A Trust Region Method for Solving Distributed Parameter Identification Problems [*]

Yan-fei Wang   Ya-xiang Yuan
State Key Laboratory of Scientific and Engineering Computing,
Institute of Computational Mathematics and Scientific/Engineering Computing,
Academy of Mathematics and System Sciences, CAS
Beijing, 100080, P.R.China
Email: wyf@lsec.cc.ac.cn yyx@lsec.cc.ac.cn

November 7, 2001

Abstract. This paper is concerned with the ill-posed problem of identifying a parameter in an elliptic equation. Its solution is obtained by applying trust region method, which exhibits attractive theoretical convergence properites and seems promising. Numerical experiment is given to illustrate the efficiency of the proposed method.

Key Words. parameter identification, ill-posed problems, trust region.
AMS Subject Classifications: 65K10, 35R30, 35R25, 65J15, 65J20

1 Introduction

Parameter identification problems play an important role in many applications in science and industry (see [1, 3]). By parameter identification, we refers that the estimation of coefficients in a differential equation from observations of the solution to that equation. We call the coefficients the system parameters, and the solution and its derivatives the state variables. The forward problem is to compute the state variables given the system parameters and appropriate boundary conditions, which is a well-posed problems. However in parameter identification, the problems is typically ill-posed (see [5]).

[*] Partially supported by Chinese NSF grant 19731010 and the Knowledge Innovation Program of CAS

For example, we consider the problem of identifying a distributed parameter $q = q(x)$ in the one-dimentional steady-state diffusion equation in the form

$$-\nabla(q\nabla u) = g, \text{ in } (0,1) \tag{1}$$

with Dirichlet boundary conditions

$$u(0) = u_0, \ u(1) = u_1.$$

This is used to model for example, the steady-state temperature distribution within a thin metal rod (see [12]). Another example is the inverse groundwater filtration problem of reconstructing the diffusitivity q of a seidment from measurements of the piezometric head u in the steady state case (see [1] for further applications). We take the former case as our example. In this setting, the state variable is the temperature distribution $u(x), \ x \in (0,1)$, the system parameters are diffusion coefficient $q(x)$ and the heat source term $g(x)$. The inverse problems stated here is determining parameter $q(x)$ by giving $g(x)$ and $u(x)$ for $x \in [0,1]$.

For sake of simplifying the notations, we outline the problem in the abstract operator form

$$F(q)u = g, \tag{2}$$

where $F(q)$ represents a parameter-dependent differential operator from the parameter space Q to the state space U, $q \in Q$ represents the distributed parameter to be estimated, and $u \in U$ represents the correspoinding state variable. In case of the above example, q represents the diffusion coefficient, and

$$F(q) = -\nabla(q\nabla(\cdot)).$$

Since u is the observation data, therefore, it may contain noise. Assume that the observed data can be expressed as

$$u_e = u + e \tag{3}$$

with Gaussian noise e.

Because of the ill-posedness of the problem (1), some kind of regularization technique has to be applied (see [5, 13, 24]). Perhaps Tikhonov regularization method (see [9, 20]) is the most well-known method for dealing with such kind of problems.

Given the regularization parameter $\alpha > 0$, choose $q^\alpha \in Q$ to solve the unconstrained minimization problem

$$\min_{q\in Q} M^\alpha[q] := \|F(q)u_e - g\|^2 + \alpha\|q\|^2, \tag{4}$$

where $\alpha > 0$ is called the regularization parameter and $\|q\|^2$ serves as the stabilizer.

Assume the forward problem solving for u is well-posed, then we can denote the solution by

$$f(q) := u = F^{-1}(q)g. \tag{5}$$

Clearly we want to minimize the following constrained functional

$$J_{q\in Q}(q) = \frac{1}{2}\|u - u_e\|^2, \tag{6}$$

$$s.\ t.\ F(q)u = g. \tag{7}$$

By (5), problem (6)-(7) is equivalent to the unconstrained regularized least squares minimization problem

$$\min J_{q\in Q}(q) = \frac{1}{2}\|f(q) - u_e\|^2. \tag{8}$$

Certainly we can use the Tikhonov regularization to (5), for which, we have the following minimization problem:

$$\min J_{q\in Q}(q) = \frac{1}{2}\|f(q) - u_e\|^2 + \alpha\theta(q), \tag{9}$$

where $\theta(q)$ is a regularized fuctional whose duty is to impose stability, $\alpha > 0$ is a regularization parameter.

This paper will deal with the problem in a different way: i.e., we use some kind of approximation to the original problem (8), then the trust region technique is used.

2 Finite Dimensional Approximation: Trust Region Method

First we introduce the trust region method in a general way. Trust region methods are a group of methods for ensuring global convergence while retaining fast local convergence in optimization algorithms. For example, we consider the minimization problem

$$\min_{x\in\mathcal{R}^n} f(x). \tag{10}$$

In trust region methods, we first choose a trial step length Δ, and then use the quadratic model to select the best step of (at most) this length for the quadratic model by solving

$$\min \psi(x_c + \xi) = f(x_c) + (g(x_c), \xi) + \frac{1}{2}(H_c\xi, \xi), \tag{11}$$

$$s.\ t.\ \|\xi\| \leq \Delta_c. \tag{12}$$

The trial step length Δ_c is considered an estimate of how far we trust the quadratic model, hence it is called a trust radius and the resultant method is called a trust region method.

In this section, we will consider the approximation minimization problem (8) by utilizing the trust region technique mentioned above.

As a rule, the numerical solution of the fundamental problem would be impossible without the use of computers. In general, the traditional way of implementation is to perform a finite-dimentional approximation of the problem under considerations.

Let P_n denote a projection of Q onto an n-dimentional subspace Q_n, i.e., $P_n : Q \longrightarrow Q_n$ and $\bar{P}_n : Q_n \longrightarrow Q$, which possess a number of remarkable properties:

(1) the operators P_n and $\bar{P}_n$ are continuous for all n;

(2) $P_n\bar{P}_n = I_n$, where I_n is the identity operator on the space X_n;

(3) $\bar{P}_nP_nq \in D(F)$ for any $q \in D(F)$ and all positive integers n.

Similarly, let R_m denote a projection of U onto an m-dimentional subspace U_m, i.e., $R_m : U \longrightarrow U_m$ and $\bar{R}_m : U_m \longrightarrow U$. Now we can define F_{mn} the finite approximation to the nonlinear operator F:

$$F_{mn}(q)u := R_mF(P_nq)u. \tag{12}$$

Now in finite dimensional case, the minimization problem is in the form

$$\min J_{q\in Q_n}(q) = \frac{1}{2}\|u - u_e\|^2, \tag{13}$$

$$s.\ t.\ F_{mn}(q)u = g. \tag{14}$$

Furthermore, if we denote $f_{mn}(q) = F_{mn}^{-1}(q)g$, (13)-(14) can be transformed into the following unconstrained minimization problem

$$\min J_{q\in Q_n}(q) = \frac{1}{2}\|f_{mn}(q) - u_e\|^2. \tag{15}$$

Since F is differentiable, each F_{mn} is differentiable. Let us denote $J_{q\in Q_n}(q)$ by J_n or $J_n(q)$, the gradient of the functional $J_{q\in Q_n}(q)$ by $grad(J_n)$, the approximate Hessian of the functional $J_{q\in Q_n}(q)$ by $Hess(J_n)$. At each iteration, a trial step is calculated by solving the subproblem

$$\min_{s\in Q_n} \phi_k(s) := grad(J_n)_k^Ts + \frac{1}{2}(Hess(J_n)_ks, s), \tag{16}$$

$$s.\ t.\ I(s) \leq \Delta_k, \tag{17}$$

in finite spaces Q_n and U_m. For simplicity, we assume that $m \equiv n$. In the above expression, $I(s)$ denotes some kind of modular. For example, we can take $I(s)$ as $\frac{1}{2}\|s\|^2_{L^2}$. Here, for our purpose, we take $I(s)$ as $\frac{1}{2}\|Ls\|^2$, where L denotes some kind of differential operator, which is bounded, self-adjoint positive definite. In (16)-(17), $grad(J_n)_k$ is the gradient at the current approximate solution, $Hess(J_n)_k$ is an $n \times n$ symmetric matrix which approximates the Hessian of $J_{q\in Q_n}(q)$ and $\Delta_k > 0$ is a trust region radius. Let s_k be a solution of (16)-(17). The predicted reduction is defined by the reduction in the approximate model, i.e.,

$$Pred_k = \phi_k(0) - \phi_k(s_k) = -\phi_k(s_k). \tag{18}$$

Unless the current point q_k is a stationary point and $Hess(J_n)_k$ is positive semi-definite, the predicted reduction is always positive. The actual reduction is the reduction in the objective function

$$Ared_k = J_{q\in Q_n}(q_k) - J_{q\in Q_n}(q_k + s_k). \tag{19}$$

And we define the ratio between the actual and the predicted reduction by

$$r_k = \frac{Ared_k}{Pred_k} \tag{20}$$

which is used to decide whether the trial step is acceptable and to adjust the new trust region radius.

With the above analysis, we generate the trust region algorithm for solving parameter identification problem as follows.

Algorithm 2.1 *(Trust region algorithm for parameter identification problem)*

STEP 1 *Given the initial guess value $q_1 \in \mathcal{R}^n$, $\Delta_1 > 0$, $0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 \leq \tau_0 \leq \tau_2 < 1$, $\tau_2 > 0$, $k := 1$;*

STEP 2 *If* **the stopping rule is satisfied** *then STOP; Else, solve (16)-(17) giving s_k;*

STEP 3 *Compute r_k;*

$$q_{k+1} = \begin{cases} q_k & if \quad r_k \leq \tau_0, \\ q_k + s_k & otherwise; \end{cases} \tag{21}$$

Choose Δ_{k+1} that satisfies

$$\Delta_{k+1} = \begin{cases} [\tau_3\|s_k\|, \tau_4\Delta_k] & if \quad r_k < \tau_2, \\ [\Delta_k, \tau_1\Delta_k] & otherwise; \end{cases} \tag{22}$$

STEP 4 Evaluate $grad(J_n)_k$ *and* $Hess(J_n)_k$*; k:=k+1; GOTO STEP 2.*

The constant τ_i $(i=0,\cdots,4)$ can be chosen by users. Typical values are $\tau_0=0,\ \tau_1=2,\ \tau_2=\tau_3=0.25,\ \tau_4=0.5$. For other choices of those constants, please see [6], [7], [14], [17], etc.. The parameter τ_0 is usually zero (see [6], [18]) or a small positive constant (see [4] and [19]). The advantage of using zero τ_0 is that a trial step is accepted whenever the objective function is reduced. Hence it would not throw away a "good point", which is a desirable property especially when the function evaluations are very expensive (see [21]).

In STEP 2, the stopping rule is based on the so-called discrepancy principle, which will be stated in the next section.

There are several ways to evaluate the gradient of the leat squares cost functional, say, finite differences method, adjoint or costate methods (see [1], [3], [23]). For finite differences method, for example, assuming a discretization of the parameter of the form

$$q=\sum_{i=1}^{n}c_i\psi_i,$$

can be obtained, then the gradients can be approximated by

$$(grad(J_n(q)))_i\approx\frac{J_n(q+h_i\psi_i)-J_n(q)}{h_i},\ i=1,2,\cdots,n, \tag{23}$$

where h_i is a relatively small scalar compared to the i^{th} component of the discretized parameter q. For distributed parameter identification, finite difference gradient computations are expensive. From equations (13)-(14) we know each gradient evaluation reqiures n evaluations of $f_{mn}(q)=F_{mn}^{-1}(q)g$, and each computation of $F_{mn}^{-1}(q)$ entails the approximate solution of a differential equation. When n is large, gradient approximations based directly on (23) are extremely expensive, requiring $n+1$ evaluations of the functional $J_n(q)$, and hence $n+1$ solutions of the equation (2).

Adjoint or costate methods for parameter identification were introduced by Chavant and Lemonier (see [3]). These kinds of methods can remarkably reduce the cost of gradient evaluation. Consider the least squares problem:

$$J_n(q)=\frac{1}{2}\|f_{mn}(q)-u_e\|^2. \tag{24}$$

Letting $res(q)=f_{mn}(q)-u_e$ denote the residual and using the fact that $\frac{d}{dh}F_{mn}(q+he_i)|_{h=0}=\frac{dF_{mn}}{dq}e_i$, we obtain a representation for the components of the gradient of $J_n(q)$. For $i = 1, 2, \cdots, n$,

$$\begin{aligned}
(grad(J_n(q)))_i &:= \frac{d}{dh}J_n(q + he_i)|_{h=0} \\
&= (\frac{d}{dh}f_{mn}(q + he_i)|_{h=0}, res(q)) \\
&= -(F^{-1}(q)\frac{dF_{mn}}{dq}e_i F_{mn}^{-1}(q)g, res(q)) \\
&= (\frac{dF_{mn}}{dq}e_i u, v),
\end{aligned}$$

where u solves the state equation (14) and v solves the adjoint or costate equation

$$F_{mn}^{*}(q)v = -res(q).$$

In the above expression, $F_{mn}^{\star}$ stands for the adjoint of the operator F_{mn}.

Compared with the finite difference computation (23), the costate gradient computation requires only one inversion of the operator $F_{mn}(q)$, together with one inversion of its adjoint.

Now we turn to Hessian computations. We use Gauss-Newtion method to approximate the exact Hessian of $J_{q\in Q_n}(q)$. For ease of notation, we simply denote $Hessian$ the exact Hessian of $J_{q\in Q_n}(q)$. In context of the least squares functional (24), its Hessian can be expressed as

$$Hessian = Hess(J_n(q)) + \frac{d^2 f_{mn}}{dq^2}res(q), \tag{25}$$

where

$$Hess(J_n(q)) = (\frac{df_{mn}}{dq})^{\star}(\frac{df_{mn}}{dq}).$$

$Hess(J_n(q))$ is the so-called Gass-Newton approximation to the Hessian. This evaluation has some computational advantages. First, it can sometimes be much easier to compute than the full Hessian, since it does not involve the second derivative term $\frac{d^2 f_{mn}}{dq^2}$, which has a tensor representation. Moreover, from Proposition 2.4 in the following context, we can conclude that the scaled trust region step is a strict decent direction if the first derivative $\frac{df_{mn}}{dq}$ has full rank. Even if $\frac{df_{mn}}{dq}$ does not have full rank, we can adjust the Lagrangian parameter α, such that the trust region step is a decent direction.

Take $Q_n = U_m = \mathcal{R}^n$, then for subproblem (16)-(17), we have the following lemma.

Lemma 2.2 *A vector $s^* \in \mathcal{R}^n$ is a solution of the problem*

$$\min_{s \in \mathcal{R}_n} \phi(s) := grad(J_n)^T s + \frac{1}{2}(Hess(J_n)s, s), \tag{26}$$

$$s.\ t.\ I(s) \le \Delta, \tag{27}$$

where $I(s) = \frac{1}{2}\|Ls\|^2$, L is a bounded self-adjoint positive semi-definite linear operator, $grad(J_n) \in \mathcal{R}^n$, $Hess(J_n) \in \mathcal{R}^{n \times n}$ is a symmetric matrix, and $\Delta > 0$, if and only if there exists $\alpha^* \ge 0$ such that

$$(Hess(J_n) + \alpha^* L^* L)s^* = -grad(J_n) \tag{28}$$

and that $Hess(J_n) + \alpha^* L^* L$ is positive semi-definite, $I(s^*) \le \Delta$ and

$$\alpha^*(\Delta - I(s^*)) = 0. \tag{29}$$

Proof. Equations (28)-(29) are just the KKT conditions of the optimization problem (26)-(27). It remains to show $Hess(J_n) + \alpha^* L^* L$ is positive semi-definite. Assume that $s^* \ne 0$. Since s^* solves (26)-(27), it also solves $\min\{\phi(s) : I(s) = I(s^*)\}$. It follows that $\phi(s) \ge \phi(s^*)$ for all s such that $I(s) = I(s^*)$. This inequality together with (28) gives

$$-s^T(Hess(J_n) + \alpha^* L^* L)s^* + \frac{1}{2}(Hess(J_n)s, s) \ge$$

$$-s^{*T}(Hess(J_n) + \alpha^* L^* L)s^* + \frac{1}{2}(Hess(J_n)s^*, s^*). \tag{30}$$

Rearranging terms in (30) gives

$$\frac{1}{2}(s^* - s)^T(Hess(J_n) + \alpha^* L^* L)(s^* - s) \ge \frac{1}{2}\alpha(\|Ls\|^2 - \|Ls^*\|^2) = 0 \tag{31}$$

for all s such that $I(s) = I(s^*)$. Since $s \ne 0$, it follows from (31) that $Hess(J_n) + \alpha^* L^* L$ is positive semi-definite. If $s^* = 0$, it follows from (28) that $grad(J_n) = 0$. Therefore $s^* = 0$ solves $\min\{\frac{1}{2}(Hess(J_n)s, s) : I(s) \le \Delta\}$ and we must conclude that $Hess(J_n)$ is positive semi-definite. Since $\alpha^* \ge 0$ is necessary, and hence $Hess(J_n) + \alpha^* L^* L$ is positive semi-definite. Q.E.D

Lemma 2.2 establishes necessary conditions concerning the pair α^*, s^* when s^* solves (26)-(27). Our next result establishes sufficient conditions that will ensure s is a solution to (26)-(27).

Lemma 2.3 *Let $\alpha^* \in \mathcal{R}$, $s^* \in \mathcal{R}^n$ satisfy*

$$(Hess(J_n) + \alpha^* L^* L)s^* = -grad(J_n) \tag{32}$$

with $Hess(J_n) + \alpha^* L^* L$ is positive semi-definite. Then we have the following results:

(1) If $\alpha^* = 0$ and $I(s^*) \leq \Delta$ then s^* solves (26)-(27);

(2) If $I(s^*) = \Delta$ then s^* solves

$$\phi(s^*) = \min\{\phi(s) : I(s) = \Delta\};$$

(3) If $\alpha^* \geq 0$ and $I(s^*) = \Delta$ then s^* solves (26)-(27). Furthermore, if $Hess(J_n) + \alpha^* L^* L$ is positive definite then s^* is unique in each of cases (1), (2) and (3).

Proof. If s^*, α^* satisfy (32) then

$$grad(J_n)^T s + \frac{1}{2} s^T (Hess(J_n) + \alpha^* L^* L)s \geq$$

$$grad(J_n)^T s^* + \frac{1}{2} s^{*T} (Hess(J_n) + \alpha^* L^* L)s^* \tag{33}$$

holds for any $s \in \mathcal{R}^n$. It follows that

$$\phi(s) \geq \phi(s^*) + \frac{1}{2}\alpha^*(\|Ls^*\|^2 - \|Ls\|^2). \tag{34}$$

Hence statements (1), (2) and (3) can be directly obtained from (34). The uniqueness, follows from the fact that $Hess(J_n) + \alpha^* L^* L$ is positive definite. Q.E.D

From lemmas 2.2 and 2.3, we know that if $Hess(J_n) + \alpha^* L^* L$ is positive definite, s^* is uniquely defined by

$$s^* = -(Hess(J_n) + \alpha^* L^* L)^{-1} grad(J_n). \tag{35}$$

To emphasize the fact that s is dependet on the parameter α, we write

$$s_\alpha = -(Hess(J_n) + \alpha L^* L)^{-1} grad(J_n), \tag{36}$$

which has the following property:

Proposition 2.4 *Assume that L is bounded self-adjoint and positive definte, $Hess(J_n) + \alpha L^* L$ is positive definite. Then for $L = I$ (the identity operator), the norm of the search direction s_α is strictly decreasing as α increases from zero; For $L \neq I$, the norm of the scaled search direction $D^{\frac{1}{2}} s_\alpha$ is strictly decreasing as α increases from zero, where $D = L^* L$.*

Proof. First, we prove the result for $L = I$. It is easy to show

$$\frac{d}{d\alpha}\|s_\alpha\| = \frac{(s_\alpha, \frac{ds_\alpha}{d\alpha})}{\|s_\alpha\|}.$$

Differentiating the equation (36) with α for $L = I$, we have

$$(Hess(J_n) + \alpha I)\frac{ds_\alpha}{d\alpha} = -s_\alpha.$$

Hence

$$\begin{aligned}\frac{d}{d\alpha}s_\alpha &= -(Hess(J_n) + \alpha I)^{-1}s_\alpha \\ &= (Hess(J_n) + \alpha I)^{-2}grad(J_n)\end{aligned}$$

and

$$\frac{d}{d\alpha}\|s_\alpha\| = -\frac{grad(J_n)^T(Hess(J_n) + \alpha L^*L)^{-3}grad(J_n)}{\|s_\alpha\|}.$$

Since $Hess(J_n) + \alpha I$ is positive definite according to the assumption, the above relation implies that $\|s_\alpha\|$ is strictly decreasing as α increase from zero. The first assertion follows.

Next we prove the result for $L \neq I$. Noticing that $(Hess(J_n) + \alpha L^*L)^{-1}$ can be rewritten as

$$(Hess(J_n) + \alpha D)^{-1} = D^{-\frac{1}{2}}(D^{-\frac{1}{2}}Hess(J_n)D^{-\frac{1}{2}} + \alpha I)^{-1}D^{-\frac{1}{2}},$$

and if we denote $D^{-\frac{1}{2}}grad(J_n) = g_n$, then the search direction s_α can be written as

$$s_\alpha = D^{-\frac{1}{2}}(D^{-\frac{1}{2}}Hess(J_n)D^{-\frac{1}{2}} + \alpha I)^{-1}g_n.$$

Hence

$$D^{\frac{1}{2}}s_\alpha = (D^{-\frac{1}{2}}Hess(J_n)D^{-\frac{1}{2}} + \alpha I)^{-1}g_n.$$

Note that $D^{-\frac{1}{2}}Hess(J_n)D^{-\frac{1}{2}} + \alpha I$ is positive definite, the result follows for $L \neq I$ by using the same technique for the proof of $L = I$. Q.E.D

Proposition 2.4 is important while implementing trust region algorithm. No matter how large α is, the norms $\|s_\alpha\|$ for $L = I$ or $\|D^{\frac{1}{2}}s_\alpha\|$ for $L \neq I$ are strictly decreasing as α increases from zero. The proposition also tells us that at least the upper bound of $\|s_\alpha\|$ is decreasing. Hence, the search direction can not go everywhere.

The following theorem will show the monotonicity of the objective functional.

Theorem 2.5 *Assume that L and $Hess(J_n(q_k)) + \alpha L^*L$ are positive definite, q_k is the current iteration point which does not satisfy the necessary conditon of Lemma 2.2 and $grad(J_n(q_k)) \neq 0$. Then for s a solution of (26)-(27) and Δ_k sufficently small, we have $J_n(q_k + s) < J_n(q_k)$.*

Proof. Since $J_n(q)$ is twice continuously differentiable, it follows that

$$J_n(q_k + s) = J_n(q_k) + grad(J_n(q_k))^T s + O(\|s\|^2).$$

Hence it suffices to show that there exists a constant $C > 0$ for which

$$grad(J_n(q_k))^T s \leq -C\|s\|. \tag{37}$$

Note that the solution s can be expressed as

$$s = -(Hess(J_n(q_k)) + \alpha L^* L)^{-1} grad(J_n),$$

the result is clear. Q.E.D

From Theorem 2.5 we know that the trust region algorithm gives decrease in the objective functional outside the region of convergence (i.e., we can not trust the "trust region") and the trust region constraint $\|s\| \leq \Delta$ is active. Once the iterates are inside the region of convergence, we take the Gauss-Newton step and the trust region constraint $\|s\| \leq \Delta$ becomes inactive.

3 Choosing Regularization Parameter and the Stopping Criterion

In inverse and ill-posed problems, Lagrangian parameter, also known as regularization parameter (see equations (4) and (9)) plays an important role in quantifying the tradeoff between error amplification due to instability and truncation due to regularization. There are two kinds of methods: one is a *a-priori* one, the other is *a-posteriori* one. An *a-priori* esimation of the Lagrangian parameter is easily performed compared to an *a-posteriori* one, but an *a-posteriori* esimation of the Lagrangian parameter is more feasible in practice. For such kind of method, please see [2, 5, 11, 8, 10, 16] etc.. Here, we will choose the regularization parameter in a different way. According to Algorithm 2.1, the trust region constraint is inactive if the current iterate is inside the trust region. In such case we accept the iterate without solving the trust region subproblem. Once the current iterate is outside the trust region, the trust region constraint is active, we have to solve the trust region subproblem. Hence our new way of choosing regularization parameter is based on the relation between the current iterate s_k and the trust region Δ_k. We will give an detail analysis in the following paragraph.

The regularization parameter stated here refers that when we solve the subproblem (26)-(27), the Lagrangian parameter is added. Note that solving for a solution s^* of the subproblem (26)-(27) is equivalent to solve the equations (28)-(29), hence the Lagragian parameter α has to be determined in each iteration. In the present section, we will use trust region technique to determine the parameter, which means this technique relies on the trust region radius.

We will assume that the state variable u is condaminated with error, i.e., instead of u, we may have a perturbed version u_e with error level δ such that

$$\|u - u_e\| \leq \delta.$$

In such case, the solution of the problem may be very sensitive to the small perturbations in the state variable u.

Now we introduce the trust region technique to determine the regularization parameter.

Lemma 1 indicates that if s_k is a solution of (26) and (27), then there is a unique $\alpha_k \geq 0$ that satisfies (28) and (29). From equations (35) and (36) we know that the parameter $\alpha_k > 0$ and satisfies

$$\|s_{\alpha_k,k}\| = \Delta_k, \tag{38}$$

i.e.

$$\|(Hess(J_n(q_k)) + \alpha_k L^*L)^{-1} grad(J_n(q_k))\| = \Delta_k. \tag{39}$$

Thus, similar to techniques for subproblems of trust region algorithms for unconstrained optimization (see [21], [22]), we can apply Newton's method to the nonlinear equation

$$\Gamma_k(\alpha_k) := \frac{1}{\|s_{\alpha_k,k}\|} - \frac{1}{\Delta_k}. \tag{40}$$

The reason for considering (40) instead of the simpler equation

$$\|s_{\alpha_k,k}\| - \Delta_k = 0 \tag{41}$$

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 $\Gamma_k(\alpha_k)$ can be easily computed, hence Newton's method can be used to calculate α^*, the solution of $\Gamma_k(\alpha_k) = 0$. For simplicity, we denote $A = Hess(J_n(q_k))$ and

$b = grad(J_n(q_k))$. Applying Newton's method to (40), we can compute the iteration sequence $\{\alpha_k\}$ by the follwing formula

$$\alpha_+ = \alpha_k - \frac{\|s^2_{\alpha_k,k}\|}{b^T(A+\alpha_k L^*L)^{-2}L^*L(A+\alpha_k L^*L)^{-1}b}[1 - \frac{\|s_{\alpha_k,k}\|}{\Delta}] \tag{42}$$

with α^+ the next iterate.

The following algorithm updates α_k by Newton's method applied to (40).

Algorithm 3.1 *(Newton's method for computing α)*

```
Until convergence do
STEP 1  Factor $A + \alpha_k L^*L = R^TR$;
STEP 2  Solve $R^TRs_{\alpha_k,k} = -b$;
STEP 3  Solve $R^TRw = s_{\alpha_k,k}$;
STEP 4  Let $\alpha_k := \alpha_k - \frac{\|s_{\alpha_k,k}\|^2}{\|w^TL^*Ls_{\alpha_k,k}\|}(1 - \frac{\|s_{\alpha_k,k}\|}{\|\Delta_k\|})$.
```

In this algorithm, R^TR is the Cholesky factorization of matrix $A + \alpha_k L^*L$ with $R \in \mathcal{R}^{n\times n}$ upper triagular. It is necessary to safeguard α_k in order to obtain a positive definte $A + \alpha L^*L$ and guarantee convergence. This in practice can be satisfied by observing the fact that the function $\Gamma_k(\alpha_k)$ is concave and strictly increasing, hence if we choose the initial guess value $\hat{\alpha} > 0$ such that $\Gamma_k(\hat{\alpha}) < 0$ then at each iteration, Newton algorithm generates a monotonically increasing sequence converging to the solution of $\Gamma_k(\alpha_k) = 0$.

We should also point out that, Algorithm 3.1 can still be implemented even if L is semi-definite as long as $A + \alpha_k L^*L$ is positive definte. With the above analysis this in fact is feasible.

For the present version of trust region iteration the discrepacy principle is an appropriate stopping criterion for this purpose. Assume that

$$\|u_e - f(q_{true})\| \leq \delta$$

and to emphasize the dependency on δ we let $\{q_k^\delta\}$ denote the iterates if u_e instead of u is used in the iteration. According to the discrepancy principle the iteration is terminated at the first occurence of the index $k = k_D$ such that

$$\|u_e - f(q_{k_D}^\delta)\| \leq \tau\delta \tag{46}$$

with $\tau > 1$ being another parameter.

This stopping rule for the trust region method is well-defined since according to Theorem 4, $\|u_e - f(q_k^\delta)\|$ is monotonically decreasing in k.

4 Numerical Test

We give a numerical example to test the efficiency of the proposed trust region method. Our example is based on the steady-state diffusion equation given in section 1.

The interval is chosen to be $[0,1]$, the boundary conditions u in (1) as

$$u_0 = u_1 \equiv 0,$$

the heat soure term $g(x)$ as

$$g(x) = 1$$

and the starting parameter q_1 as

$$q_1 \equiv 1.$$

We define the exact solution as $q_{true} = 1 + 0.75e^{-50(x-0.25)^2}$, then generate u according to equation (1).

We apply standard piecewise linear finite element discretization with nodes $x_i = ih$, $h = 1/(n+1)$, $n = 50$. Mid-point quadrature is used to evaluate the finite element stiffness matrix, and the discrete system is formulated as follows:

$$F(q)u = g.$$

To simulate the observation data, we add Gaussian noise to the exact value u as

$$u_e = u + \delta rand(u).$$

Our problem is to estimate q given the observation data u_e. We consider the approximation problem to least squares functional $J(q) = ||F^{-1}(q)g - u_e||^2$:

$$\phi(s) := grad(J)^T s + \frac{1}{2}(Hess(J)s, s), \tag{47}$$

$$s.\ t.\ I(s) \le \Delta, \tag{48}$$

where $I(s) = \frac{1}{2}||Ls||^2$, L is the discrete one-dimensional Laplacian. Trust region Algorithm 2.1 is implemented to solve the above problem. The results are shown in figure 1–figure 4.

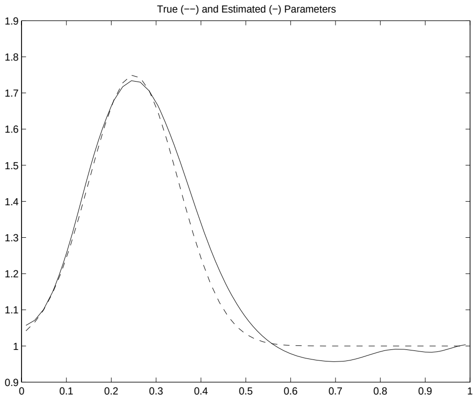

Figure 1

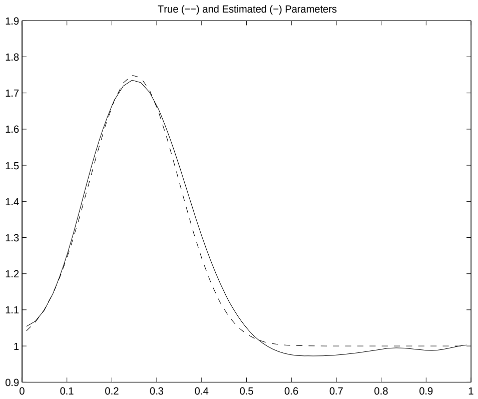

Figure 2

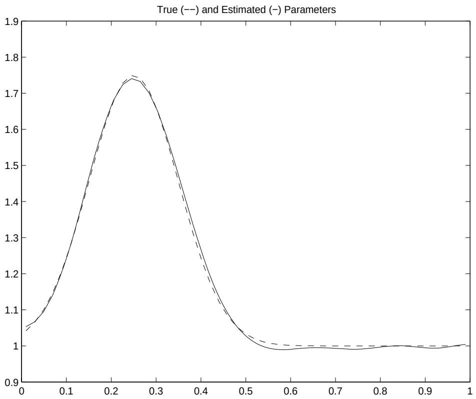

Figure 3

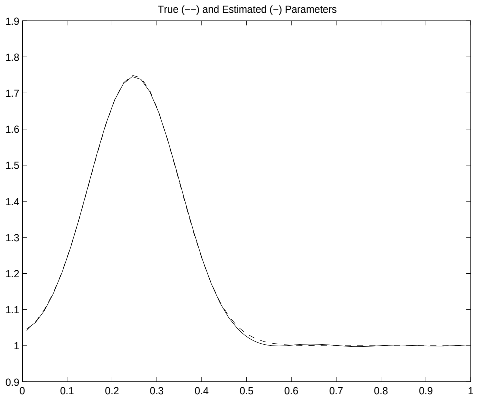

Figure 4

In all the figures, we choose the initial trust region radius as $\Delta_1 = 1.0$, the initial guess value of α as 0.1, the dominant parameter τ as 1.01. The error levels we used and the iterations are displayed in table 1. Note that the choice the initial guess value α is not so crucial. We have tested on other choices of α values, say $\alpha = 5$, 10 or 100, and got the same results.

We observe from figure 3 and figure 4 that if the error level is small, then the approximated solution can approximate the exact solution perfectly well.

Table 1 The error levels and the iterations

	error level (δ)	iterations
figure 1	$\delta = 2\%$	15
figure 2	$\delta = 1\%$	16
figure 3	$\delta = 0.5\%$	18
figure 4	$\delta = 0.1\%$	22

We also perform Tikhonov regularization (9) to solve the diffusion equation.

The stabilizer $\theta(q)$ is chosen as $\frac{1}{2}\|Lq\|^2$, L is the discrete one-dimensional Laplacian. We use quasi-Newton iteration to minimize Tikhonov functional

$$\frac{1}{2}\|F^{-1}(q)g - u_e\|^2 + \alpha\theta(q)$$

and the computational results are shown in figure 5–figure 8. The error levels we used and the iterations are displayed in table 2. We use an *a-priori* estimation of the regularization parameter α. In all of the figures 5–8, we choose the regularization parameter α as 0.1, the dominant parameter τ as 2.0. Note that this choice of the regularization parameter is crucial. It can not be too large or too small. If α is too large, its solution may be far from the noise-free solution since the new problem is a poor approximation to the original problem; if α is too small, the influence of the data errors may cause instabilities. We can see this phenomena from figures 6, 9 and figure 10. We add the same noise level $\delta = 0.01$ in these figures. In figure 9 we choose $\alpha = 5.0$, in figure 10 we choose $\alpha = 1.0 \times 10^{-4}$. If α is larger than 5.0 or smaller than 1.0×10^{-4}, the results will be more worse.

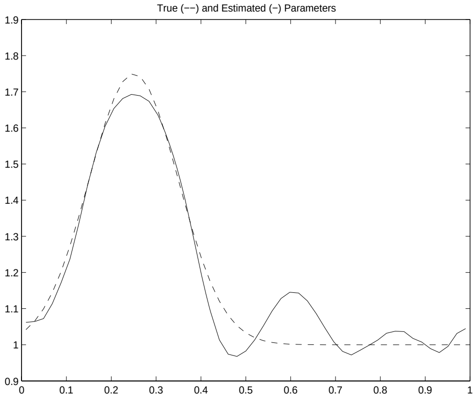

Figure 5

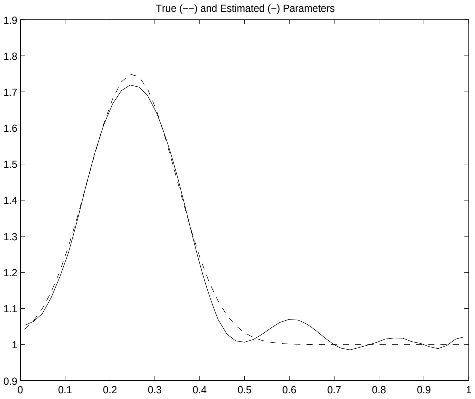

Figure 6

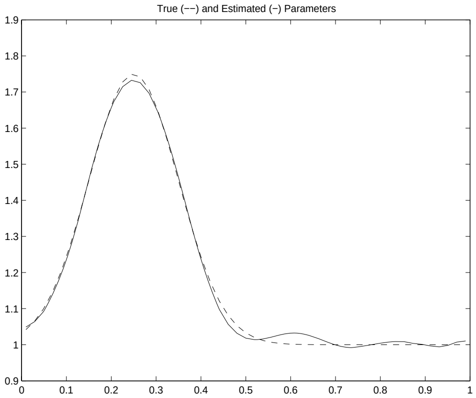

Figure 7

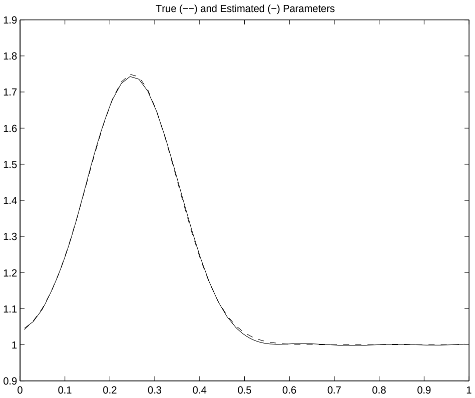

Figure 8

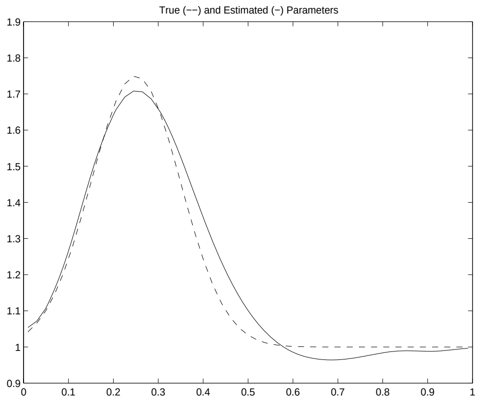

Figure 9

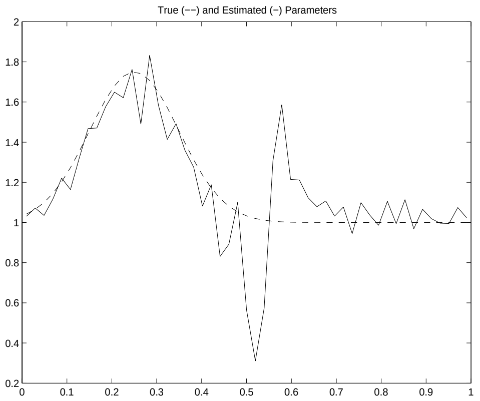

Figure 10

Table 2 The error levels and the iterations

	error level (δ)	iterations
figure 5	$\delta = 2\%$	2
figure 6	$\delta = 1\%$	2
figure 7	$\delta = 0.5\%$	3
figure 8	$\delta = 0.1\%$	3
figure 9	$\delta = 1\%$	2
figure 10	$\delta = 1\%$	3

From figure 1–figure 8, we observed that the behavior of the trust region method and the Tikhonov regularization method is very similar. They are both stable methods. We also observed from figure 3–figure 4 and figure 7–figure 8 that if the error level is small, the solution obtained by trust region method can approximate the exact solution as well as by Tikhonov regularization method.

5 Conclusion and Future work

The numerical experiment illustrates that the trust region method is stable for solving ill-posed problems, at least for distributed parameter identification problem concerned in this paper. We do not claim that the trust region algorithm is better than Tikhonov regularization, which has been developed for about 40 years starting from the basic works by Tikhonov. But at least it can give a comparative results. We may conclude that the trust region technique is suitable for regularizing ill-posed problems. But how to prove the regularity of the trust region algorithm remains an interesting topic, we will give a further research later.

Acknowledgment. The authors would like to thank the referee's comments on Proposition 2.4.

References

[1] Banks T and Kunisch K 1989, *Estimation Techniques for Distributed Parameter Systems*, Birkhäuser.

[2] Björck A and Eldén L 1979, Methods in Numerical Algebra for Ill-posed Problems, Technical Report LiTH-MAT-R-1979-33, Department of Mathematics, Linköping University.

[3] Chavent G and Lemonnier P 1974, Identification de la Non-Linearite' D'Une Equation Parabolique Quasi-lineare, *Applied Math. and Opt.* **26**, 121-162.

[4] Duff I S, Nocedal J and Reid J K 1987, The Use of Linear Programming for the Solution of Sparse Sets of Nonlinear Equations, *SIAM J. Sci. Stat. Comput. 8*, 99-108.

[5] Engl H W, Hanke M and Neubauer 1996, *Regularization of Inverse Problems*, Dordrecht: Kluwer.

[6] Fletcher R 1987, *Practical Methods of Optimization* (second edition), John Wiley and Sons, Chichester.

[7] Fletcher R 1982, A Model Algorithm for Composite NDO Problem, *Math. Prog. Study* **17**, 67-76.

[8] Golub G, Heath M and Wahba G 1979, Generalized Cross-Validation as a method for Choosing a Good Ridge Parameter, *Technometrics* **21**, 215-223.

[9] Groetsch C W 1984 *The Theory of Tikhonov Regularization for Fredholm Equations of The First Kind* (Boston,MA:Pitman)

[10] Hansen P C 1992, Analysis of Discrete Ill-Posed Problems by means of the *L*-curve, *SIAM Review* **34**, 561-580.

[11] Kunisch K 1993, On a Class of Damped Morozov Principle, *Computing* **50**, 185-198.

[12] Lin C C and Segel L A 1988, *Mathematics Applied to Deterministic Problems in the Natural Sciences*, SIAM.

[13] Louis A K 1992, *Inverse und Schlecht Gestellte Problems*, Teubner, Stuttgart.

[14] Moré J J 1983, Recent Developments in Algorithms and Software for Trust Region Methods, in: Bachem A, Grötschel and Korte B, eds., *Mathematical Programming: The State of the Art*, Springer-Verlag, Berlin, 258-287.

[15] Moré J J and Sorensen D C 1983, Computing a Trust Region Step, *SIAM J.Sci.Stat.Comput.* **4**, 553-572.

[16] Morozov V A 1984 *Methods for Solving Incorrectly Posed Problems* (New York: Springer)

[17] Powell M J D 1984, Nonconvex Minimization Calculations and the Conjugate Gradient Method, in: Griffiths, ed., *Numerical Analysis*, Lecture Notes in Mathematics 1066, Springer-Verlag, Berlin, 122-141.

[18] Powell M J D 1975, Convergence Properties of a Class of Minimization Algorithms, in: Mangasarian O L, Meyer R R and Robinson S M, eds., *Nonlinear Programming* **2**, Academic Press, Berlin, 1-27.

[19] Sorensen D C 1982, Newton's Method with a Model Trust Region Modification, *SIAM J.Numer.Anal.* **19**, 409-426.

[20] Tikhonov A N and Arsenin V Y 1977, *Solutions of Ill-Posed Problems*, New York: Wiley.

[21] Ya-xiang Yuan 1994, Nonlinear Programming: Trust Region Algorithms, in: S.T. Xiao and F. Wu, eds., Proceedings of Chinese SIAM annual meeting (Tsinghua University, Beijing) pp. 83 –97.

[22] Ya-xiang Yuan 1998, Matrix Computation Problems in Trust Region Algorithms for Optimization, in: Q.C. Zeng, T.Q. Li, Z.S. Xue and Q.S. Cheng, eds. Proceedings of the 5th CSIAM annual meeting, (Tsinghua Unviersity Press, Beijing) pp. 52–64.

[23] Vogel C R and Wade J G 1995, Analysis of costate discretizations in parameter estimation for linear evolution equations, *SIAM Journal on Control and Optimization* **33**, 227-254.

[24] Vogel C R 1987, An Overview of Numerical Methods for Nonlinear Ill-posed Problems, in: H W Engl and C W Groetsch, eds: *Inverse and Ill-posed Problems*, Orlando, FL: Academic, 231-245.