}_k + \boldsymbol{h}_{k+1}(\gamma_k^2\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_k - \boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_{k+1}). \tag{22}$$

From (22), we know that $\boldsymbol{A}_{k+1}$ needs to satisfy the following generalized quasi-Newton equation

$$\boldsymbol{A}_{k+1}\boldsymbol{d}_k = \tilde{\boldsymbol{y}}_k, \tag{23}$$

where $\tilde{\boldsymbol{y}}_k=(2\gamma_k-1)\boldsymbol{g}_{k+1}-\gamma_k^2\boldsymbol{g}_k+\boldsymbol{h}_{k+1}(\gamma_k^2\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_k-\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_{k+1})-\boldsymbol{J}_{k+1}^{\mathrm{T}}\boldsymbol{J}_{k+1}\boldsymbol{d}_k$.

Remark 1.1 If $D_k<0$, we set $\gamma_k=1$, and then we get $\boldsymbol{h}_{k+1}=\boldsymbol{0}$ and $\tilde{\boldsymbol{y}}_k=\boldsymbol{g}_{k+1}-\boldsymbol{g}_k-\boldsymbol{J}_{k+1}^{\mathrm{T}}\boldsymbol{J}_{k+1}\boldsymbol{d}_k$. The conic model is reduced to the quadratic one.

To determine $\boldsymbol{A}_{k+1}$ uniquely, we impose the additional condition that among all symmetric matrices satisfying the secant equation (23), $\boldsymbol{A}_{k+1}$ is, in some sense, closest to the current matrix $\boldsymbol{A}_k$. In other words, $\boldsymbol{A}_{k+1}$ is the solution to the following problem

$$\min\|\boldsymbol{A}-\boldsymbol{A}_k\| \tag{24}$$

$$\text{s.t. } \boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}},\ \boldsymbol{A}\boldsymbol{d}_k=\tilde{\boldsymbol{y}}_k. \tag{25}$$

Many matrix norms can be used in (24), and each norm gives rise to a different quasi-Newton method. Now, we use the weighted Frobenius norm to produce a quasi-Newton updating formula of $\boldsymbol{A}_{k+1}$.

Theorem 1.2 Assume that $\boldsymbol{v}_k\in R^n$ satisfies $\boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{d}_k>0$, and $\boldsymbol{W}\in R^{n\times n}$ is the nonsingular symmetric matrix satisfying the relation $\boldsymbol{W}\boldsymbol{v}_k=\boldsymbol{W}^{-1}\boldsymbol{d}_k$, then the updating

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \frac{(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)\boldsymbol{v}_k^{\mathrm{T}} + \boldsymbol{v}_k(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)^{\mathrm{T}}}{\boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{d}_k} - \frac{(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)^{\mathrm{T}}\boldsymbol{d}_k}{(\boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{d}_k)^2}\boldsymbol{v}_k\boldsymbol{v}_k^{\mathrm{T}} \tag{26}$$

is the unique solution to the minimizing problem

$$\min\{\|\boldsymbol{A}-\boldsymbol{A}_k\|_{W,F}:\boldsymbol{A}\boldsymbol{d}_k=\tilde{\boldsymbol{y}}_k,\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{A}\},$$

where $\|\boldsymbol{A}\|_{W,F}=\|\boldsymbol{WAW}\|_F$.

Proof See Theorem 5.1.7 in [5].

In particular, if we set $\boldsymbol{v}_k=\boldsymbol{d}_k$, we get PSB analogue update

$$\boldsymbol{A}_{k+1}=\boldsymbol{A}_k+\frac{(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)\boldsymbol{d}_k^{\mathrm{T}}+\boldsymbol{d}_k(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)^{\mathrm{T}}}{\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{d}_k}-\frac{(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)^{\mathrm{T}}\boldsymbol{d}_k}{(\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{d}_k)^2}\boldsymbol{d}_k\boldsymbol{d}_k^{\mathrm{T}},\tag{27}$$

and if we set $\boldsymbol{v}_k=\boldsymbol{y}_k=\boldsymbol{g}_{k+1}-\boldsymbol{g}_k$, then we get the DFP analogue update

$$\boldsymbol{A}_{k+1}=\boldsymbol{A}_k+\frac{(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)\boldsymbol{y}_k^{\mathrm{T}}+\boldsymbol{y}_k(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)^{\mathrm{T}}}{\boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{d}_k}-\frac{(\tilde{\boldsymbol{y}}_k-\boldsymbol{A}_k\boldsymbol{d}_k)^{\mathrm{T}}\boldsymbol{d}_k}{(\boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{d}_k)^2}\boldsymbol{y}_k\boldsymbol{y}_k^{\mathrm{T}}.\tag{28}$$

According to the different values of $\boldsymbol{v}_k$, we can get other different updates.

Once $\boldsymbol{A}_{k+1}$ is determined, we get the update of $\boldsymbol{B}_{k+1}$,

$$\boldsymbol{B}_{k+1}=\boldsymbol{J}_{k+1}^{\mathrm{T}}\boldsymbol{J}_{k+1}+\boldsymbol{A}_{k+1}+\boldsymbol{h}_{k+1}\boldsymbol{g}_{k+1}^{\mathrm{T}}+\boldsymbol{g}_{k+1}\boldsymbol{h}_{k+1}^{\mathrm{T}}.$$

We know that $\boldsymbol{B}_k$ is symmetric, but we cannot keep the positive definiteness of $\boldsymbol{B}_k$ for these updates. Since the trust region approach does not need $\boldsymbol{B}_k$ to be positive definite to obtain global convergence, we construct an algorithm with trust region strategy.

In the following paper, the notation $\|\cdot\|$ denotes the Euclidean norm on R^n. At the kth iteration, we add a constraint $\left\|\frac{d}{1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d}}\right\|\leqslant\Delta_k$ to the local conic model (9), that is, we can get $\boldsymbol{d}_k$ by solving the following subproblem

$$\begin{aligned}&\min\phi_k(d)=f_k+\frac{\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}}{1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d}}+\frac{1}{2}\frac{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{d}}{(1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d})^2}\\&\mathrm{s.t.}\ \left\|\frac{\boldsymbol{d}}{1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d}}\right\|\leqslant\Delta_k.\end{aligned}\tag{29}$$

If we set $\boldsymbol{w}=\frac{\boldsymbol{d}}{1+\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{d}}$, then the trust region subproblem with a conic model (29) is transformed to the following general trust region subproblem with a quadratic model

$$\begin{aligned}&\min\psi_k(\boldsymbol{w})=f_k+\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{w}+\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{w}\\&\mathrm{s.t.}\ \|\boldsymbol{w}\|\leqslant\Delta_k.\end{aligned}\tag{30}$$

The choice of the trust region radius is important to the efficiency of the trust region method, so we use the adaptive trust region method given by Zhang et al.[6] to implement our algorithm. That is, in (30), $\Delta_k=c^p\|\boldsymbol{g}_k\|\cdot\|\boldsymbol{B}_k^{-1}\|$, where $0<c<1$, and p is a nonnegative integer. $\boldsymbol{B}_k$ is a safely positive definite matrix based on Schnabel and Eskow modified Cholesky factorization[7]. We get $\boldsymbol{w}_k$ and set the trial step $\boldsymbol{d}_k=\frac{\boldsymbol{w}_k}{1-\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{w}_k}$. Then the actual reduction of $f(x)$ is defined by

$$are\boldsymbol{d}_k=f_k-f(\boldsymbol{x}_k+\boldsymbol{d}_k)\tag{31}$$

and the predictive reduction of $f(\boldsymbol{x})$ is defined by

$$pre\boldsymbol{d}_k=\phi_k(\boldsymbol{0})-\phi_k(\boldsymbol{d}_k)=\psi(\boldsymbol{0})-\psi(\boldsymbol{w}_k)=-\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{w}_k-\frac{1}{2}\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{w}_k.\tag{32}$$

Further, we define $r_k=ared_k/pred_k$ to be a measure of the improvement. If the $ared_k$ is satisfactory compared with $pred_k$, the trial step $\boldsymbol{d}_k$ is accepted and $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\boldsymbol{d}_k$. Otherwise, we reduce the trust region radius (increasing p by one) to solve the subproblem (30) again until a satisfactory trial step is obtained.

Remark 1.3 We will not solve the subproblem (30) exactly to get $\boldsymbol{w}_k$, instead we can use some inexact methods to solve it. No matter which inexact method is used, we want $\boldsymbol{w}_k$ to satisfy the following condition

$$pred_k \geqslant \beta \|\boldsymbol{g}_k\| \min\left\{\Delta_k, \frac{\|\boldsymbol{g}_k\|}{\|\boldsymbol{B}_k\|}\right\}, \tag{33}$$

where $\beta \in (0, 1]$.

Now we describe our new algorithm as follows.

Algorithm 1.4

Step 0. Given $\boldsymbol{x}_0$, $\boldsymbol{h}_0$, $\boldsymbol{A}_0$, $\varepsilon > 0$, $0 < \eta < 1$. Set $k = 0$, $p = 0$.

Step 1. Compute $r(\boldsymbol{x}_k)$, $\boldsymbol{J}_k$ and set $\boldsymbol{g}_k = \boldsymbol{J}_k^{\mathrm{T}} r(\boldsymbol{x}_k)$. If $\|\boldsymbol{g}_k\| \leqslant \varepsilon$, then stop.

Step 2. Set $f_k = \frac{1}{2} r(\boldsymbol{x}_k)^{\mathrm{T}} r(\boldsymbol{x}_k)$, $\boldsymbol{B}_k = \boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{J}_k + \boldsymbol{A}_k + \boldsymbol{h}_k \boldsymbol{g}_k^{\mathrm{T}} + \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{h}_k$.

Step 3. Solve the subproblem (30) to get $\boldsymbol{w}_k$, and set $\boldsymbol{d}_k = \frac{\boldsymbol{w}_k}{1 - \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{w}_k}$.

Step 4. Compute $ared_k$, $pred_k$ and $\boldsymbol{r}_k$.

Step 5. If $\boldsymbol{r}_k \geqslant \eta$, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$, generate $\boldsymbol{h}_{k+1}$, $\boldsymbol{A}_{k+1}$, set $k = k + 1$, $p = 0$, go to step 1. Otherwise set $p = p + 1$, go to step 3.

Remark 1.5 Generally, we set $\boldsymbol{h}_0 = \boldsymbol{0}$ and $\boldsymbol{A}_0 = \boldsymbol{0}$, that is, we use Gauss-Newton iteration to get $\boldsymbol{x}_1$.

Remark 1.6 In order to keep $1 + \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{d}_k > 0$, we just need to keep $1 - \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{w}_k > 0$. So we need to keep $\|\boldsymbol{h}_k\| \Delta_k < 1$. If the trust region radius Δ_k satisfies $\|\boldsymbol{h}_k\| \Delta_k \geqslant 1$, we will set $\Delta_k = \frac{\alpha}{\|\boldsymbol{h}_k\|}$, where $0 < \alpha < 1$.

2 Convergence Property

In this section, we discuss the global and local superlinear convergence properties of our algorithm. First of all, we would like to make the following assumptions.

Assumption 2.1 (i) The level set $L(\boldsymbol{x}_0) = \{\boldsymbol{x} \mid f(\boldsymbol{x}) \leqslant f(\boldsymbol{x}_0)\}$ is bounded for any given $\boldsymbol{x}_0 \in R^n$; $r_i(\boldsymbol{x})$ $(i = 1, 2, \cdots, m)$ is twice continuously differentiable in $L(\boldsymbol{x}_0)$ and bounded. (ii) $\|\boldsymbol{B}_k\|$ and $\|\boldsymbol{h}_k\|$ are both uniformly bounded, that is, there exist $K_1 > 0$ and $K_2 > 0$ such that $\|\boldsymbol{B}_k\| \leqslant K_1$ and $\|\boldsymbol{h}_k\| \leqslant K_2$ for all k.

Remark 2.2 If Assumption 3.1 holds and $\boldsymbol{B}_k$ is invertible, since $\|\boldsymbol{B}_k^{-1}\| \|\boldsymbol{B}_k\| \geqslant 1$, we know that there exists a positive number $K_3 > 0$ such that for all k

$$\|\boldsymbol{B}_k^{-1}\| \geqslant K_3. \tag{34}$$

Lemma 2.3 Suppose that Assumption 2.1 holds and that $\boldsymbol{w}_k$ is an inexact solution of the subproblem (30) such that (33) holds. Then Algorithm 1.4 cannot cycle infinitely between Step 3 and Step 5.

Proof Suppose that Algorithm 1.4 cycles infinitely between Step 3 and Step 5 at iteration k. We define the cycling index at iteration k by $k(i)$, then we have

$$r_{k(i)} < \eta, \quad i = 1, 2, \cdots \tag{35}$$

and $\Delta_{k(i)} = c^i \|\boldsymbol{g}_k\| \|\boldsymbol{B}_k^{-1}\| \to 0$ as $i \to \infty$. On the other hand, when $\Delta_{k(i)} \to 0$, we have $\boldsymbol{w}_{k(i)} \to 0$. Then when $\|\boldsymbol{w}_k\|$ is sufficiently close to zero, since $\|\boldsymbol{h}_k\|$ is bounded, we have $1/(1 - \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{w}_k) = 1 + O(\|\boldsymbol{w}_k\|)$. Since $r_i(\boldsymbol{x})$ $(i = 1, 2, \cdots, m)$ is twice continuously differentiable in $L(\boldsymbol{x}_0)$, then there exits $K_4 > 0$ such that $\|\boldsymbol{H}(\boldsymbol{x})\| \leqslant K_4$ holds for all $\boldsymbol{x} \in L(\boldsymbol{x}_0)$. Then from the boundedness of $\|\boldsymbol{B}_k\|$ and $\|\boldsymbol{H}(\boldsymbol{x})\|$, we obtain

$$\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{d}_k = \frac{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{w}_k}{1 - \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{w}_k} = \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{w}_k + O(\|\boldsymbol{w}_k\|^2), \tag{36}$$

$$\boldsymbol{d}_k^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}) \boldsymbol{d}_k = \frac{\boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}) \boldsymbol{w}_k}{(1 - \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{w}_k)^2} = \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}) \boldsymbol{w}_k + o(\|\boldsymbol{w}_k\|^2), \tag{37}$$

$$\begin{aligned} | ared_k - pred_k | &= | f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \boldsymbol{d}_k) + \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{w}_k + \frac{1}{2} \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{B}_k \boldsymbol{w}_k | \\ &= | - \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{d}_k - \frac{1}{2} \boldsymbol{d}_k^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}_k + \theta_k \boldsymbol{d}_k) \boldsymbol{d}_k + \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{w}_k + \frac{1}{2} \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{B}_k \boldsymbol{w}_k | \\ &= | - \frac{1}{2} \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}_k + \theta_k \boldsymbol{d}_k) \boldsymbol{w}_k + \frac{1}{2} \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{B}_k \boldsymbol{w}_k + O(\|\boldsymbol{w}_k\|^2) | \end{aligned}$$

$$\leqslant \frac{1}{2}(K_1+K_4)\|\boldsymbol{w}_k\|^2 + O(\|\boldsymbol{w}_k\|^2) = O(\|\boldsymbol{w}_k^2\|). \tag{38}$$

Then we have

$$|r_{k(i)}-1| = \frac{|ared_{k(i)} - pred_{k(i)}|}{|pred_{k(i)}|} \leqslant \frac{O(\|w_{k(i)}\|^2)}{\beta\|\boldsymbol{g}_k\|\min\left\{\|\boldsymbol{w}_{k(i)}\|, \frac{\|\boldsymbol{g}_k\|}{\|\boldsymbol{B}_k\|}\right\}}. \tag{39}$$

Since $\|\boldsymbol{w}_{k(i)}\|\to 0$ as $i\to\infty$, then from (39) we have $|r_{k(i)}-1|\to 0$ as $i\to\infty$, which implies that $r_{k(i)}\geqslant\eta$ holds for sufficiently large i. This is a contradiction to (35).

Lemma 2.4 If $\boldsymbol{B}_k$ is uniformly bounded, there exist two positive scalars c_1 and c_2 such that $c_1\leqslant\|\boldsymbol{B}_k^{-1}\|\leqslant c_2$.

Proof In the sense of modified Cholesky factorization, we have $\boldsymbol{B}_k = \boldsymbol{B}_k + \boldsymbol{E}_k$ is safely positive definite. Since $\boldsymbol{B}_k$ is uniformly bounded, there exist two positive scalars $\mu_1\leqslant\mu_2$ such that

$$0<\mu_1\leqslant\lambda_i(\boldsymbol{B}_k)\leqslant\mu_2,\quad k=1,2,\cdots$$

Then $c_1\leqslant\|\boldsymbol{B}_k^{-1}\|\leqslant c_2$ for $k=1,2,\cdots$, where $c_1=\frac{1}{\mu_2}$, $c_2=\frac{1}{\mu_1}$.

Theorem 2.5 Suppose that the conditions of Lemma 2.3 hold. If $\varepsilon=0$, Algorithm 1.4 either stops in finitely many steps or generates an infinite sequence $\{\boldsymbol{x}_k\}$ such that

$$\lim_{k\to\infty}\|\boldsymbol{g}_k\|=0 \tag{40}$$

Proof Suppose that (40) is not true, then there exists a positive scalar ε and an infinite set T, such that $\forall k\in T$, $\|\boldsymbol{g}_k\|\geqslant\varepsilon$. Since by Lemma 2.3 and Lemma 2.4, we have c^p and $\|\boldsymbol{B}_k^{-1}\|$ are both bounded away from zero, then we have Δ_k is bounded away from zero $\forall k\in T$. On the other hand, we have

$$f(\boldsymbol{x}_k)-f(\boldsymbol{x}_{k+1})\geqslant\eta\cdot pred_k\geqslant\eta\cdot\beta\cdot\|\boldsymbol{g}_k\|\min\left\{\Delta_k,\frac{\|\boldsymbol{g}_k\|}{\|\boldsymbol{B}_k\|}\right\}\geqslant\eta\cdot\beta\cdot\varepsilon\min\left\{\Delta_k,\frac{\varepsilon}{K_1}\right\}. \tag{41}$$

Since $\{f(\boldsymbol{x}_k)\}$ is monotonically decreasing and bounded below, then from (41) we have $\lim\limits_{\substack{k\to\infty\\k\in T}}\Delta_k=0$. This is a contradiction.

Next, we establish the local superlinear convergence for our algorithm.

Theorem 2.6 Suppose that Assumption 2.1 holds and the sequence $\{\boldsymbol{x}_k\}$ generated by Algorithm 1.4 converges to $\boldsymbol{x}^*$, where $\boldsymbol{H}(\boldsymbol{x}^*)$ is positive definite. If $\boldsymbol{B}_k$ is positive definite for k sufficiently large and the following condition

$$\lim_{k\to\infty}\frac{\|(\boldsymbol{B}_k-\boldsymbol{H}(\boldsymbol{x}^*))\hat{\boldsymbol{w}}_k\|}{\|\hat{\boldsymbol{w}}_k\|}=0 \tag{42}$$

holds, where $\hat{\boldsymbol{w}}_k=-\boldsymbol{B}_k^{-1}\boldsymbol{g}_k$, then $\boldsymbol{x}_k$ converges to $\boldsymbol{x}^*$ superlinearly.

Proof From the assumption we know that $\boldsymbol{B}_k=\boldsymbol{B}_k$ when k is sufficiently large. Moreover, $\hat{\boldsymbol{w}}_k=-\boldsymbol{B}_k^{-1}\boldsymbol{g}_k$ is the solution of the following subproblem

$$\begin{aligned}&\min\ \phi_k(\boldsymbol{w}) = f_k+\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{w}+\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{w}\\ &\mathrm{s.t.}\ \|\boldsymbol{w}\|\leqslant\|\boldsymbol{g}_k\|\|\boldsymbol{B}_k^{-1}\|.\end{aligned} \tag{43}$$

First of all, we show that for k sufficiently large, we have

$$\frac{f(\boldsymbol{x}_k)-f(\boldsymbol{x}_k+\boldsymbol{d}_k)}{\phi(\boldsymbol{0})-\phi(\boldsymbol{d}_k)}>\eta, \tag{44}$$

where $\boldsymbol{d}_k=\frac{\hat{\boldsymbol{w}}_k}{1-\boldsymbol{h}_k^{\mathrm{T}}\hat{\boldsymbol{w}}_k}$. It follows from (42) that

$$\boldsymbol{g}_k+\boldsymbol{H}(\boldsymbol{x}^*)\hat{\boldsymbol{w}}_k = o(\|\hat{\boldsymbol{w}}_k\|),$$

i.e.

$$\hat{\boldsymbol{w}}_k = -\boldsymbol{H}(\boldsymbol{x}^*)^{-1}\boldsymbol{g}_k+o(\|\hat{\boldsymbol{w}}_k\|),$$

then

$$\|\hat{\boldsymbol{w}}_k\| \leqslant \|\boldsymbol{H}(\boldsymbol{x}^*)^{-1}\| \cdot \|\boldsymbol{g}_k\| + o(\|\hat{\boldsymbol{w}}_k\|), \quad \frac{\|\boldsymbol{g}_k\|}{\|\hat{\boldsymbol{w}}_k\|} \geqslant \frac{1}{\|\boldsymbol{H}(\boldsymbol{x}^*)^{-1}\|} + \frac{o(\|\hat{\boldsymbol{w}}_k\|)}{\|\hat{\boldsymbol{w}}_k\|}.$$

Since $\boldsymbol{H}(\boldsymbol{x}^*)$ is positive definite, we have that $\|\hat{\boldsymbol{w}}_k\| = O(\|\boldsymbol{g}_k\|)$.

From Theorem 2.5, we know that $\boldsymbol{g}_k \to \mathbf{0}$, so we have $\hat{\boldsymbol{w}}_k \to \mathbf{0}$.

$$\begin{aligned} f(\boldsymbol{x}_k + \hat{\boldsymbol{d}}_k) - \phi(\hat{\boldsymbol{d}}_k) &= \frac{\boldsymbol{g}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k}{1 - \boldsymbol{h}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k} + \frac{1}{2} \frac{1}{(1 - \boldsymbol{h}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k)^2} \hat{\boldsymbol{w}}_k^{\mathrm{T}} \boldsymbol{M}_k \hat{\boldsymbol{w}}_k - \boldsymbol{g}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k - \frac{1}{2} \hat{\boldsymbol{w}}_k^{\mathrm{T}} \boldsymbol{B}_k \hat{\boldsymbol{w}}_k \\ &= \frac{1}{1 - \boldsymbol{h}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k} \boldsymbol{g}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k \boldsymbol{h}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k + \frac{1}{2} \hat{\boldsymbol{w}}_k^{\mathrm{T}} (\boldsymbol{H}(\boldsymbol{x}^*) - \boldsymbol{B}_k) \hat{\boldsymbol{w}}_k \\ &\quad + \frac{1}{2} \hat{\boldsymbol{w}}_k^{\mathrm{T}} (\boldsymbol{M}_k - \boldsymbol{H}(\boldsymbol{x}^*)) \hat{\boldsymbol{w}}_k + \frac{1}{2} \left(\frac{1}{(1 - \boldsymbol{h}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k)^2} - 1 \right) \hat{\boldsymbol{w}}_k^{\mathrm{T}} \boldsymbol{M}_k \hat{\boldsymbol{w}}_k, \end{aligned} \tag{45}$$

where $\boldsymbol{M}_k = \boldsymbol{H}\left(\boldsymbol{x}_k + \frac{\theta \hat{\boldsymbol{w}}_k}{1 - \boldsymbol{h}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k}\right)$ for some $\theta \in (0, 1)$. Since $\boldsymbol{h}_k$ is uniformly bounded, we have

$$\frac{1}{1 - \boldsymbol{h}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k} = 1 + O(\|\hat{\boldsymbol{w}}_k\|). \tag{46}$$

Moreover, from $\boldsymbol{g}_k \to 0$, $\hat{\boldsymbol{w}}_k \to 0$ and (42), we know that

$$\frac{|f(\boldsymbol{x}_k + \hat{\boldsymbol{d}}_k) - \phi(\hat{\boldsymbol{d}}_k)|}{\|\hat{\boldsymbol{w}}_k\|^2} \to 0 \text{ as } k \to \infty. \tag{47}$$

From Remark 2.2, we know that

$$\phi(\mathbf{0}) - \phi(\hat{\boldsymbol{d}}_k) = -\boldsymbol{g}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k - \frac{1}{2} \hat{\boldsymbol{w}}_k^{\mathrm{T}} \boldsymbol{B}_k \hat{\boldsymbol{w}}_k = \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{B}_k^{-1} \boldsymbol{g}_k - \frac{1}{2} \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{B}_k^{-1} \boldsymbol{g}_k = \frac{1}{2} \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{B}_k^{-1} \boldsymbol{g}_k \geqslant \frac{1}{2} K_3 \|\boldsymbol{g}_k\|^2. \tag{48}$$

Therefore

$$\begin{aligned} \lim_{k \to \infty} \left| \frac{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \hat{\boldsymbol{d}}_k)}{\phi(\mathbf{0}) - \phi(\hat{\boldsymbol{d}}_k)} - 1 \right| &= \lim_{k \to \infty} \frac{|f(\boldsymbol{x}_k + \hat{\boldsymbol{d}}_k) - \phi(\hat{\boldsymbol{d}}_k)|}{|\phi(\mathbf{0}) - \phi(\hat{\boldsymbol{d}}_k)|} \\ &\leqslant \lim_{k \to \infty} \frac{|f(\boldsymbol{x}_k + \hat{\boldsymbol{d}}_k) - \phi(\hat{\boldsymbol{d}}_k)|}{\frac{1}{2} K_3 \|\boldsymbol{g}_k\|^2} = \lim_{k \to \infty} \frac{|f(\boldsymbol{x}_k + \hat{\boldsymbol{d}}_k) - \phi(\hat{\boldsymbol{d}}_k)|}{\|\boldsymbol{w}_k\|^2}. \end{aligned} \tag{49}$$

The last equation of (49) holds because of $\|\hat{\boldsymbol{w}}_k\| = O(\|\boldsymbol{g}_k\|)$. From (49) and (47), we know that (44) holds for k sufficiently large, so $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \hat{\boldsymbol{d}}_k$ when k is sufficiently large. Since $\boldsymbol{g}_{k+1} = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k - \boldsymbol{B}_k \hat{\boldsymbol{w}}_k$, then by the way to prove (45), we have

$$\frac{\|\boldsymbol{g}_{k+1}\|}{\|\hat{\boldsymbol{w}}_k\|} = \frac{\|\boldsymbol{g}_{k+1} - \boldsymbol{g}_k - \boldsymbol{B}_k \hat{\boldsymbol{w}}_k\|}{\|\hat{\boldsymbol{w}}_k\|} \to 0 \tag{50}$$

From $\hat{\boldsymbol{d}}_k = \hat{\boldsymbol{w}}_k / (1 - \boldsymbol{h}_k^{\mathrm{T}} \hat{\boldsymbol{w}}_k)$ and (46) we have

$$\frac{\|\hat{\boldsymbol{d}}_k\|}{\|\hat{\boldsymbol{w}}_k\|} \to 1 \text{ as } k \to \infty. \tag{51}$$

Then, from (50) and (51), we obtain

$$\lim_{k \to \infty} \frac{\|\boldsymbol{g}_{k+1}\|}{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|} = 0. \tag{52}$$

Since $f(\boldsymbol{x})$ is twice continuously differentiable and $\boldsymbol{H}(\boldsymbol{x}^*)$ is positive definite, it is easy to show that

$$\lim_{k \to \infty} \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|} = 0 \tag{53}$$

Thus, $\boldsymbol{x}_k$ converges to $\boldsymbol{x}^*$ superlinearly.

3 Numerical Results

In this section, we test our Algorithm 1.4 on a set of standard nonlinear least-squares problems given in [8]. In the following, we use *cls* to stand for our new algorithm. In order to show the performance of *cls*, we compare *cls* with the quasi-Newton trust region algorithm using the quadratic model (7), which is denoted by *qls* in the remainder All the programs are written in MATLAB with double precision For comparison, *qls* also uses the adaptive trust region strategy.

In this paper, we use the conjugate gradient method proposed by Steihaug[9] to solve the quadratic trust region subproblems (30). It can be proved that this inexact method can keep (33) hold with $\beta=\frac{1}{2}$. For comparison, *cls* and *qls* use the same subroutine to solve the quadratic trust region subproblem.

The parameters are chosen as follows $\eta = 0\ 1$, $\varepsilon = 10^{-8}$, $c = 0\ 25$, $\alpha = 1-\varepsilon$ We denote the number of gradient evaluations by *ng*, the number of function evaluations by *nf*. *L* is the common logarithm of parameter factor to the standard initial point according to the rule given in [8]. *f* is the final objective function value that we obtain. Note that, in the following tables, f^* denotes the optimal objective function value.

From the numerical results, we have the following conclusions

1 From the numerical results in Table 2 and Table 3, we can see that for most of the problems, *cls* performs better than *qls*, and for some problems, *cls* reaches the minimum, but *qls* fails

2 From the numerical results about *cls* in Table 2 and Table 3, we can see that the numbers *nf* and *ng* for the update (28) are smaller than those required for the update (27) for most of the problems considered

3 Also, we have compared our results about *cls* in Table 2 with the results in [4], we find that our algorithm performs better

Table 1　List of test problems

No	Name of Problem	*n*	*m*
1	Rosenbrock function	2	2
2	Helical valley function	3	3
3	Powell singular function	4	4
4	Freudenstein and Roth function	2	2
5	Bard function	3	15
6	Kowalk and Osborne function	4	11
7	Waston function	6	31
8	Waston function	9	31
9	Waston function	12	31
10	Box three-dimensional function	3	10
11	Jennirich and Sampson function	2	10
12	Brown and Dennis function	4	20
13	Brown almost-linear function	40	40
14	Osborne 1 function	5	33

Table 2　Numerical results

No(L)	f^*	*cls*			*qls*		
		nf	*ng*	*f*	*nf*	*ng*	*f*
1(0)	0	74	44	9. 638 7e- 026	55	15	1 780 8e- 021
1(1)	0	96	58	3. 610 8e- 023	103	27	1 728 9e- 024
2(0)	0	27	15	4. 639 6e- 022	54	18	6 731 1e- 021
2(1)	0	30	16	1. 375 6e- 020	77	32	4 297 4e- 019
3(0)	0	36	24	3. 233 0e- 013	46	23	4 032 7e- 012
3(1)	0	46	29	1. 636 0e- 013	67	29	6 649 5e- 013
4(0)	22. 492 1…	38	15	22 4921	23	10	22 492 1
4(1)	22. 492 1…	43	23	22 4921	61	20	22 492 1
5(0)	$4.\ 107\ 43\cdots10^{-3}$	26	15	0 004 1	70	21	1. 181 5†
5(1)	8. 714 3…	501*	490	8 728 0	501*	440	8 722 5
6(0)	$1.\ 537\ 52\cdots10^{-4}$	36	20	2 118 4e- 004	54	23	2 118 4e- 004
7(0)	$1.\ 143\ 83\cdots10^{-3}$	44	19	0 001 1	83	21	0 001 1
8(0)	$6.\ 998\ 8\cdots10^{-7}$	56	37	6. 998 8e- 007	75	16	6 998 8e- 007
9(0)	$2\ 361\ 19\cdots10^{-10}$	114	95	6. 794 9e- 009	501*	464	9 852 4e- 009
10(0)	0	26	11	2 829 8e- 015	22	9	2 707 2e- 020
10(1)	0	83	31	9. 1362e- 018	409	88	7 551 8e- 013
11(0)*	62. 181…	26	17	62 1811	292	212	62 181 1
11(1)*	62. 181…	87	55	62 1811	299	89	129. 790 1†
12(0)*	42 911. 1…	40	21	4. 2911e+ 004	62	30	4 291 1e+ 004
12(1)*	42 911. 1…	45	22	4. 2911e+ 004	66	31	4 291 1e+ 004
13(0)*	0	12	8	4. 238 3e- 023	44	32	2 922 8e- 010
13(1)*	0	158	108	2 489 1e- 012	345	235	1 128 9e- 011
14(0)*	$2.\ 732\ 4\cdots10^{-5}$	94	70	2 732 4e- 005	187	48	2 732 4e- 005

*: We stop *cls* or *qls* when *nf* > 500 †: *cls* reaches the minimum, but *qls* fails *: For these problems, we stop *cls* and *qls* when $\frac{\|\boldsymbol{d}_k\|}{\|\boldsymbol{x}_{k+1}\|+\|\boldsymbol{x}_k\|}\leqslant\varepsilon$

Table 3 Numerical results

No(L)	f^*	*cls*			*qls*		
		nf	*ng*	*f*	*nf*	*ng*	*f*
1(0)	0	53	23	1.777 0e-022	86	38	8 196 3e-023
1(1)	0	129	76	7.088 5e-016	200	63	1.009 5e-023
2(0)	0	61	19	1.718 2e-019	74	27	2 168 3e-018
2(1)	0	24	14	1.167 7e-009	52	28	5.990 2e-023
3(0)	0	42	27	4.166 1e-013	57	30	2 373 8e-012
3(1)	0	74	38	2.418 0e-014	126	46	1.579 7e-012
4(0)	22.492 1…	72	22	22 492 1	216	37	22 492 1
4(1)	22.492 1…	75	27	22 492 1	274	48	22 492 1
5(0)	4.107 43…10^{-3}	25	12	0.004 1	47	16	0.004 1
5(1)	8.714 3…	501*	481	8 761 7	501*	465	8 736 3
6(0)	1.537 52…10^{-4}	41	23	2.118 4e-004	86	27	2 118 4e-004
7(0)	1.143 83…10^{-3}	48	18	0.001 1	79	21	0.001 1
8(0)	6.998 8…10^{-7}	64	42	6.998 8e-007	54	16	6 998 8e-007
9(0)	2 361 19…10^{-10}	501*	484	2.333 6e-008	501*	453	3.959 2e-007
10(0)	0	29	13	4.572 9e-024	33	11	3.281 9e-021
10(1)	0	83	35	1.016 9e-017	245	42	3.362 6e-023
11(0)*	62.181…	29	19	62 181 1	35	21	62 181 1
11(1)*	62.181…	95	56	62 181 1	346	119	129.790 1†
12(0)*	42 911.1…	58	24	4.291 1e+004	79	25	4.291 1e+004
12(1)*	42 911.1…	39	26	4.291 1e+004	82	38	4.291 1e+004
13(0)*	0	11	6	1.019 7e-011	10	7	1.023 9e-027
13(1)*	0	154	105	3.561 3e-011	313	229	0
14(0)*	2.732 4…10^{-5}	158	114	2.732 4e-005	160	46	2 732 4e-005

4 Conclusions

An adaptive conic trust region method for nonlinear least-squares problems has been presented Under certain mild conditions, we establish the global and local superlinear convergence results for the proposed method The numerical results show that the conic model in connection with the adaptive trust region strategy and approximation solution of subproblem is competitive

[References]

[1] Dennis J E, Gay D M, Welsch R E. An adaptive nonlinear least-squares algorithm[J]. ACM Transactions on Math Software, 1981(7): 348-368.

[2] Dennis J E, Gay D M, Welsch R E. Algorithms 573 NL2SOL-an adaptive nonlinear least algorithm E4[J]. ACM Transaetions on Math Software, 1981(7): 369-383.

[3] Davidon W C. Conic approximation and collinear scaling for optimizers[J]. SIAM J Numer Ana, 1980(17): 268-281.

[4] Han Q M, Sheng S B. Conic model algorithms for nonlinear least-squares problems[J]. Numerical Mathematics, A Journal of Chinese Universities, 1995(1): 48-59.

[5] Yuan Y, Sun W. Optimization Theory and Methods[M]. Beijing Science Press, 1997.

[6] Zhang X S, Zhang J L, Liao L Z, et al An adaptive trust region method and its convergence[J]. Science in China (Series A), 2002, 45: 620-631.

[7] Schnabel R B, Eskow E. A new modified cholesky factorization[J]. SIAM Journal on Scientific Computing 1990(11): 1 136-1 158.

[8] Moré J J, Garbow B S, Hillstrom K E. Testing unconstrained optimization software[J]. ACM Trans Math Software, 1981, 7(1): 17-41.

[9] Steihaug T. The conjugate gradient method and trust region in large scale optimization[J]. SIAM J Numer Anal, 1983(20): 626-637.

[责任编辑: 陆炳新]