Inverse Problems **21** (2005) 821–838 doi:10.1088/0266-5611/21/3/003

Convergence and regularity of trust region methods for nonlinear ill-posed inverse problems

Yanfei Wang[1,2] **and Yaxiang Yuan**[2]

[1] State Key Laboratory of Remote Sensing Science, Jointly sponsored by the Institute of Remote Sensing Applications, Chinese Academy of Sciences and Beijing Normal University, PO Box 9718, Beijing, 100101, People's Republic of China
[2] State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, PO Box 2719, Beijing, 100080, People's Republic of China

E-mail: yfwang@irsa.ac.cn

Received 19 August 2004, in final form 15 February 2005
Published 11 March 2005
Online at stacks.iop.org/IP/21/821

Abstract
Trust region methods have been well developed for well-posed problems, but there is little literature available on their applications to ill-posed inverse problems. In this paper, we apply trust region methods for solving nonlinear ill-posed inverse problems. In particular, we study the convergence and regularity of the standard trust region method when applying it to ill-posed problems. We also show that the trust region method is a regularization. A numerical test on inverse gravimetry is included to demonstrate our theoretical analysis and regularization property of the trust region method.

1. Introduction

In scientific and engineering computing, it is a key matter to solve nonlinear operator equations

$$F(x) = y \tag{1}$$

numerically in a stable way, where x is unknown, y is given data or observations and $F : D(F) \subset X \longrightarrow Y$ is a nonlinear operator, X and Y are both Hilbert spaces. A typical example is the following nonlinear Fredholm integral equation of the first kind

$$F(x)(t) = \int_a^b k(t, s, x(s))\,\mathrm{d}s = y(t), \qquad s \in [a, b], x \in X, \tag{2}$$

where k is a nonlinear kernel about x.

Problem (1) is said to be well-posed if: (1) for any data $y \in Y$, there exists a solution $x \in X$ such that $F(x) = y$; (2) the solution x is unique; and (3) the solution x is stable with respect to perturbations in y. However, these conditions are not necessarily satisfied in general, which is known as ill-posed problems. For instance, neither condition (1) nor condition (2) holds if F does not have a closed range. Also it is well known that if x is uniquely determined by y, the mapping $y \to x$ may still lack continuity. This gives severe numerical trouble especially when the given data y_δ are noisy,

$$\|y_\delta - y\| \leqslant \delta. \tag{3}$$

Therefore, for ill-posed problems, a regularization method needs to be used to obtain reasonable approximations to the solution x of equation (1).

After using regularization methods, the ill-posed problem is replaced by a stabilized problem (see [4, 6, 13, 16]). In particular, one solves the following unconstrained optimization problem:

$$\min_{x\in X} J_\alpha[x, y] := \|F(x) - y_\delta\|^2 + \alpha\Omega(x), \tag{4}$$

where $\alpha > 0$ is called the regularization parameter, $\Omega(x)$ serves as the stabilizer which stabilizes the minimization process and also provides *a priori* information about the solution. The above method is known as the Tikhonov regularization. The typical selection of the stabilizer is $\Omega(x) = \|x\|^2$. In this case, the solution x^α satisfies the first-order necessary condition

$$F'(x^\alpha)^*(F(x^\alpha) - y_\delta) + \alpha x^\alpha = 0, \tag{5}$$

but the above problem is still not suitable for numerical computation in general. The numerical solution of the minimization problem (4) requires the linearization of the function $F(x)$. This can be done by Taylor's expansion of $F(x)$ at the kth iteration point x_k, thus the minimization problem becomes

$$\min_{\xi\in X} J_\alpha[\xi, y] := \|y_\delta - F(x_k) - F'(x_k)\xi\|^2 + \alpha\Omega(\xi), \tag{6}$$

where ξ is the search direction, by a proper choice of the parameter α, the new iteration point can be calculated by $x_{k+1} = x_k + \xi_k$. For simplicity, in the following, we assume that $\Omega(\cdot) = \|\cdot\|^2$. Equation (6) can be solved through the first-order necessary condition

$$(F'(x_k)^*F'(x_k) + \alpha I)\xi = F'(x_k)^*(y_\delta - F(x_k)). \tag{7}$$

This equation is known as the Euler equation, which is the regularization of the linear equation

$$F'(x_k)\xi = y_\delta - F(x_k). \tag{8}$$

The key matter in equation (7) is the choice of the regularization parameter α. If α is chosen properly, problem (7) is well-posed, see, e.g., [4, 13]. However, α often has to be chosen *a posteriori*, and iterative methods for its determination have to be applied. Sometimes, it is difficult to find *a posteriori* criterion, say, the problems in remote sensing due to too much uncertainty. In such cases, we have to resort to other methods.

Apart from the Tikhonov regularization method, the Levenberg–Marquardt method is another stable method for solving nonlinear operator equations. Recently, this method has also been applied for solving the inverse ground water filtration problem and has been proved to be a regularization when the Levenberg–Marquardt parameter satisfies Morozov's discrepancy principle (see [8]).

Trust region methods have been widely used for solving nonlinear problems. The Levenberg–Marquardt method can be considered as a special case of the trust region method. But trust region methods are more direct in practical computation. In the trust region method, one adjusts the trust region radius instead of the Levenberg–Marquardt parameter. The trust region method is usually generated in the following way: first, we form the minimal least squares problem

$$\min_{x\in X} J(x) := \tfrac{1}{2}\|F(x) - y_\delta\|^2. \tag{9}$$

Then at each iteration a trial step is calculated by solving the subproblem

$$\min_{\xi\in X} \psi_k(\xi) := (\mathrm{grad}(J)_k, \xi) + \tfrac{1}{2}(\mathrm{Hess}(J)_k\xi, \xi), \tag{10}$$

$$\text{s.t. } \|\xi\| \leqslant \Delta_k, \tag{11}$$

where $\mathrm{grad}(J)_k$ is the gradient of J at the kth iterate,

$$\mathrm{grad}(J)(x) = F'(x)^*(F(x) - y_\delta), \tag{12}$$

$\mathrm{Hess}(J)_k$ is the Hessian of J at the kth iterate,

$$\mathrm{Hess}(J)(x) = F'(x)^*F'(x) + F''(x)^*(F(x) - y_\delta) \tag{13}$$

and Δ_k is the trust region radius. The trust region subproblem (TRS) (10)–(11) is an approximation to the original optimization problem (9) with a trust region constraint which prevents the trial step being too large.

In many cases, the second term of (13) is difficult to compute or the computation is too costly. The easiest way is to ignore the second term. For the Gauss–Newton method, we solve the following problem at the kth iteration:

$$F'(x_k)^*F'(x_k)\xi = -F'(x_k)^*(F(x_k) - y_\delta), \qquad x_{k+1} = x_k + \xi.$$

While for the Levenberg–Marquardt method, we solve the following problem at the kth iteration:

$$(F'(x_k)^*F'(x_k) + \alpha_k I)\xi = -F'(x_k)^*(F(x_k) - y_\delta), \qquad x_{k+1} = x_k + \xi,$$

and adjust the Levenberg–Marquardt parameter α_k in an appropriate way.

For ill-posed problems, a way of adjusting α_k is by Morozov's *posteriori* discrepancy principle, say [4, 6, 8]. Another way is by trust region methods (see sections 2 and 4 for details). In [8], the author also made tests to compare the two kinds of parameter selection methods. It was concluded that both methods could generate comparable results. However, the author further pointed out that, 'whether the standard trust region implementations of the Levenberg–Marquardt iterations are also a regularization remains a very interesting open problem'. Some concerns about the design of a suitable stopping rule are also included in that paper. This open problem was first addressed in [4]. We will solve the open problem proposed in [4, 8]. Particularly, we give a positive answer that the standard trust region methods are indeed a regularization.

Now, we give a detailed interpretation of the standard trust region method. When introducing the Gauss–Newton and Levenberg–Marquardt methods, we ignore the second term of (13). Accordingly, in the trust region method, the second term of (13) is also ignored. Thus the trust region subproblem (TRS) can be formulated as follows:

$$\min_{\xi\in X} \tilde{\psi}_k(\xi) := (\mathrm{grad}(J)_k, \xi) + \tfrac{1}{2}(\widetilde{\mathrm{Hess}}(J)_k\xi, \xi), \tag{14}$$

$$\text{s.t. } \|\xi\| \leqslant \Delta_k, \tag{15}$$

where

$$\widetilde{\mathrm{Hess}}(J)(x_k) = F'(x_k)^*F'(x_k). \tag{16}$$

The trust region algorithm considered in the paper is based on the minimization process TRS (14)–(15). A trust region algorithm generates a new point which lies in the trust region, and then decides whether it accepts a new point or rejects it. At each iteration, the trial step ξ_k is normally calculated by solving the 'trust region subproblem' (14)–(15). Here $\Delta_k > 0$ is a trust region radius. Generally, a trust region algorithm uses

$$r_k = \frac{\mathrm{Ared}_k}{\mathrm{Pred}_k} \tag{17}$$

to decide whether the trial step ξ_k is acceptable and how the next trust region radius is chosen, where

$$\mathrm{Pred}_k = \tilde{\psi}_k(0) - \tilde{\psi}_k(\xi_k) \tag{18}$$

is the predicted reduction in the approximate model, and

$$\mathrm{Ared}_k = J(x_k) - J(x_k + \xi_k) \tag{19}$$

is the actual reduction in the objective functional.

Now we give the trust region algorithm for solving nonlinear ill-posed problems.

Algorithm 1.1. *(Trust region algorithm for nonlinear ill-posed problems)*

STEP 1 Choose parameters $0 < \tau_3 < \tau_4 < 1 < \tau_1, 0 \leqslant \tau_0 \leqslant \tau_2 < 1, \tau_2 > 0$ *and initial*

values x_0, $\Delta_0 > 0$*; Set* $k := 1$.

STEP 2 If the stopping rule is satisfied *then STOP; Else, solve (14)–(15) to give* ξ_k.

STEP 3 Compute r_k*;*

$$x_{k+1} = \begin{cases} x_k & \text{if} \quad r_k \leqslant \tau_0, \\ x_k + \xi_k & \text{otherwise}. \end{cases} \tag{20}$$

Choose Δ_{k+1} *that satisfies*

$$\Delta_{k+1} \in \begin{cases} [\tau_3\|\xi_k\|, \tau_4\Delta_k] & \text{if} \quad r_k < \tau_2, \\ [\Delta_k, \tau_1\Delta_k] & \text{otherwise}. \end{cases} \tag{21}$$

STEP 4 Evaluate $\mathrm{grad}(J)_k$ *and* $\widetilde{\mathrm{Hess}}(J)_k$*;* $k := k+1$*; GOTO STEP 2.*

In STEP 2, the stopping rule is based on the discrepancy principle which will be addressed in section 3. The detailed implementation of STEP 2 will be explained in section 4.

The constants τ_i $(i = 0, \ldots, 4)$ can be chosen by users. Typical values are $\tau_0 = 0$, $\tau_1 = 2, \tau_2 = \tau_3 = 0.25, \tau_4 = 0.5$. The parameter τ_0 is usually zero or a small positive constant. The advantage of using zero τ_0 is that a trial step is accepted whenever the objective function is reduced. When the objective function is not easy to compute, it seems that we should not throw away any 'good' point that reduces the objective function (see [17] for details).

For subproblem (14)–(15), we have the following properties (see [9, 12]):

Lemma 1.2. *A vector* $\xi^* \in X$ *is a solution to (14)–(15) if and only if there exists* $\alpha^* \geqslant 0$ *such that*

$$(\widetilde{\mathrm{Hess}}(J)_k + \alpha^* I)\xi^* = -\mathrm{grad}(J)_k \tag{22}$$

and that $\widetilde{\mathrm{Hess}}(J) + \alpha^* I$ *is positive semi-definite,* $\|\xi^*\| \leqslant \Delta_k$ *and*

$$\alpha^*(\Delta_k - \|\xi^*\|) = 0. \tag{23}$$

From lemma 1.2, we see if $\widetilde{\mathrm{Hess}}(J) + \alpha^* I$ is positive definite, ξ^* is uniquely defined by

$$\xi^* = -(\widetilde{\mathrm{Hess}}(J) + \alpha^* I)^{-1}\mathrm{grad}(J). \tag{24}$$

To emphasize the fact that ξ is dependent on the parameter α, we write

$$\xi(\alpha) = -(\widetilde{\mathrm{Hess}}(J) + \alpha I)^{-1}\,\mathrm{grad}(J). \tag{25}$$

The following property has been established in [5, 19]:

Proposition 1.3. If $\widetilde{\mathrm{Hess}}(J) + \alpha I$ is positive definite, then $\|\xi(\alpha)\|$ is strictly decreasing as α increases.

The choice of the trust region method for the solution of problem (1) is motivated by the characteristics of many application problems, such as the inverse gravimetry problem (see section 5) and the parameter identification problem (see [15]), i.e., the ill-posedness of these kinds of problems. The global convergence of trust region methods for well-posed unconstrained problems is well established, see, e.g., [5, 19]. The purpose of this paper is to investigate the convergence and regularity of the trust region method for nonlinear ill-posed problems; meanwhile, the TRS is solved through (14)–(15).

It is worthwhile to note that in [14], the author proposed a constrained least squares regularization method for solving nonlinear ill-posed problems. This is perhaps the first paper to combine trust region algorithms with the regularization of ill-posed problems. Instead of the Tikhonov regularization (4), the author first forms a constrained least squares errors (LSE) problem

$$\min_{x\in X} J(x) := \tfrac{1}{2}\|F(x) - y_\delta\|^2,$$
$$\text{s.t. } c(x) \leqslant \beta^2,$$

where β is the regularization parameter, $c(x)$ is a penalty functional whose purpose is to stabilize the minimization and provide *a priori* information about the solution; hence it is called the 'regularization constraint'. Then under the six assumptions (A1)–(A6) (see [14]) the author proves the local convergence of the constrained LSE. For the numerical solution of the constrained LSE, the author first applies a Gauss–Newton approximation to the objective functional $f(x) := \|F(x) - y_\delta\|^2$ and obtains a quadratic approximation to $f(x)$; then retaining the quadratic regularization constraint and imposing an additional quadratic trust region constraint, the author employs the singular value decomposition to find the solutions of the trust region subproblem. Actually, the author only combines trust region algorithms in the numerical test and does not consider the regularization of the trust region algorithms. Moreover, the formulation of the trust region subproblem is not classical because of the imposed regularization constraint.

Let us denote a solution of (1) by x^+, $\xi_k^+ = x^+ - x_k$ is the difference between the true solution and the iterate x_k, and suppose F is Fréchet differentiable. We recall a widely used assumption for analysis of the nonlinear ill-posed problems,

Assumption 1.4. *For a certain ball $\mathcal{B} \subset D(F)$ around the exact solution x^+ of equation (1) and some $1 > d > 0$, F satisfies the local property*

$$\|F(x) - F(\hat{x}) - F'(\hat{x})(x - \hat{x})\| \leqslant d\|F(x) - F(\hat{x})\| \tag{26}$$

for all $x, \hat{x} \in \mathcal{B}$.

If we replace x by x^+, $\hat{x}$ by x_k in (26), then we obtain

$$\left\|y - F(x_k) - F'(x_k)\xi_k^+\right\| \leqslant d\|y - F(x_k)\|, \qquad 0 < d < 1. \tag{27}$$

The above condition is strong enough to ensure at least local convergence of the iterates to a solution x of (1) in $\mathcal{B}$, cf section 2.

Usually, when we deal with a nonlinear problem, we also assume that the Lipschitz conditions for F' are satisfied, i.e.,

$$\|F'(u) - F'(\hat{x})\| \leqslant \hat{d}\|u - \hat{x}\|.$$

This yields the usual Fréchet estimate

$$\|F(x) - F(\hat{x}) - F'(\hat{x})(x - \hat{x})\| \leqslant \frac{\hat{d}}{2}\|x - \hat{x}\|^2. \tag{28}$$

For ill-posed problems, (28) provides little information about the local behaviour of F around x, because the left-hand side of (28) can be significantly smaller than the right-hand side even for an arbitrarily small $\|x - \hat{x}\|$. For example, fix $x \in D(F)$ and assume that F is continuous and compact, then $F'(x)$ is compact. Hence, for every sequence $\{\hat{x}_k\}$ with $\|\hat{x}_k - x\| = \varepsilon$ (ε is arbitrarily small) for all $k \in \mathbb{N}$, the left-hand side of (28) tends to 0 as $k \to \infty$ whereas the right-hand side remains $\frac{\hat{d}}{2}\varepsilon^2$ for all k. Now it is clear that assumption 1.4 comes from the remedy of the situation induced by (28). More detailed interpretation and several examples satisfying it can be found in [2, 7]. In their papers, the constant d is restricted to $\left(0, \frac{1}{2}\right)$ for the analysis of the convergence of the nonlinear Landweber iteration. This assumption is helpful for analysing the properties of the trust region algorithm which is presented in this paper.

The paper is organized as follows: in section 2, we analyse the convergence properties of the trust region algorithm when applying to ill-posed inverse problems; in section 3, the regularity of the trust region algorithm is proved; in section 4, the numerical implementation for discrete problems is discussed; in section 5, we use the trust region algorithm for solving the inverse gravimetry problem; finally in section 6, some concluding remarks are presented.

2. Convergence properties for exact data

For simplicity, we introduce some notation in the situation when the exact data are given. We denote

$$T_k = F'(x_k), \qquad g_k = -F'(x_k)^*(y - F(x_k))$$

for the exact data, and

$$\xi_{k,\alpha_k} = -(T_k^* T_k + \alpha_k I)^{-1} g_k \tag{29}$$

as the trial step in each iteration. Then at the kth step, we solve (29) to get the step ξ_{k,α_k}.

If ξ_{k,α_k} is a solution of equations (22) and (23), then there is a unique $\alpha_k > 0$ that satisfies (22) and (23). From (29), the parameter $\alpha_k > 0$ satisfies

$$\left\|\xi_{k,\alpha_k}\right\| = \Delta_k, \tag{30}$$

i.e.,

$$\|(T_k^* T_k + \alpha_k I)^{-1} g_k\| = \Delta_k. \tag{31}$$

Denote the residual (or the discrepancy) $y - F(x_k)$ by res_k. Note that

$$T_k(T_k^* T_k + \alpha_k I)^{-1} T_k^* = T_k T_k^*(T_k T_k^* + \alpha_k I)^{-1},$$
$$(T_k T_k^* + \alpha_k I)[I - T_k T_k^*(T_k T_k^* + \alpha_k I)^{-1}] = \alpha_k I,$$

we have

$$\begin{aligned}\mathrm{res}_k - T_k\xi_{k,\alpha_k} &= \mathrm{res}_k - T_k(T_k^* T_k + \alpha_k I)^{-1} T_k^* \,\mathrm{res}_k \\ &= \mathrm{res}_k - T_k T_k^*(T_k T_k^* + \alpha_k I)^{-1} \,\mathrm{res}_k \\ &= \alpha_k (T_k T_k^* + \alpha_k I)^{-1} \,\mathrm{res}_k.\end{aligned} \tag{32}$$

By spectral analysis,

$$\|(T_k T_k^* + \alpha_k I)^{-1}\| \leqslant \frac{1}{\alpha_k}.$$

Therefore,

$$\left\| \mathrm{res}_k - T_k \xi_{k,\alpha_k} \right\| = \alpha_k \| (T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k \| \leqslant \| \mathrm{res}_k \|. \tag{33}$$

For ill-posed problems, the norm of the discrepancy, i.e., $\|y - F(x_k)\|$, is widely used to determine whether the iteration process should be carried out or not ([4, 6]). For the iteration process of the trust region algorithm for the exact data case, we use the stopping rule as follows:

If $\|y - F(x_k)\| \leqslant \epsilon$ *(the tolerance), then the iteration process is terminated.*

According to the local property of the functional $F(x)$, if $x^+, x_k \in \mathcal{B}$, then the inequality (27) is satisfied. Therefore it is reasonable to assume that before the stopping rule is satisfied, the inequality,

$$\left\| y - F(x_k) - F'(x_k)\xi_{k,\alpha_k} \right\| \geqslant c_1 \| y - F(x_k) \|, \tag{34}$$

holds for $c_1 \in (d, 1)$. Thus combining (34) with (33) we conclude that

$$c_1 \| \mathrm{res}_k \| \leqslant \left\| \mathrm{res}_k - T_k \xi_{k,\alpha_k} \right\| \leqslant \| \mathrm{res}_k \|.$$

In the following results, we show that $x_k + \xi_{k,\alpha_k}$ is a better approximation of x^+ than x_k, while its proof is similar to the one of proposition 2.1 in [8].

Theorem 2.1. *Assume that assumption 1.4 holds and* $T_k T_k^* + \alpha_k I$ *is positive definite. Then*

$$\|x^+ - x_k\|^2 - \left\| x^+ - \left(x_k + \xi_{k,\alpha_k}\right) \right\|^2 > 2(c_1 - d) \| (T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k \| \| \mathrm{res}_k \|. \tag{35}$$

Proof. From (32) it follows that

$$\begin{aligned}
\left\| \xi_{k,\alpha_k} - \xi_k^+ \right\|^2 &= \left\| \xi_{k,\alpha_k} \right\|^2 - 2\left(\xi_{k,\alpha_k}, \xi_k^+\right) + \left\| \xi_k^+ \right\|^2 \\
&= (\mathrm{res}_k, T_k T_k^* (T_k T_k^* + \alpha_k I)^{-2} \mathrm{res}_k) - 2\left((T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k, T_k \xi_k^+\right) + \left\| \xi_k^+ \right\|^2 \\
&= -(\mathrm{res}_k, T_k T_k^* (T_k T_k^* + \alpha_k I)^{-2} \mathrm{res}_k) + 2\left((T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k, \mathrm{res}_k - T_k \xi_k^+\right) \\
&\quad - 2\alpha_k (\mathrm{res}_k, (T_k T_k^* + \alpha_k I)^{-2} \mathrm{res}_k) + \left\| \xi_k^+ \right\|^2.
\end{aligned}$$

Thus we obtain

$$\begin{aligned}
\left\| \xi_k^+ \right\|^2 - \left\| \xi_{k,\alpha_k} - \xi_k^+ \right\|^2 &= (\mathrm{res}_k, T_k T_k^* (T_k T_k^* + \alpha_k I)^{-2} \mathrm{res}_k) \\
&\quad + 2\alpha_k (\mathrm{res}_k, (T_k T_k^* + \alpha_k I)^{-2} \mathrm{res}_k) - \left((T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k, \mathrm{res}_k - T_k \xi_k^+\right).
\end{aligned}$$

Recalling the assumption that $T_k T_k^* + \alpha_k I$ is positive definite and noting that

$$(\mathrm{res}_k, T_k T_k^* (T_k T_k^* + \alpha_k I)^{-2} \mathrm{res}_k) = \left\| \xi_{k,\alpha_k} \right\|^2 > 0,$$

we have

$$\begin{aligned}
\left\| \xi_k^+ \right\|^2 - \left\| \xi_{k,\alpha_k} - \xi_k^+ \right\|^2 &> 2\alpha_k \| (T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k \|^2 \\
&\quad - 2 \| (T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k \| \left\| \mathrm{res}_k - T_k \xi_k^+ \right\| \\
&= 2 \| (T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k \| \left(\alpha_k \| (T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k \| - \left\| \mathrm{res}_k - T_k \xi_k^+ \right\| \right).
\end{aligned} \tag{36}$$

By (34),

$$\alpha_k \| (T_k T_k^* + \alpha_k I)^{-1} \mathrm{res}_k \| = \left\| y - F(x_k) - F'(x_k)\xi_{k,\alpha_k} \right\| \geqslant c_1 \| \mathrm{res}_k \|, \tag{37}$$

where $c_1 \in (d, 1)$. By assumption 1.4 and (27), we obtain

$$\left\| \mathrm{res}_k - T_k \xi_k^+ \right\| \leqslant d \| \mathrm{res}_k \|, \qquad 0 < d < 1. \tag{38}$$

Combining (36), (37) and (38) yields

$$\left\|\xi_k^+\right\|^2 - \left\|\xi_{k,\alpha_k} - \xi_k^+\right\|^2 > 2(c_1 - d)\|(T_k T_k^* + \alpha_k I)^{-1}\,\mathrm{res}_k\|\|\mathrm{res}_k\|.$$

This completes the proof. □

The following lemma will indicate that the sequence $\{\alpha_k\}$ generated in each iteration is uniformly bounded.

Lemma 2.2. *Let Δ_k, g_k and α_k as in algorithm 1.1. Then there exists a constant $\omega > 0$ such that $\Delta_k \geqslant \omega\|g_k\|$. Moreover, $\{\alpha_k\}$ is uniformly bounded.*

Proof. Suppose, in contrast, that the first assertion is not true. Then

$$\liminf_{k\to\infty} \frac{\Delta_k}{\|g_k\|} = 0. \tag{39}$$

By (21) and (39), there exists a subsequence $\{k_i\}$ such that

$$\Delta_{k_i+1} < \Delta_{k_i} \tag{40}$$

and

$$\frac{\Delta_{k_i}}{\|g_{k_i}\|} \to 0 \qquad \text{as} \qquad i \to \infty. \tag{41}$$

The limit property in (41) implies that

$$\begin{aligned}
\mathrm{Pred}_{k_i} &= \psi_{k_i}(0) - \psi_{k_i}(\xi_{k_i}) \\
&\geqslant \psi_{k_i}(0) - \psi_{k_i}\left(-\frac{\Delta_{k_i}}{\|g_{k_i}\|} g_{k_i}\right) \\
&= \Delta_{k_i}\|g_{k_i}\| + O\left(\Delta_{k_i}^2\right) \\
&= \Delta_{k_i}\|g_{k_i}\|\left(1 + O\left(\frac{\Delta_{k_i}}{\|g_{k_i}\|}\right)\right).
\end{aligned} \tag{42}$$

This, together with

$$\mathrm{Ared}_{k_i} = \mathrm{Pred}_{k_i} + O\left(\Delta_{k_i}^2\right), \tag{43}$$

yields that

$$r_{k_i} = \frac{\mathrm{Ared}_{k_i}}{\mathrm{Pred}_{k_i}} \to 1 \qquad \text{as} \qquad i \to \infty. \tag{44}$$

Therefore

$$\Delta_{k_i+1} \geqslant \Delta_{k_i} \tag{45}$$

for all sufficiently large i, which contradicts (40). This proves that the first assertion is true. By the first assertion, $\frac{\|g_k\|}{\Delta_k}$ is bounded. This together with (31) leads to the second assertion, the boundedness of the sequence $\{\alpha_k\}$. □

Theorem 2.3. *Assume that assumption 1.4 holds. Given the exact data y, the sequence $\{x_k\}$ generated by the trust region algorithm converges to a solution of $F(x) = y$ as $k \to \infty$.*

Proof. By theorem 2.1, $x_{k+1} = x_k + \xi_{k,\alpha_k}$ is a better approximation to the exact solution x^+ than x_k,

$$\|x_{k+1} - x^+\| \leqslant \|x_k - x^+\|.$$

Thus $\|x_k - x^+\|, k \geqslant 1$ is a monotonically decreasing sequence. Next we show the sequence $\{x_k\}$ converges to a solution of $F(x) = y$, where we use a similar technique as in [4] and [11]. In particular, we will show the iteration errors $e_k = x^+ - x_k, k \in N$ form a Cauchy sequence.

Given $k, l \in N$ with $k > l$, let $j \in \{l, \dots, k\}$ be chosen so that

$$\|y - F(x_j)\| \leqslant \|y - F(x_i)\|, \qquad i = l, \dots, k.$$

Now consider

$$\|e_j - e_l\|^2 = \|e_l\|^2 - \|e_j\|^2 + 2(e_j, e_j - e_l), \tag{46}$$

and denote $w_k = (T_k T_k^* + \alpha_k I)^{-1} \operatorname{res}_k$, where $\operatorname{res}_k = y_\delta - F(x_k)$. Then we have

$$\xi_{k,\alpha_k} = T_k^* w_k,$$

$$x_{k+1} = x_k + \xi_{k,\alpha_k} = \cdots = x_0 + \sum_{j=0}^{k} T_j^* w_j$$

and

$$x_j - x_l = \sum_{i=l}^{j-1} T_i^* w_i.$$

Note that

$$|(e_j, e_j - e_l)| = \left| \sum_{i=l}^{j-1} (e_j, T_i^* w_i) \right| \leqslant \sum_{i=l}^{j-1} \|w_i\| \|T_i e_j\|$$

and

$$\begin{aligned}
\|T_i e_j\| &= \|T_i e_i - T_i(x_j - x_i)\| \\
&\leqslant \|y - F(x_i) - F'(x_i)e_i\| + \|F(x_j) - F(x_i) - F'(x_i)(x_j - x_i)\| + \|y - F(x_j)\| \\
&\leqslant d\|y - F(x_i)\| + d\|F(x_j) - F(x_i)\| + \|y - F(x_j)\| \\
&\leqslant 2d\|y - F(x_i)\| + (1 + d)\|y - F(x_j)\| \\
&\leqslant (1 + 3d)\|y - F(x_i)\|,
\end{aligned}$$

where $e_i = x^+ - x_i$. Then combining (35) with the above expression, we obtain

$$\begin{aligned}
|(e_j, e_j - e_l)| &\leqslant (1 + 3d) \sum_{i=l}^{j-1} \|w_i\| \|y - F(x_i)\| \\
&\leqslant \frac{1 + 3d}{2(c_1 - d)} (\|x^+ - x_l\|^2 - \|x^+ - x_j\|^2).
\end{aligned}$$

This, together with (46), yields

$$\|e_j - e_l\|^2 \leqslant C(\|x^+ - x_l\|^2 - \|x^+ - x_j\|^2),$$

where $C = \frac{1+c_1+2d}{c_1-d}$ does not depend on j, k, l. Similarly we have

$$\|e_j - e_k\|^2 \leqslant C(\|x^+ - x_j\|^2 - \|x^+ - x_k\|^2).$$

Hence,

$$\begin{aligned}
\|x_k - x_l\|^2 &\leqslant 2(\|x_k - x_j\|^2 + \|x_j - x_l\|^2) \\
&= 2(\|e_k - e_j\|^2 + \|e_j - e_l\|^2) \\
&\leqslant 2C(\|x^+ - x_l\|^2 - \|x^+ - x_k\|^2).
\end{aligned}$$

Recalling the monotonicity of the iteration error $\|x_k - x^+\|$, the right-hand side approaches zero as $k, l \to \infty$, and hence $\{x_k\}$ is a Cauchy sequence.

Denote the limit of x_k by x. Because

$$\begin{aligned}\|\mathrm{res}_k\| &\leqslant \|T_k T_k^* + \alpha_k I\| \|w_k\| \\ &= (\|T_k\|^2 + \alpha_k)\|w_k\|,\end{aligned}$$

and α_k is uniformly bounded according to lemma 2.2, we have

$$\|w_k\| \geqslant \frac{1}{\|T_k\|^2 + M}\|\mathrm{res}_k\|,$$

where M is the upper bound of α_k. This proves that

$$\|y - F(x_k)\|^2 \leqslant \frac{\|T_k\|^2 + M}{2(c_1 - d)}(\|x^+ - x_k\|^2 - \|x^+ - x_{k+1}\|^2). \tag{47}$$

Recalling that $\|T_k\|$ is uniformly bounded, we obtain the convergence of $\sum_{i=0}^{\infty}\|y - F(x_i)\|^2$ from estimate (47). Therefore $F(x_k) \to y$ as $k \to \infty$, and x is a solution of $F(x) = y$. □

3. Regularity for inexact data

We assume that the rhs y is contaminated by noise, i.e., instead of y, we may then have a perturbed version y_δ with an error level δ such that

$$\|y - y_\delta\| \leqslant \delta.$$

In this case, the solution of the ill-posed problem would be very sensitive to the small perturbations in the rhs y.

For algorithm 1.1, the stopping rule we choose is the discrepancy principle (see [4]), i.e., *the iteration should be terminated at the first occurrence of the index k such that*

$$\left\|F\left(x_k^\delta\right) - y_\delta\right\| \leqslant \tau\delta, \tag{48}$$

where $\tau > 1$ *is the dominant parameter and can be chosen by users.*

Now we consider the regularity of the trust region algorithm in the perturbed case. Accordingly the corresponding iterates will be denoted by x_k^δ. We also assume that $k(\delta)$ is the smallest iteration index k such that the discrepancy inequality

$$\left\|y_\delta - F\left(x_{k(\delta)}^\delta\right)\right\| \leqslant \tau\delta, \qquad \tau > 1 \tag{49}$$

holds.

Theorem 3.1. *Assume that assumption 1.4 holds and* $T_k T_k^* + \alpha_k I$ *is positive definite. Let* τ *in (49) be chosen such that* $\tau > \frac{1+d}{1-d}$. *Let* x^+ *be a solution of* $F(x) = y$ *with* F *satisfying (26) for some* $d > 0$ *in a ball* $\mathcal{B} \subset \mathcal{D}(F)$ *around* x. *Then algorithm 1.1 terminates after* $k(\delta) < \infty$ *iterations. Moreover, for* $k = 0, 1, \ldots, k(\delta)$, $\left\|x^+ - x_k^\delta\right\|$ *is monotonically decreasing.*

Proof. We will prove that

$$\left\|x^+ - x_{k+1}^\delta\right\| \leqslant \left\|x^+ - x_k^\delta\right\| \tag{50}$$

with x^+ a solution of $F(x) = y$. By assumption 1.4, we can estimate that

$$\begin{aligned}\left\|y_\delta - F\left(x_k^\delta\right) - F'\left(x_k^\delta\right)\left(x^+ - x_k^\delta\right)\right\| &\leqslant \delta + \left\|F(x^+) - F\left(x_k^\delta\right) - F'\left(x_k^\delta\right)\left(x^+ - x_k^\delta\right)\right\| \\ &\leqslant \delta + d\left\|y - F\left(x_k^\delta\right)\right\| \\ &\leqslant (1 + d)\delta + d\left\|y_\delta - F\left(x_k^\delta\right)\right\|.\end{aligned}$$

Since $\|y_\delta - F(x_k^\delta)\| > \tau\delta$ as $k < k(\delta)$, hence

$$\delta < \frac{1}{\tau}\|y_\delta - F(x_k^\delta)\|$$

and

$$\|y_\delta - F(x_k^\delta) - F'(x_k^\delta)(x^+ - x_k^\delta)\| \leqslant \frac{1+d+\tau d}{\tau}\|y_\delta - F(x_k^\delta)\|.$$

By assumption, $0 < \frac{1+d+\tau d}{\tau} < 1$, hence (27) is fulfilled with y replaced by y_δ and x_k by x_k^δ. Consequently theorem 2.1 applies and the monotonicity assertion (50) follows as in the proof of theorem 2.1.

Next we show that there is only a finite number of iterations. In fact using the same arguments as in the proof of (47), we have

$$\|y_\delta - F(x_k^\delta)\|^2 \leqslant \frac{L}{2(c_1 - d)}\left(\|x^+ - x_k^\delta\|^2 - \|x^+ - x_{k+1}^\delta\|^2\right) \tag{51}$$

with $L = \sup\{\|F'(x_k^\delta)\| + M\}$ for all $k < k(\delta)$. By (50) and taking the sum of (51) for $k = 0, 1, \ldots, k(\delta) - 1$, we obtain

$$k(\delta)\tau^2\delta^2 \leqslant \sum_{k=0}^{k(\delta)-1}\|y_\delta - F(x_k^\delta)\|^2 \leqslant \frac{L}{2(c_1-d)}\|x^+ - x_0\|^2 < \infty.$$

This indicates that $k(\delta)$ is a finite number. □

Now, we consider the case $x_{k(\delta)}^\delta$ as $\delta \to 0$, and have the following result.

Theorem 3.2. *If $k(\delta) = k$ for all δ sufficiently small, then $x_k^\delta \to x_k$ for $k(\delta) = k$ as $\delta \to 0$.*

Proof. By continuity, if $k(\delta) = k$ for all $\delta > 0$, then $x_k^\delta \to x_k$ as $\delta \to 0$, where x_k is the kth trust region iterate with the exact right-hand side y. Since $k(\delta) = k$, hence by the discrepancy principle, $F(x_k) \to y$ as $\delta \to 0$. Hence, $x_{k(\delta)}^\delta$ converges to the solution x_k of $F(x) = y$. □

With the above preparation, now we can prove that the trust region algorithm with appropriate conditions is a regularization method.

Theorem 3.3. *Assume that F satisfies (26) in some ball $\mathcal{B} \subset \mathcal{D}(F)$ and let y_δ, x_k^δ be defined as before. Then the iterates x_k^δ generated by algorithms 1.1 converge to a solution of equation (1) as $k \to \infty$ and $\delta \to 0$.*

Proof. From theorem 2.3 we know that iterates x_k converge to a solution of $F(x) = y$. Combining this fact with theorem 3.2, we find that iterates x_k^δ converge to a solution of $F(x) = y$ for $k = k(\delta)$ as $k \to \infty$ and $\delta \to 0$.

Now, assume that $k(\delta) \to \infty$ as $\delta \to 0$, and denote by x^+ the limit of the iterates x_k. x^+ is a solution of $F(x) = y$. It suffices to consider subsequences $\{k(\delta_n)\}_n$ which are monotonically increasing to infinity as $n \to \infty$ and $\delta_n \to 0$. For example, consider $k(\delta_m) > k(\delta_n)$ for $m > n$. By the monotonicity of x_k^δ (see (50)), we have

$$\|x_{k(\delta_m)}^{\delta_m} - x^+\| \leqslant \|x_{k(\delta_n)}^{\delta_m} - x^+\| \leqslant \|x_{k(\delta_n)}^{\delta_m} - x_k(\delta_n)\| + \|x_{k(\delta_n)} - x^+\|.$$

Thus given a sufficiently small number $\epsilon > 0$ and for some sufficiently large number n, we have that $\|x_{k(\delta_n)} - x^+\| \leqslant \epsilon/2$ by theorem 2.3. On the other hand, for sufficiently large number m and fixed n, $\|x_{k(\delta_n)}^{\delta_m} - x_k(\delta_n)\| \leqslant \epsilon/2$ by theorem 3.1. This proves that $\|x_{k(\delta_m)}^{\delta_m} - x^+\| \leqslant \epsilon$ for all m sufficiently large, and thereafter $x_{k(\delta_m)}^{\delta_m} \to x^+$ as $m \to \infty$ and hence $x_k^\delta \to x^+$ as $k \to \infty$ and $\delta \to 0$. □

4. Numerical implementation

To solve (9) numerically, the problem has to be discretized. Let P_n denote a projection of X onto an n-dimensional subspace X_n, and Q_m denote a projection of Y onto an m-dimensional subspace Y_m. Hence we can define F_{mn} the finite approximation to the nonlinear operator F:

$$F_{mn}(x) := Q_m F(P_n x). \tag{52}$$

Since F is differentiable, each F_{mn} is differentiable. For the discrete version (52), we consider the minimal least squares problem in the form

$$\min_{x\in X_n} J_n(x) := \tfrac{1}{2}\|F_{mn}(x) - y_m\|^2. \tag{53}$$

In this case, at each iteration a trial step is calculated by solving the subproblem

$$\min_{\xi\in X_n} \tilde{\psi}_k(\xi) := (\mathrm{grad}(J_n)_k, \xi) + \tfrac{1}{2}(\widetilde{\mathrm{Hess}}(J_n)_k\xi, \xi), \tag{54}$$

$$\text{s.t. } \|\xi\| \leqslant \Delta_k, \tag{55}$$

in finite spaces X_n and Y_m. Where, $\mathrm{grad}(J_n)(x)$ and $\widetilde{\mathrm{Hess}}(J_n)(x)$ can be evaluated respectively by

$$\mathrm{grad}(J_n)(x) = F'_{mn}(x)^T(F_{mn}(x) - y_m)$$

and

$$\widetilde{\mathrm{Hess}}(J_n)(x) = F'_{mn}(x)^T F'_{mn}(x).$$

For simplicity, we use the same notation T_k for $F'_{mn}(x_k)$ and g_k for $-F'_{mn}(x_k)^T(y_m - F_{mn}(x_k))$. Noted in section 2, the parameter α_k can be determined from equation (30) or (31). For solving the parameter α_k, we apply Newton's method to the nonlinear equation

$$\Gamma_k(\alpha_k) := \frac{1}{\|\xi_{k,\alpha_k}\|} - \frac{1}{\Delta_k} = 0. \tag{56}$$

The reason for considering (56) instead of the simpler equation

$$\|\xi_{k,\alpha_k}\| = \Delta_k \tag{57}$$

is that $\Gamma_k(\alpha_k)$ is close to a linear function. Thus Newton's method would give a faster convergence. In fact the first-order and second-order derivatives of $\Gamma(\alpha_k)$ can be easily computed; hence Newton's method can be used to calculate α_k (see [17, 18] for details). The iteration formula can be given as

$$\alpha_+ = \alpha_k - \frac{\|\xi_{k,\alpha_k}\|^3}{g_k^T\left(T_k^T T_k + \alpha_k I\right)^{-3} g_k}\left[\frac{1}{\|\xi_{k,\alpha_k}\|} - \frac{1}{\Delta_k}\right]. \tag{58}$$

The following algorithm (see [9]) updates α_k by Newton's method applied to (56).

Algorithm 4.1. *(Newton's method for computing α_k)*

Until $\|\alpha_+ - \alpha_k\| \leqslant tol$ do

STEP 1 Factor $T_k^T T_k + \alpha_k I = R^T R$.

STEP 2 Solve $R^T R\xi_{k,\alpha_k} = -g_k$.

STEP 3 Solve $R^T w = \xi_{k,\alpha_k}$.

STEP 4 Let $\alpha_+ := \alpha_k - \left(\frac{\|\xi_{k,\alpha_k}\|}{\|w\|}\right)^2\left(1 - \frac{\|\xi_{k,\alpha_k}\|}{\|\Delta_k\|}\right)$.

In this algorithm, $R^T R$ is the Cholesky factorization of the matrix $T_k^T T_k + \alpha_k I$ with $R \in R^{n\times n}$ upper triangular. It is necessary to safeguard α_k in order to obtain a positive definite $T_k^T T_k + \alpha_k I$ and guarantee convergence. This, in practice, can be satisfied by observing the fact (see [10]) that the function $\Gamma_k(\alpha_k)$ is concave and strictly increasing. Hence if we choose the initial guess value $\alpha_k > 0$ such that $\Gamma_k(\alpha_k) < 0$ then at each iteration, the Newton algorithm generates a monotonically increasing sequence converging to the solution of $\Gamma_k(\alpha_k) = 0$. The tol for choosing the final value of α_k should be in (0, 1). But for fast convergence, tol should not be chosen too small. We choose $tol = 0.001$ in our test of the next section.

We also remark that for α_k solved by algorithm 4.1, $T_k^T T_k + \alpha_k I$ is positive definite at each iteration. Hence the assumption that $T_k^* T_k + \alpha_k I$ is positive definite in sections 2 and 3 is reasonable.

Remark 4.2. The finite-dimensional problem (53) can also be solved with any standard nonlinear optimization package, such as those available on NEOS (www-neos.mcs.anl.gov/neos/). To obtain satisfactory results, we need to use a regularization technique, such as the regularized Gauss–Newton method (see [1]). This is because the derivative operator $F'_{mn}(x)$ is a compact operator, hence the finite-dimensional problem is ill-posed. For ill-posed problems, nonlinear optimization methods cannot be used directly due to the noise and truncation error propagation.

Remark 4.3. Applying trust algorithms to nonlinear ill-posed problems, we actually need to implement two cycles of iterations: outer loop and inner loop. The outer loop consists of updating the iterate x_k and the trust region radius Δ_k and evaluating the gradient $\mathrm{grad}(J_n)_k$ and Hessian $\widetilde{\mathrm{Hess}}(J_n)_k$, the inner loop is seeking a trial step ξ_k and a regularization parameter α_k by solving the TRS (54)–(55). The main computational cost of the trust region algorithm is the solution of the TRS. In algorithm 4.1, we perform the Cholesky factorization on a symmetric matrix, which requires the amount of computation $O\left(\frac{1}{6}n^3\right)$. Solving for ξ_{k,α_k} and w, the amount of computation is $O(3n^2)$. So the computational cost in each inner loop is $O\left(\frac{1}{6}n^3 + 3n^2\right)$. Assume that there are k_{in} inner iterations, then the total computational cost in the inner loop is $O\left(\frac{1}{6}k_{in}n^3 + 3k_{in}n^2\right)$. Note that for the inner loop, the gradient $\mathrm{grad}(J_n)_k$ and Hessian $\widetilde{\mathrm{Hess}}(J_n)_k$ are used only once, which are pre-computed in the outer loop. It requires matrix–matrix computation $O(n^3)$ to obtain $\widetilde{\mathrm{Hess}}(J_n)_k$ and matrix–vector computation $O(n^2)$ to obtain $\mathrm{grad}(J_n)_k$. In theorem 3.1, we have proved that the trust region algorithm terminates after $k(\delta) < \infty$ iterations. So the total computational cost in the outer loop is $O(k(\delta)n^3 + k(\delta)n^2)$. Therefore, upon convergence, the total computational cost (inner loop plus outer loop) is $O\left(\left(\frac{1}{6}k_{in} + k(\delta)\right)n^3 + (3k_{in} + k(\delta))n^2\right)$. This computational effort is acceptable in modern computers.

5. Numerical results

We take a well-known example that appears in inverse gravimetry ([13]), which is in the form of the nonlinear Fredholm integral equation of the first kind

$$F(x)(t) = \int_a^b k(t, s, x(s))\,\mathrm{d}s = y(t), \qquad t \in [e, f], \tag{59}$$

with the kernel $k(t, s, x(s)) = \ln \frac{(t-s)^2+H^2}{(t-s)^2+(x(s)-H)^2}$, and $y(t)$ is the measured term. This numerical example is also considered in [14]. Clearly the kernel k is defined on the set

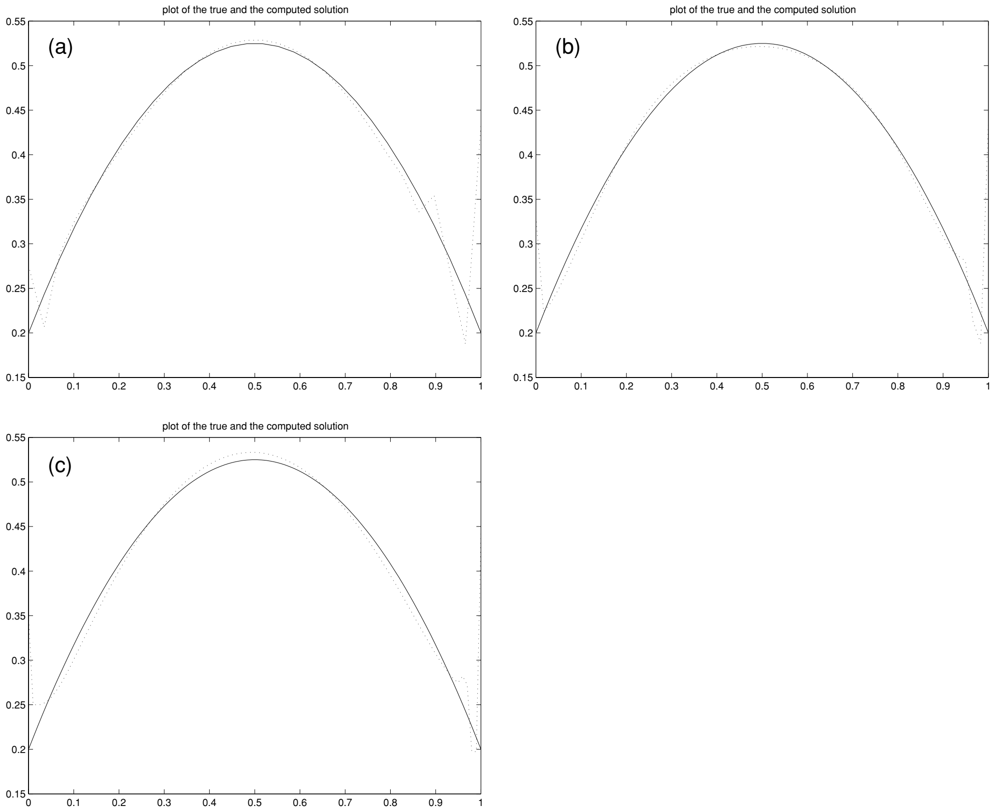

Figure 1. Solutions of the inverse gravimetry problem. The solid line represents the true solution, x_{true}; the dashed line the approximate solution, x_{appr}.

$\Pi = \{[e, f] \times [a, b] \times R\}$ and $k(t, s, x(s)) \in C^1(\Pi)$. The first derivative $F'(x) : X \longrightarrow Y$ is given by

$$[F'(x)u](t) = \int_a^b \frac{\partial k}{\partial x}(t, s, x(s))u(s)\,\mathrm{d}s, \qquad t \in [e, f], \tag{60}$$

while the kernel $\frac{\partial k}{\partial x}(t, s, x(s))$ is evaluated by

$$\frac{\partial k}{\partial x}(t, s, x(s)) = \frac{2(H - x(s))}{(t - s)^2 + (x(s) - H)^2}.$$

To solve (59) numerically, we choose linearly independent basis functions $\{\phi_j\}_{j=1}^n \subset X = H_0^1(a, b)$ and take approximations $\hat{x}(s) = \sum_{j=1}^n x_j \phi_j(s)$, $x = (x_1, x_2, \dots, x_n)^T \in R^n$ and then reduce it to a finite-dimensional problem. The integral operator F gives rise to an operator $F_{mn} : R^n \longrightarrow R^m$,

$$[F_{mn}(x)]_i = \int_a^b k(t_i, s, \hat{x}(s))\,\mathrm{d}s, \qquad 1 \leqslant i \leqslant m. \tag{61}$$

Similarly the derivative operator $F'(x)$ yields an $m \times n$ matrix:

$$[F'_{mn}(x)]_{ij} = \int_a^b \frac{\partial k}{\partial x}(t_i, s, \hat{x}(s))\phi_j(s)\,\mathrm{d}s, \qquad 1 \leqslant i \leqslant m, \qquad 1 \leqslant j \leqslant n, \tag{62}$$

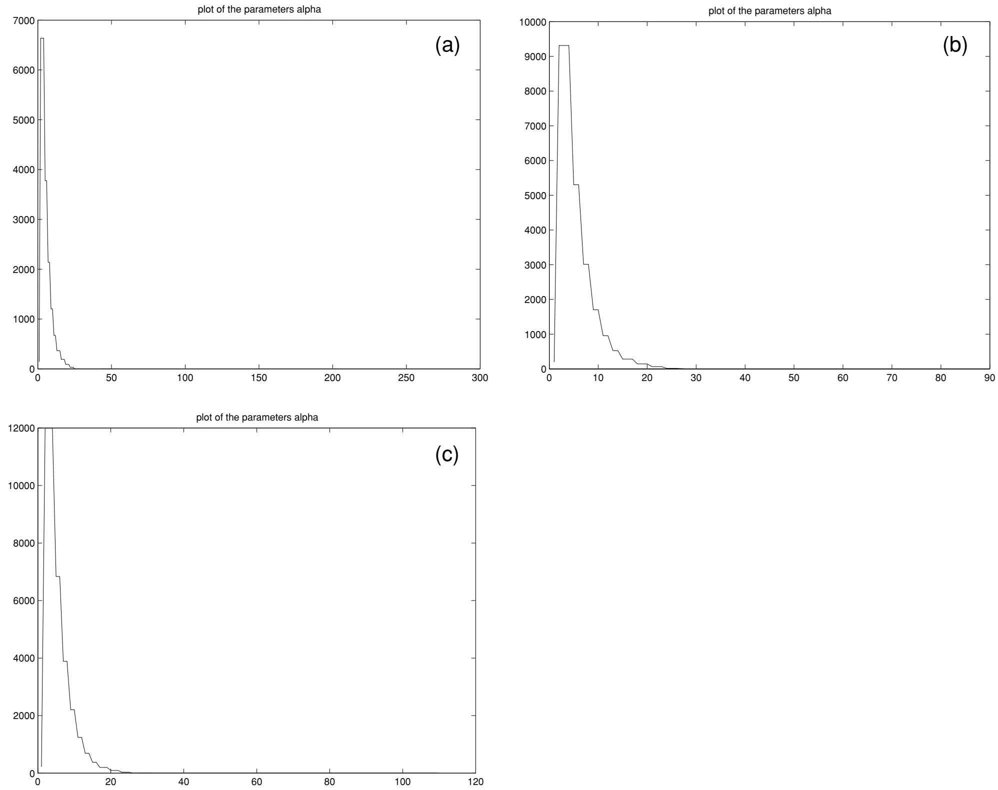

Figure 2. Parameters α generated by algorithm 4.1.

where $\phi_j(s)$ are the standard linear basic functions

$$\phi_j(s) = \begin{cases} \dfrac{s - s_{j-1}}{h}, & \text{if} \quad s \in [s_{j-1}, s_j], \\ \dfrac{s_{j+1} - s}{h}, & \text{if} \quad s \in [s_j, s_{j+1}], \\ 0, & \text{else}, \end{cases}$$

and $s_j = jh$, $h = \frac{1}{n}$, $j = 1, 2, \ldots, n$. Then integral (59) can be computed numerically.

We take $[a, b] = [e, f] = [0, 1]$, $H = 0.1$ and different m, n to give a discretization. Our true function is $x_{\text{true}}(s) = 1.3s(1 - s) + 0.2$, and it is discretized by evaluating it at the points s_i to give the components x_i of x. The right-hand side y is generated by integral (59). To simulate the practical problems, the perturbed right-hand side is chosen by

$$y_\delta = y_{\text{true}} + \delta * \text{rand}(\text{size}(y_{\text{true}})) \tag{63}$$

where rand$(\cdot)$ is the Gaussian white noise having the same dimension as that of y_{true}. The numerical results are shown in figures 1–3.

In section 3, in the process of proving the monotonicity of the iteration error $\|x^+ - x_k^\delta\|$, we let $\tau > \frac{1+d}{1-d}$. Practically, $\tau > 1$ can be chosen by users. In theory, in order to make the approximation better, τ should be chosen close to 1 (see [4]). This means d should be close to 0.

First we choose $n = m = 30$, $\tau = 1.4$, $\Delta_0 = 0.001$, $\alpha_0 = 0.01$ with a small perturbation $\delta = 0.005$. It needs 83 outer iterations and 300 inner iterations to generate convergence.

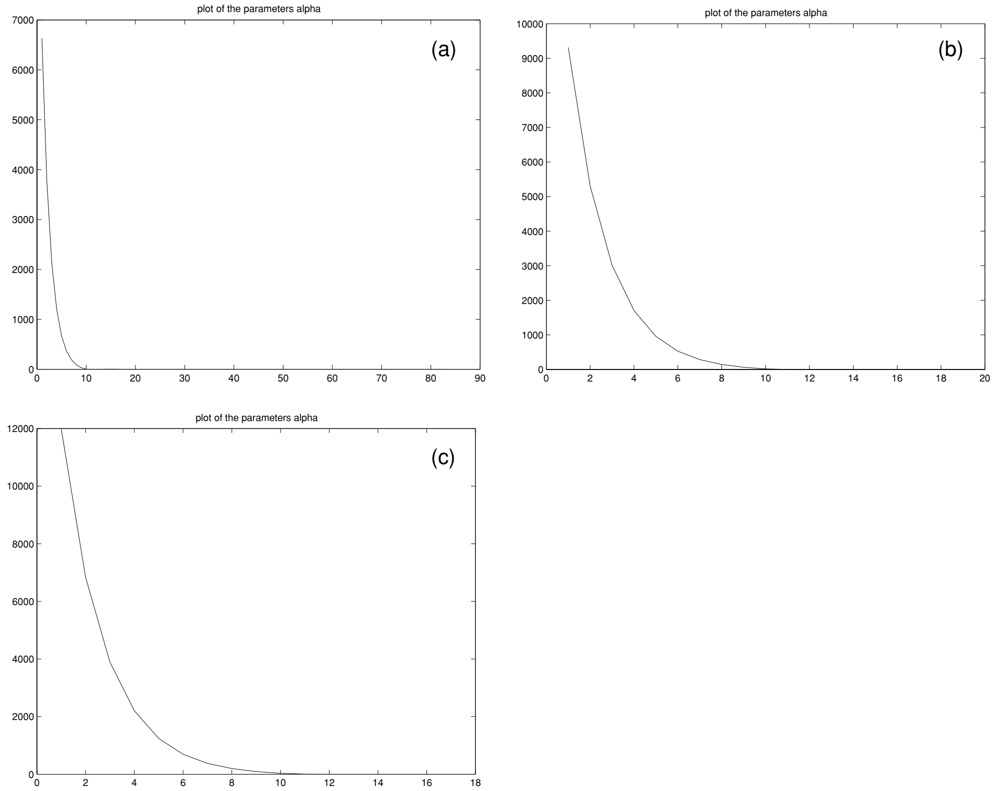

Figure 3. Last values of parameters α in each iteration.

The discrepancy norm $\|y_\delta - F(x_{k=83})\|$ is 0.0070 and the relative error level $\frac{\|y_\delta - F(x_{k=83})\|}{\|F(x_{k=83})\|}$ is 0.0014. The recorded total CPU time is 1.25 s. The solution is shown in figure 1(a), where, the solid line represents the true solution, the dashed line represents the approximations. This interpretation will also be used for the other two graphs in figure 1.

Then we choose $n = m = 60$, $\tau = 1.3$, $\Delta_0 = 0.001$, $\alpha_0 = 0.01$ but with a perturbation $\delta = 0.01$. It needs 20 outer iterations and 90 inner iterations to generate convergence. The discrepancy norm $\|y_\delta - F(x_{k=20})\|$ is 0.0112 and the relative error level $\frac{\|y_\delta - F(x_{k=20})\|}{\|F(x_{k=20})\|}$ is 0.0015. The recorded CPU time is 1.109 s. The graph of the solution is shown in figure 1(b).

Finally we choose $n = m = 100$, $\Delta_0 = 0.001$, $\alpha_0 = 0.01$ with a large perturbation $\delta = 0.05$ and $\tau = 1.2$ to give a computation. It needs 18 outer iterations and 110 inner iterations to generate convergence. The discrepancy norm $\|y_\delta - F(x_{k=18})\|$ is 0.0540 and the relative error level $\frac{\|y_\delta - F(x_{k=18})\|}{\|F(x_{k=18})\|}$ is 0.0057. The recorded CPU time is 2.828 s. The graph of the solution is shown in figure 1(c).

From figure 1, we see that the computed results are satisfactory. The degree of approximation depends on the mesh-grid density and the noise level δ. It is true that the discretized problem will be a better approximation to the original problem if the discretization points are enormous, but it will be more ill-conditioned. Therefore, due to the ill-posedness of the problem, we cannot expect the computation results to be totally the same.

In our test, we also record the values of α_k during each iteration. We find that the values of α_k are increasing in each loop of algorithm 4.1, but in general it is decreasing. This is true for our algorithms. The phenomena are shown in figures 2 and 3, in which, from (a) to (c), each of the figures corresponds to the cases of figure 1. In figure 2, all of the values of α_k in

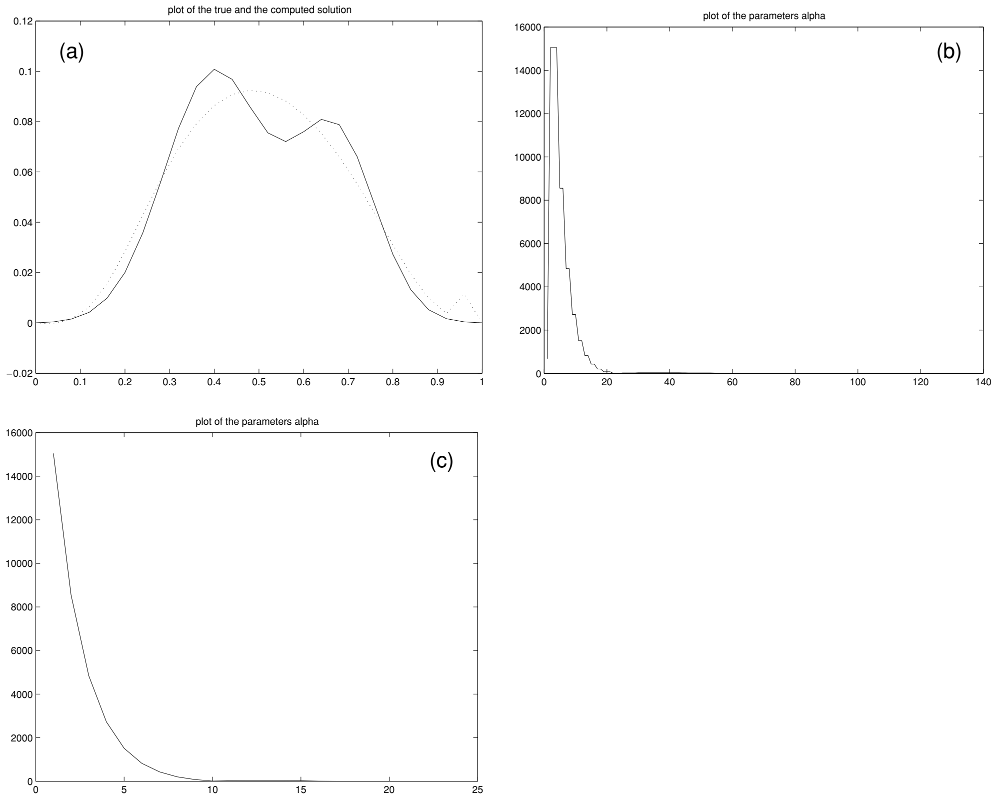

Figure 4. Solutions of the inverse gravimetry problem. (a) The solid line represents the negative true solution, $-x_{\text{true}}$; the dashed line represents the approximate solution, $-x_{\text{appr}}$. (b) Parameters α generated by algorithm 4.1. (c) Last values of parameters α in each iteration.

each iteration are recorded; in figure 3, only the last values of α_k in each iteration are recorded. These values of α_k can suppress the small singular values of $F'_{mn}(x_k)$ during each iteration; hence they also serve as the regularization parameters.

In [14], the author chose another true solution to give a numerical simulation. The true function is based on the linear combination of two Gaussians

$$x_{\text{true}}(s) = c_1 \exp(d_1(s - p_1)^2) + c_2 \exp(d_2(s - p_2)^2) + c_3 s + c_4,$$

where $c_1 = -0.1, c_2 = -0.075, d_1 = -40, d_2 = -60, p_1 = 0.4, p_2 = 0.67$, and c_3, c_4 are chosen so that $x(0) = x(1) = 0$. In the tests, the interval $[a, b] = [e, f] = [0, 1]$, $H = 0.2$ and the mesh grid numbers are 26. We give a plot of the simulation results in figure 4 by our algorithms for the given parameters $\Delta_0 = 0.001, \alpha_0 = 0.01$ with a perturbation $\delta = 0.005$ and $\tau = 1.01$. It needs 24 outer iterations and 135 inner iterations to generate convergence. The discrepancy norm $\|y_\delta - F(x_{k=24})\|$ is 0.0042 and the relative error level $\frac{\|y_\delta - F(x_{k=24})\|}{\|F(x_{k=24})\|}$ is 0.0040. The recorded CPU time is 0.359 s. The behaviour of the regularization parameters α is similar to that in the former test. It seems that our algorithms generate comparable results.

We should point out that in both tests, we add another stopping rule, i.e., if Pred $\leqslant \epsilon$ (ϵ is the tolerance) then the iteration should also be terminated. Here, the tolerance ϵ is chosen as $1.0e - 7$. However, Pred $\leqslant \epsilon$ is never activated. In fact, if we use Pred $\leqslant \epsilon$ as the stopping rule, it will need many iterations. In such case, the reconstruction results are not very reasonable. Perhaps choosing a smaller ϵ will work better.

6. Conclusion

We establish the convergence and regularity of the trust region method for nonlinear ill-posed inverse problems. The results obtained in this paper are based on the choice of the approximate Hessian $\widetilde{\text{Hess}}(x) = F'(x)^T F'(x)$ instead of the exact Hessian $\text{Hess}(x) = F'(x)^T F'(x) + F''(x)^T (F(x) - y)$. For the latter, we have not gotten such kinds of results so far. We conjecture that the trust region algorithm with the exact Hessian is also a regularization algorithm provided that some stronger conditions are imposed. However, it is unreasonable to use the exact Hessian in practice due to the difficulty in computing the exact Hessian $\text{Hess}(x)$ and the huge expense on it. We can use some other techniques, for instance, in each iteration, $F''(x_k)$ is replaced by B_k, where B_k satisfies the quasi-Newton conditions (see [3]), but the regularity of such an algorithm is not so clear right now.

Acknowledgments

We thank Professor Zuhair Nashed and Professor Qiyu Sun for their suggestion on improving the paper and fruitful discussion. We also wish to thank the referees for their constructive comments and kindly pointing out [14], which may be an original reference for using trust region methods for nonlinear ill-posed inverse problems. The work is partially supported by SRF for ROCS, SEM and also partially supported by Chinese NSF grant 10231060.

References

[1] Bakushinsky A and Goncharsky A 1994 *Ill-posed Problems: Theory and Applications* (Dordrecht: Kluwer)
[2] Binder A, Hanke M and Scherzer O 1996 On the Landweber iteration for nonlinear ill-posed problems *J. Inverse Ill-posed Problems* **4** 381–9
[3] Dennis J E, Gay D M and Welsch R E 1981 An adaptive nonlinear least-squares algorithm *ACM Trans. Math. Softw.* **7** 369–83
[4] Engl H W, Hanke M and Neubauer A 1996 *Regularization of Inverse Problems* (Dordrecht: Kluwer)
[5] Fletcher R 1987 *Practical Methods of Optimisation* 2nd edn (Chichester: Wiley)
[6] Groetsch C W 1984 *The Theory of Tikhonov Regularization for Fredholm Equations of The First Kind* (Boston, MA: Pitman)
[7] Hanke M, Neubauer A and Scherzer O 1995 A convergence analysis of Landweber iteration for nonlinear ill-posed problems *Numer. Math.* **72** 21–37
[8] Hanke M 1997 A regularizing Levenberg–Marquardt scheme, with applications to inverse groundwater filtration problems *Inverse Problems* **13** 79–95
[9] Moré J J and Sorensen D C 1983 Computing a trust region step *SIAM J. Sci. Stat. Comput.* **4** 553–72
[10] Nocedal J and Yuan Y 1998 Combining trust region and line search techniques *Advances in Nonlinear Programming* ed Y Yuan (Dordrecht: Kluwer) pp 153–75
[11] Scherzer O 1996 A convergence analysis of a method of steepest descent and a two-step algorithm for nonlinear ill-posed problems *Numer. Funct. Anal. Optim.* **17** 197–214
[12] Sorensen D C 1982 Newton's method with a model trust region modification *SIAM J. Numer. Anal.* **19** 409–26
[13] Tikhonov A N and Arsenin V Y 1977 *Solutions of Ill-Posed Problems* (New York: Wiley)
[14] Vogel C R 1990 A constrained least-squares regularization method for nonlinear ill-posed problems *SIAM J. Control Optim.* **28** 34–49
[15] Wang Y and Yuan Y 2003 A trust region method for solving distributed parameter identification problems *J. Comput. Math.* **21** 759–72
[16] Xiao T, Yu S and Wang Y 2003 *Numerical Methods for Inverse Problems* (Beijing: Science Press) (in Chinese)
[17] Yuan Y 1994 Nonlinear programming: trust region algorithms *Proc. Chinese SIAM Ann. Meeting* ed S T Xiao and F Wu (Beijing: Tsinghua University Press) pp 83–97
[18] Yuan Y 1998 Matrix computation problems in trust region algorithms for optimization *Proc. 5th CSIAM Ann. Meeting* ed Q C Zeng, T Q Li, Z S Xue and Q S Cheng (Beijing: Tsinghua University Press) pp 52–64
[19] Yuan Y and Sun W 1997 *Optimization Theories and Methods* (Beijing: Science Press) (in Chinese)