

**Varying Iteration Accuracy
Using Inexact Conjugate Gradients
in Control Problems governed by PDE's**

Xiuhong Du, Eldad Haber,
Maria Karampatakis and Daniel B. Szyld

Report 08-06-27
June 2008

This report is available in the World Wide Web at
<http://www.math.temple.edu/~szyld>

**VARYING ITERATION ACCURACY
USING INEXACT CONJUGATE GRADIENTS
IN CONTROL PROBLEMS GOVERNED BY PDE'S***

XIUHONG DU[†], ELDAD HABER[‡], MARIA KARAMPATAKI[‡], AND DANIEL B. SZYLD[†]

Abstract. This paper considers the solution of certain large scale optimization problems governed by parabolic partial differential equations. A quadratic functional containing a data misfit term is minimized to approximately recover the parameter function. The resulting constrained optimization problem is solved by using the reduced Hessian approach. The conjugate gradient method is employed for the solution of the system involving matrix-vector multiplications which are nontrivial. These matrix-vector products do not need to be computed exactly. In this paper we develop a new computable criterion to establish the allowable reduction of exactness in the matrix-vector product. We show its general application and in particular to the problem at hand. Numerical experiments show that the new computable criteria is effective while other criteria normally used are not as efficient.

1. Introduction. There is an important class of problems where it is required to recover a parameter function based on the solution of partial differential equations (PDEs) in space and time variables. Several such applications arise in many fields including electromagnetic inversion, diffraction tomography, optimal design and control, see, e.g., [3], [4], [8], [17]. Several computational challenges arise due to the size of the problem.

Consider a model problem where we are interested in recovering a model (or control) function $\mathbf{m} = \mathbf{m}(x)$ based on observations of a field (or state) $\mathbf{u} = \mathbf{u}(x, T)$ for some terminal time T , where \mathbf{u} is related to \mathbf{m} by a forward problem

$$\Delta \mathbf{u} = \mathbf{u}_t, \quad x \in \Omega \tag{1.1a}$$

$$\mathbf{u} = \mathbf{m}, \quad x \in \partial\Omega \tag{1.1b}$$

$$\mathbf{u} = \mathbf{u}_0, \quad x \in \Omega / \partial\Omega, \quad \text{for } t = 0 \tag{1.1c}$$

where $\Omega \subset \mathbb{R}^3$.

The forward problem is to find \mathbf{u} given \mathbf{m} , while the inverse problem is to recover \mathbf{m} given observations on the field \mathbf{u} . However, as it is well-known, the inverse problem is not well-posed. There are many possible controls \mathbf{m} which yield a field \mathbf{u} close to the observations. Thus the problem becomes: from all possible control solutions find the one closest to prior information. This leads to an optimization problem where equations (1.1) appear as constraints. A typical formulation of this optimization problem would be to minimize a data misfit term and a regularization term, subject to the forward problem being satisfied.

In order to solve problem (1.1) we follow the discretize-optimize approach. That is, we first discretize the optimization problem and then solve a discrete optimization problem. One of the reasons that leads us to this approach is that the forward problem is parabolic. In such cases most numerical discretization methods are differentiable and therefore the discretize-optimize approach is natural.

*This version dated 27 June 2008

[†]Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, Pennsylvania 19122-6094, USA (dxhdxh@temple.edu, szyld@temple.edu). This research is supported in part by the U.S. Department of Energy under grant DE-FG02-05ER25672.

[‡]Department of Mathematics and Computer Science, Emory University, Atlanta, GA, 30322, USA (haber@mathcs.emory.edu, mkaramp@emory.edu).

Our goal is to solve the inverse problem, however the forward problem needs to be solved first. Upon discretization in space and time, we obtain a very large and sparse system where we assume that the coefficient matrix is invertible. That is, there is a unique solution to the forward problem. Due to the size of the problem it is unreasonable to store or to explicitly invert this matrix, however as we shall see later, there is no need for that.

There are several ways to solve such an optimization problem; see, e.g., [1], [2], [7], [9]. One of them is the all-at-one approach, where the forward and inverse problems are solved simultaneously. Introducing the Lagrangian, we obtain a KKT system. In this paper we solve the KKT system using the reduced Hessian method [12]. The advantage of the reduced Hessian method is that it leads to a symmetric positive definite system and thus could be solved using the conjugate gradient method (CG). However, the main disadvantage of the reduced Hessian method is that each matrix-vector product is highly expensive. Each CG iteration involves the solution of the forward and the adjoint problem. That means that we need to solve two (discretized) PDEs per CG step, and since the problem under consideration is large, an iterative method has to be employed for the solution of these two PDEs. Thus an iterative solver is embedded within an outer one (that is, the one used for the solution of the reduced Hessian system). The cost, of course, of these inner computations may be high. However if the calculations are performed inexactly there may be significant savings in computational effort. In other words, relaxing the accuracy of these inner matrix-vector products would decrease the cost of the overall calculations and thus make the reduced Hessian method attractive. The natural question that arises then is how inexact these inner matrix-vector multiplications are allowed to be performed in order to ensure the convergence of the outer iterations. In the context of nonlinear optimization it is common practice to solve the linear equations at each step of a Newton method inaccurately as long as we are far from the solution, but as we get closer, we need to increase the accuracy, if we wish to achieve quadratic convergence [10]. But in the context of linear systems, such as the one with reduced Hessian, it was shown in [16] that it is actually beneficial to perform the calculations in an increasingly inexact way as the iteration progresses; see also [5], [18].

Our goal in this paper is to give a new computable criteria to determine the inexactness of each matrix-vector product especially for the problem described. We compare several approaches of determining an appropriate way to perform the calculations and to apply them to a nontrivial PDE constrained optimization problem. We show how we can balance the inner and outer tolerance in the reduced Hessian method. One approach allows the matrix-vector multiplications to be performed in an increasingly exact way (as for inexact Newton), while another approach performs the calculations in an increasingly inexact way as the iteration progresses (as suggested in [16]). It will be obvious from our numerical experiments that a substantial saving in computational effort is gained due to the latter approach. We also test having a fix tolerance for the inexact matrix-vector product.

The rest of the paper is organized as follows: In the next section we present the discretization of the forward problem and the formulation of the problem as an optimization problem. In section 3 we discuss the reduced Hessian method. In section 4 we give some results on inexact CG. In section 5 we give an algorithm to solve the complete inverse problem. In section 6 we present some numerical results comparing the different approaches mentioned before; and the paper concludes in section 7.

2. Details of the problem.

2.1. The forward problem. Recall that the forward problem is to find a field \mathbf{u} given \mathbf{m} satisfying the parabolic PDE (1.1). To be more specific, we use nodal discretization with a finite difference method for the space variables and a backward-Euler method for the time variable. On a uniform tensor grid on the interval $[0, 1]^3$, the forward problem yields the discrete system

$$M\mathbf{m} - \hat{A}u_{n+1} = \frac{u_{n+1} - u_n}{\delta t}, \quad \text{for } n = 0, \dots, k, \quad (2.1)$$

or

$$-u_n + (I + \delta t \hat{A})u_{n+1} - \delta t M\mathbf{m} = 0, \quad \text{for } n = 0, \dots, k, \quad (2.2)$$

where u_n is the approximation to $\mathbf{u}(x, n\delta t)$, \hat{A} is the discretization of the negative Laplace operator in 3D, and M is a matrix that incorporates the boundary conditions \mathbf{m} into the finite difference. In the forward problem one is given some boundary conditions \mathbf{m} and solves for the field \mathbf{u} as a function of time. The variable \mathbf{m} represents the nodes on the boundary and \mathbf{u} represents the nodes inside the domain.

It is possible to rewrite the system as a lower block bidiagonal system

$$\underbrace{\begin{bmatrix} B & & & & & \\ -I & B & & & & \\ & -I & B & & & \\ & & \ddots & \ddots & & \\ & & & -I & B & \end{bmatrix}}_K \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix}}_{\mathbf{u}} - \delta t \underbrace{\begin{bmatrix} M \\ M \\ \vdots \\ M \end{bmatrix}}_G \mathbf{m} = \underbrace{\begin{bmatrix} u_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_c,$$

where $B = (I + \delta t \hat{A})$. With the additional notation above, we thus rewrite our problem as

$$K\mathbf{u} - \delta t G\mathbf{m} = c. \quad (2.3)$$

There is no need to form the coefficient matrix K since the solution of the forward problem is straightforward. As soon as we have a solver for the first block we can obtain the solution for the rest by forward substitution. Nevertheless, the solution of the forward problem requires to invert k linear systems with the coefficient matrix $B = I + \delta t \hat{A}$. For 3D problems, such as our model problem, this can be an expensive process, even if very efficient solvers are used.

2.2. The optimization problem. In order to estimate \mathbf{m} we assume that, we measure some function of the field \mathbf{u} . Assume that we can write the observed data as

$$\mathbf{d}^{obs} = Q\mathbf{u} + \epsilon \quad (2.4)$$

where Q is the operator which projects the field onto the measurement locations in space and time and ϵ is noise which is Gaussian independent and identical distribution with standard deviation σ^2 . In our model problem we are interested only at the last time step. Therefore the matrix Q consists mainly of zero blocks, except for the last one, which is the identity. That is $Q = [0, 0, \dots, 0, I]$. One way to achieve our goal is to minimize the following quadratic data misfit term

$$\phi = \frac{1}{2} \|Q\mathbf{u} - \mathbf{d}^{obs}\|^2 + \frac{\gamma}{2} \|L\mathbf{m}\|^2. \quad (2.5)$$

The first term is the misfit and the second is the regularization. The regularization parameter $\gamma \geq 0$ and the regularization operator L can be chosen to penalize the energy or smoothness [14]. This misfit term arises naturally, since we want to find \mathbf{m} such that the difference between the predicted and the observed data is small.

The resulting optimization problem is written in constrained optimization form as

$$\min \phi = \frac{1}{2} \|Q\mathbf{u} - \mathbf{d}^{obs}\|^2 + \frac{\gamma}{2} \|L\mathbf{m}\|^2 \quad (2.6a)$$

$$\text{subject to } K\mathbf{u} - \delta t G\mathbf{m} = c. \quad (2.6b)$$

3. The reduced Hessian method. Introducing the Lagrangian

$$L(\mathbf{u}, \mathbf{m}, \lambda) = \frac{1}{2} \|Q\mathbf{u} - \mathbf{d}^{obs}\|^2 + \frac{\gamma}{2} \|L\mathbf{m}\|^2 + \lambda^T (K\mathbf{u} - \delta t G\mathbf{m} - c), \quad (3.1)$$

where λ is a vector of Lagrange multipliers with the same length as \mathbf{u} . A necessary condition in order to attain the minimum is that the first derivatives of (3.1) vanish. That is,

$$L_{\mathbf{u}} = Q^T(Q\mathbf{u} - \mathbf{d}^{obs}) + K^T\lambda = 0 \quad (3.2a)$$

$$L_{\mathbf{m}} = -\delta t G^T\lambda + \gamma L^T L\mathbf{m} = 0 \quad (3.2b)$$

$$L_{\lambda} = K\mathbf{u} - \delta t G\mathbf{m} - c = 0. \quad (3.2c)$$

Equations (3.2) lead to the following symmetric KKT system,

$$\begin{bmatrix} Q^T Q & 0 & K^T \\ 0 & \gamma L^T L & -\delta t G^T \\ K & -\delta t G & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{m} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q^T \mathbf{d}^{obs} \\ 0 \\ c \end{bmatrix}$$

which must be solved.

In the reduced Hessian approach, we eliminate \mathbf{u} from the last block row and λ from the first block row as follows:

$$\mathbf{u} = K^{-1}(c + \delta t G\mathbf{m}) \quad (3.3)$$

$$\lambda = K^{-T}(Q^T \mathbf{d}^{obs} - Q^T Q K^{-1} c - Q^T Q K^{-1} \delta t G\mathbf{m}) \quad (3.4)$$

Then we obtain the following reduced system for \mathbf{m} :

$$H\mathbf{m} = f \quad (3.5)$$

where

$$H = \delta t^2 J^T J + \gamma L^T L, \quad J = Q K^{-1} G$$

and

$$f = -\delta t J^T Q K^{-1} c + \delta t J^T \mathbf{d}^{obs}.$$

When solving (3.5) using CG, one needs to evaluate $H p_j$ at each step, where p_j is a direction vector. Unfortunately, evaluating $H p_j$, is nontrivial. It requires the

evaluation of $w = Jp_j$ and $J^T w$, and although the matrices G, Q and K are all large and sparse, the sensitivity matrix J is dense and thus it is never evaluated or stored. Instead the vector p_j is multiplied by G, Q, K^{-1} and their adjoints. To perform these products we rewrite the matrix H as follows

$$H = \delta t^2 G^T K^{-T} Q^T Q K^{-1} G + \gamma L^T L.$$

We postpone further analysis of the evaluation of Hp_j until section 5.

4. Inexact CG. We have reduced our inverse problem (2.6) to the solution of the symmetric positive definite linear system (3.5). It is then natural to use the conjugate gradient method (CG) for its solution. As we already mentioned, we propose to use an inexact version of CG, i.e., one in which the matrix-vector product at each iteration, say, Hp_j , is not performed exactly; see [3], [16], [18] for analysis of inexact Krylov subspace methods, including inexact CG. In the following we choose $\gamma = 0$.

In terms of implementation the only difference between standard CG, and inexact CG is the matrix-vector product. As is well known, there are several mathematically equivalent formulations of CG; see, e.g., [6], [11], [13].

In this section, we consider a general linear system $Ax = b$ with A symmetric positive definite. The inexact matrix-vector product of A and p_j is then $Ap_j + g_j$, where g_j is some “discrepancy” vector. It is often useful to think of this inexact matrix-vector product as $(A + E_j)p_j$ where E_j is an “error matrix”. In other words we have $E_j p_j = g_j$. We present our results using the algorithm from [15], where we only replace the standard step $r_{j+1} = r_j - \alpha_j Ap_j$ with an inexact version of it of the form for $\tilde{r}_{j+1} = \tilde{r}_j - \alpha_j (Ap_j + g_j)$.

Algorithm 1: inexact CG

1. $\tilde{r}_0 = b - Ax_0; p_0 = r_0$
 2. for $j = 0, 1, \dots$ until convergence, Do
 3. $\hat{q}_j = Ap_j + g_j$
 4. $\alpha_j = (\tilde{r}_j, \tilde{r}_j) / (\hat{q}_j, p_j)$
 5. $x_{j+1} = x_j + \alpha_j p_j$
 6. $\tilde{r}_{j+1} = \tilde{r}_j - \alpha_j \hat{q}_j$
 7. $\beta_j = (\tilde{r}_{j+1}, \tilde{r}_{j+1}) / (\tilde{r}_j, \tilde{r}_j)$
 8. $p_{j+1} = \tilde{r}_{j+1} + \beta_j p_j$
 9. end do
-

We collect the direction and discrepancy vectors in the following matrices, $P_m = [p_0 \ p_1 \ \dots \ p_{m-1}]$, $G_m = [g_0 \ g_1 \ \dots \ g_{m-1}]$. Let

$$T_{m+1,m} = \begin{bmatrix} (1+\beta_0)/\alpha_0 & -\beta_0/\alpha_1 & 0 & \cdots & 0 \\ -1/\alpha_0 & (1+\beta_1)/\alpha_1 & -\beta_1/\alpha_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -1/\alpha_j & (1+\beta_j)/\alpha_{j+1} & -\beta_{j+1}/\alpha_{j+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & 0 & 0 & -1/\alpha_{m-1} \end{bmatrix}.$$

It follows from the inexact CG algorithm that the following relation between these matrices hold:

$$AP_m + G_m = P_{m+1}T_{m+1,m}, \quad (4.1)$$

i.e.,

$$[Ap_0 + g_0 \quad Ap_1 + g_1 \quad Ap_2 + g_2 \quad \cdots \quad Ap_{m-1} + g_{m-1}] = [p_0 \quad p_1 \quad \cdots \quad p_m] T_{m+1,m}.$$

The residual gap is defined as $\|r_m - \tilde{r}_m\|$, where $r_m = b - Ax_m$ is the true residual and \tilde{r}_m is the computed residual (step 6 of Algorithm 1). Our goal is to find *computable* bounds for $\|E_j\|$, for $j \leq m$, in order to guarantee that the residual gap is below a prescribed tolerance. We begin with the following preliminary result.

LEMMA 4.1. *Let $\epsilon > 0$, let $r_m = b - Ax_m$ and \tilde{r}_m be the true and calculated residual of inexact CG after m iterations. Let $g_j = E_j p_j$, if*

$$\|E_j\| \leq \frac{\epsilon}{m|\alpha_j|\|p_j\|}, \quad (4.2)$$

then $\|r_m - \tilde{r}_m\| \leq \epsilon$.

Proof. Let $\hat{\alpha}_j = [\alpha_0 \quad \alpha_1 \quad \cdots \quad \alpha_{j-1}]$, and without loss of generality let $x_0 = 0$ then we have:

$$\begin{aligned} r_m &= b - Ax_m = b - A(x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \cdots + \alpha_{m-1} p_{m-1}) \\ &= b - Ax_0 - AP_m \hat{\alpha}_m = b - AP_m \hat{\alpha}_m = b - (P_{m+1} T_{m+1,m} - G_m) \hat{\alpha}_m \\ &= b - P_{m+1} T_{m+1,m} + G_m \hat{\alpha}_m = \tilde{r}_m + G_m \hat{\alpha}_m. \end{aligned}$$

Therefore

$$\begin{aligned} \|r_m - \tilde{r}_m\| &= \|G_m \hat{\alpha}_m\| = \left\| \sum_{j=0}^{m-1} \alpha_j g_j \right\| = \left\| \sum_{j=0}^{m-1} \alpha_j E_j p_j \right\| \leq \sum_{j=0}^{m-1} \|E_j\| |\alpha_j| \|p_j\| \\ &\leq \sum_{j=0}^{m-1} \|\alpha_j p_j\| \frac{\epsilon}{m|\alpha_j|\|p_j\|} \leq m \frac{\epsilon}{m} \leq \epsilon. \quad \square \end{aligned}$$

This lemma does not provide us with a computable bound, since at step 3 of the inexact CG Algorithm 1, we have p_j , but α_j has not been computed. We proceed then to find a computable bound for $|\alpha_j|$, in order to use it as an effective criterion for a bound of $\|E_j\|$. To that end, let us assume that $\|E_j\| \leq \sigma_{\min}(A)/2$, then

$$\begin{aligned} (Ap_j + g_j, p_j) &= (Ap_j, p_j) + (g_j, p_j) = (Ap_j, p_j) + (E_j p_j, p_j) \\ &\geq (Ap_j, p_j) - (E_j p_j, p_j) \geq \frac{\sigma_{\min}(A)}{2} \|p_j\|, \end{aligned} \quad (4.3)$$

Since $\alpha_j = (\tilde{r}_j, \tilde{r}_j)/(Ap_j + g_j, p_j)$, therefore we have that in this case

$$|\alpha_j| \leq \frac{2\|\tilde{r}_j\|^2}{\sigma_{\min}(A)\|p_j\|^2}.$$

Hence, for the bound in (4.2), we have that

$$\frac{\epsilon}{m|\alpha_j|\|p_j\|} \geq \frac{\epsilon\sigma_{\min}(A)\|p_j\|}{2m\|\tilde{r}_j\|^2},$$

which leads to the following result.

THEOREM 4.2. *Let $\epsilon > 0$ and let*

$$\|E_j\| \leq \min\left(\frac{\sigma_{\min}(A)}{2}, \frac{\epsilon\sigma_{\min}(A)\|p_j\|}{2m\|\tilde{r}_j\|^2}\right), \quad (4.4)$$

for $j = 0, 1, 2, \dots, m-1$, then $\|r_m - \tilde{r}_m\| \leq \epsilon$.

Observe that for most practical values of ϵ , the second quantity in (4.4) is smaller, i.e., we only need

$$\|E_j\| \leq \frac{\epsilon\sigma_{\min}(A)\|p_j\|}{2m\|\tilde{r}_j\|^2}. \quad (4.5)$$

We would like to analyze how the criterion (4.4) compares to other criteria used in the literature. In [16], Av_j is performed as in step 3 of Algorithm 1, and it is shown that if

$$\|E_j\| \leq \sigma_m(V_m)\sigma_m(H_{m+1,m})\epsilon/(m\|\tilde{r}_j\|) \text{ for } j \leq m-1, \quad (4.6)$$

then $\|r_m - \tilde{r}_m\| \leq \epsilon$.

Note that in our bound (4.4) the only quantity we need is $\sigma_{\min}(A)$, since \tilde{r}_j and p_j are available before the matrix-vector product $(A + E_j)p_j$.

We mention that in [18], an expression similar to (4.1) is obtained, but $T_{m+1,m}$ is presented as a product of different matrices. In the same reference, bounds for $|\alpha_j|\|p_j\|$ are also presented to use in criteria to guarantee that the residual gap is below a tolerance. Those bounds include quantities which often are harder to compute. and thus, we do not compare our bound (4.4) to those of [18].

In the remainder of this section we present a numerical experiment to compare our bound (4.4) with (4.6) from [16]. As we shall see, for this example, our bound is more stringent than (4.6). In the same experiment, we also calculate (4.2) a posteriori, and show how (4.4) compares to it.

Let $S = B^T B + 2I$, where B is a random $n \times n$ matrix. We consider a generic symmetric Schur complement problem of the form $A = C^T S^{-1} C$, where C is a random $n \times n$ matrix. We will first relate our condition (4.5) to this generic problem. At each step of CG, we compute Ap_j , i.e., $C^T S^{-1} Cp_j$, where the system

$$S z_j = C p_j \quad (4.7)$$

is not solved exactly. In other words, we have $S \tilde{z}_j = C p_j + \hat{g}_j$, and the inexact matrix-vector product is $\tilde{A} p_j = C^T (S^{-1} C p_j + S^{-1} \hat{g}_j)$, i.e., $E_j p_j = C^T S^{-1} \hat{g}_j$. Thus, by (4.5) we have the condition

$$\|C^T S^{-1} \hat{g}_j\| \leq \frac{\epsilon\sigma_n(A)\|p_j\|^2}{2m\|\tilde{r}_j\|^2}, \quad (4.8)$$

or alternatively we can impose that

$$\|\hat{g}_j\| \leq \frac{\epsilon \sigma_n(A) \|p_j\|^2}{2m \|C^T S^{-1}\| \|\tilde{r}_j\|^2}. \quad (4.9)$$

For the relative residual we have

$$\frac{\|\hat{g}_j\|}{\|C p_j\|} \leq \frac{\|\hat{g}_j\|}{\sigma_{\min}(C) \|p_j\|}$$

and together with the condition (4.9) we conclude that if

$$\frac{\|\hat{g}_j\|}{\|C p_j\|} \leq \frac{\sigma_n(A) \|p_j\|}{2m \sigma_{\min}(C) \|C^T S^{-1}\| \|\tilde{r}_j\|^2} =: \epsilon_{in}, \quad (4.10)$$

then the residual gap is less than ϵ .

We note that in the proposed bound for the (inner) relative residual (4.10) all quantities can either be computed *a priori* from the matrices of the problem, or they are available at the current CG step. This is in contrast with (4.6), where the quantities $\sigma_m(V_m)$ and $\sigma_m(H_{m+1,m})$ need to be estimated.

For our experiment, we choose $n = 400$, and run inexact CG, using as a right hand side $B^T e$, e being the vector of all ones. We chose $m = 60$ and $\epsilon = 10^{-8}$, and use (4.10) as the tolerance for the (inner) systems (4.7). We report this run in Figure 4.1. The computed and true residual norm are represented with dashed and dot-dashed lines, respectively (and they appear one on top of the other in the plot). Our computable tolerance (4.10) is represented with a dotted line (which corresponds to (4.4) as noted above). We also plot the non-computable bound (4.2) in a dash-dotted line, which computed in an a posteriori manner, as well as the bound (4.6) as a dashed line.

It can be appreciated in Figure 4.1 that the true and computed residuals are almost indistinguishable, as predicted by our theory. Note also that the proposed bound (4.10) is not too distant from the bound (4.6). The bound (4.2) would be less restrictive, but, as we already mentioned, it is not available at the time of the computations.

5. Application of inexact CG. We return now to our inverse problem (1.1), and its formulation using the reduced Hessian leading to the linear system (3.5). As discussed, we will use inexact CG for its solution. For this section we assume that $\gamma = 0$. For large values of γ the inexactness is less significant. In order to multiply H with a given vector p_j , we perform the following matrix-vector multiplication,

1. Multiply $G p_j$
2. Solve $K z = G p_j$ by solving $K z = G p_j$ with an inner tolerance ϵ_{in_1}
3. Multiply $Q z$
4. Multiply $Q^T Q z$
5. Solve $K^T w = Q^T Q z$ by solving with an inner tolerance ϵ_{in_2}
6. Compute $G^T w$

For the solution of the two linear systems in step 2 and 5 above, we would use the conjugate gradient method (CG) with ϵ_{in_1} and ϵ_{in_2} tolerances respectively. The question is how inexact these inner tolerances are allowed in order to ensure the convergence of the overall method. Here we give a derivation. We want to study how to implement the idea of solving these two linear systems with less and less accuracy, while obtaining a good solution to (3.5).

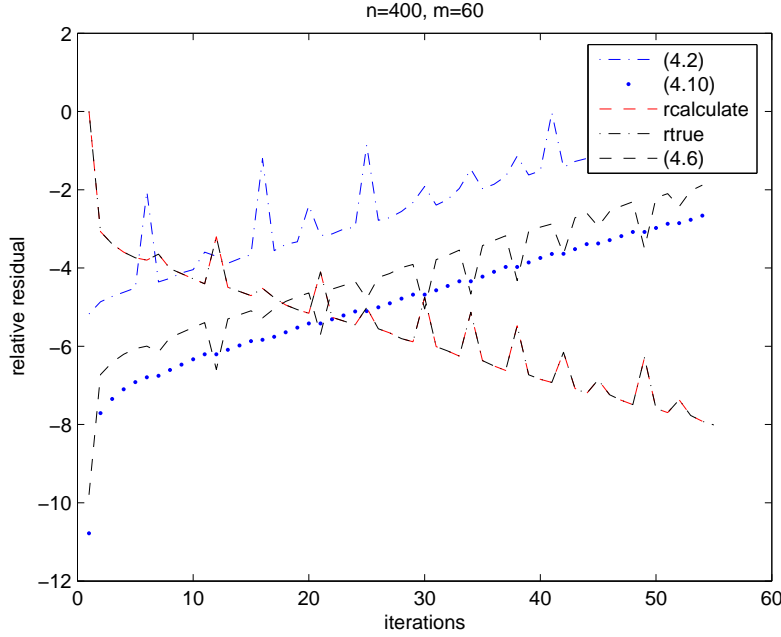


FIG. 4.1. True and computed residual norm, and different bounds on the inner relative residual.

To that end, we study how an inexact matrix-vector product Hp_j would look like using inexact solutions of the equations in steps 2 and 5 above. Let \hat{z} be the approximate solution to $Kz = Gp_j$ and let $s = K\hat{z} - Gp_j$ be its residual. Of course after step 3 and step 4 we have $Q^T Q\hat{z}$. Let \hat{w} be the approximate solution to $K^T w = Q^T Q\hat{z}$ and let $q = K^T \hat{w} - Q^T Q\hat{z}$ be its residual. Therefore, we have that $\hat{z} = K^{-1}s + K^{-1}Gp_j$ and $\hat{w} = K^{-T}q + K^{-T}Q^T Q\hat{z}$. Thus, in step 6 we have

$$\begin{aligned} G^T \hat{w} &= G^T K^{-T} q + G^T K^{-T} Q^T Q (K^{-1} s + K^{-1} G p_j) \\ &= G^T K^{-T} Q^T Q K^{-1} G p_j + (G^T K^{-T} q + G^T K^{-T} Q^T Q K^{-1} s). \end{aligned}$$

In other words, the inexact matrix vector product $G^T \hat{w}$ differs from the exact matrix-vector product $G^T w$ precisely by the vector is $G^T K^{-T} q + G^T K^{-T} Q^T Q K^{-1} s$. Then our discrepancy vector is

$$E_j p_j = G^T K^{-T} q + G^T K^{-T} Q^T Q K^{-1} s. \quad (5.1)$$

We have that $G^T \hat{w} = G^T w + E_j p_j$. This is for the j th step. Following Theorem 4.2 we will impose conditions so that

$$\|E_j\| < \epsilon \sigma_n(H) \|p_j\| \|\tilde{r}_0\| / (m \|\tilde{r}_j^2\|) \quad (5.2)$$

holds, where \tilde{r}_j is the j th computed residual, m is the maximum number of iterations that we allow, and ϵ is the tolerance that we want to obtain for the residual with the approximation to the solution of our discrete problem (3.5). From (5.1) and (5.2), it follows that if

$$\|E_j\| \leq (\|G^T K^{-T}\| \|q\| + \|G^T K^{-T} Q^T Q K^{-1}\| \|s\|) / \|p_j\| \leq \epsilon \sigma_n(H) \|p_j\| \|\tilde{r}_0\| / (m \|\tilde{r}_j^2\|),$$

then (5.2) holds. Thus, to achieve (5.2), it suffices to require that

$$\|q\| \leq \alpha \frac{\sigma_n(H) \|p_j\|^2 \|\tilde{r}_0\| \epsilon}{m \|G^T K^{-T}\| \|\tilde{r}_j\|^2},$$

and that

$$\|s\| \leq (1 - \alpha) \frac{\sigma_n(H) \|p_j\|^2 \|\tilde{r}_0\| \epsilon}{m \|G^T K^{-T} Q^T Q K^{-1}\| \|\tilde{r}_j\|^2}, \text{ for some } 0 < \alpha < 1.$$

Therefore we would impose on the relative residuals the following tolerance

$$\frac{\|s\|}{\|Gp_j\|} \leq (1 - \alpha) \frac{\sigma_n(H) \|p_j\|^2 \|\tilde{r}_0\| \epsilon}{m \|Gp_j\| \|G^T K^{-T} Q^T Q K^{-1}\| \|\tilde{r}_j\|^2} =: \epsilon_{in1} \quad (5.3)$$

and

$$\frac{\|q\|}{\|K^{-T} Q^T Q \hat{z}\|} \leq \alpha \frac{\sigma_n(H) \|p_j\|^2 \|\tilde{r}_0\| \epsilon}{m \|K^{-T} Q^T Q \hat{z}\| \|G^T K^{-T}\| \|\tilde{r}_j\|^2} =: \epsilon_{in2}.$$

The parameter α can be fixed (e.g., $\alpha = 1/2$), or it can vary from one step to the next. Thus, we have stopping criteria for each of the two residuals of the linear systems with coefficient matrices K and K^T .

For the systems that we consider here, we have

$$K = \begin{bmatrix} B & O & O & \cdots & O \\ -I & B & O & \cdots & O \\ 0 & \ddots & \ddots & \ddots & \\ \cdots & \cdots & \ddots & \ddots & \vdots \\ O & O & \cdots & -I & B \end{bmatrix},$$

having k diagonal blocks B . Let us denote by $(Gp_j)_i$ the i th block of Gp_j . Then for the solution of $Kz = Gp_j$, we have that $Bz_1 = (Gp_j)_1$, and $Bz_i = z_{i-1} + (Gp_j)_i$ for $2 \leq i \leq k$. Let s be, as before the residual $s = K\tilde{z} - Gp_j$, let

$$\begin{aligned} s_1 &= B\tilde{z}_1 - (Gp_j)_1 \\ s_i &= B\tilde{z}_i - (Gp_j)_i - \tilde{z}_{i-1} \text{ for } 2 \leq i \leq k. \end{aligned} \quad (5.4)$$

We want to have (5.3) satisfied, all we can control is the relative residual for (5.4). The relative residual for (5.4) is $\|s_1\|/\|(Gp_j)_1\|$ and $\|s_i\|/\|\tilde{z}_{i-1} + (Gp_j)_i\|$ for $i = 2 \leq k$. We have

$$\|s_1\|/\|(Gp_j)_1\| \geq \|s_1\|/\|Gp_j\|,$$

in order to satisfy (5.3), we can let $\|s_1\|/\|(Gp_j)_1\| \leq \frac{\epsilon \ell_1 \|p_j\|^2}{\|Gp_j\| \|\tilde{r}_j\|^2}$ with

$$\ell_1 = (1 - \alpha) \frac{\sigma_n(H) \|\tilde{r}_0\|}{m \|G^T K^{-T} Q^T Q K^{-1}\|}.$$

Also since

$$\begin{aligned} \frac{\|s_i\|}{\|\tilde{z}_{i-1} + (Gp_j)_i\|} &= \frac{\|s_i\|}{\|(Gp_j)_i + B^{-1}(s_{i-1} + (Gp_j)_{i-1}) + \dots + B^{1-i}(s_1 + (Gp_j)_1)\|} \\ &\simeq \frac{\|s_i\|}{(\|B^{1-i}\| + \dots + \|B^{-1}\| + 1)\|Gp_j\|} \text{ for } 2 \leq i \leq k. \end{aligned}$$

Let

$$\ell_2 = \min_{1 \leq i \leq k} \frac{1 - \|B^{-1}\|}{1 - \|B^{-1}\|^i}$$

and if

$$\frac{\|s_i\|}{\|\tilde{z}_{i-1} + (Gp_j)_i\|} \leq \frac{\ell_1 \ell_2 \|p_j\|^2}{\|Gp_j\| \|\tilde{r}_j\|^2}, \text{ for } 1 \leq i \leq k$$

we will have (5.3). We also have similar result for K^{-T} .

6. Numerical Experiments. We return to our original problem (2.6). For our numerical experiments, this problem is discretized first on a uniform $16 \times 16 \times 16$ grid, and we consider 10 time steps. That means that the size of the matrix B is 3375×3375 . Later we will perform the same experiments on $32 \times 32 \times 32$ grid. We use preconditioned conjugate gradient to solve the forward and the adjoint problems with symmetric Gauss-Seidel as a preconditioner.

As stated in the introduction, we experiment with several approaches for the tolerance of the inner iteration, which varies at each outer step of the algorithm. In our first set of experiments the outer and the inner tolerances are kept constant. In this way, 28 experiments are produced and the number of outer and inner iterations for these experiments are displayed in Table 6.1. We also provide the relative error in each case. For the inner tolerances we choose the following bound

$$\frac{\epsilon \ell_2 \|p_j\|^2}{\|Gp_j\| \|\tilde{r}_j\|^2} =: \epsilon_{in11}. \quad (6.1)$$

for $1 \leq i \leq k$, with $\ell_2 = 0.7469$ for $n = 16$ and similarly we define ϵ_{in21} for K^{-T} . In all cases reported here we use $\epsilon_{in} = \epsilon_{in11} = \epsilon_{in21}$.

As an example, in Table 6.1 when the requested outer tolerance is $\epsilon_{out} = 10^{-7}$ and the inner tolerance is $\epsilon_{in} = 10^{-7}$, we need 41 outer iterations and 15250 inner iterations. The corresponding relative error is then

$$\frac{\|x^{true} - x^{comp}\|}{\|x^{comp}\|} = 4.4584 \cdot 10^{-6}.$$

It can be appreciated from Table 6.1 that in many cases keeping the inner tolerance fixed does not always produce satisfactory result. On the other hand, in some cases (as when $\epsilon_{in} = 10^{-5}$) it is advantageous to have a larger inner tolerance.

In our next experiment we **tighten the accuracy of the matrix-vector products**. We run the same example with $\epsilon_{in} = 10^{-3} \cdot \|\tilde{r}_{k-1}\|$, while we set the outer tolerance $\epsilon_{out} = 10^{-7}$ and $\epsilon_{in11} = \epsilon_{in21} = \epsilon_{in} = 10^{-3} \cdot \|\tilde{r}_{k-1}\|$. This means that the matrix-vector multiplication may be performed in an increasingly “exact” way as the iteration progresses. This is not the approach we propose, but we present it here for completeness.

| inner outer | 10^{-7} | 10^{-6} | 10^{-5} | 10^{-4} |
|----------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 10^{-7} | 41/15250 4.4584×10^{-6} | 43/13958 4.8964×10^{-6} | 43/11783 2.6531×10^{-5} | 49/10848 2.9644×10^{-4} |
| 10^{-6} | 30/11210 4.4223×10^{-5} | 30/9785 5.9866×10^{-5} | 32/8762 6.3781×10^{-5} | 37/8246 2.9542×10^{-4} |
| 10^{-5} | 20/7496 3.8544×10^{-4} | 22/7197 3.3485×10^{-4} | 23/6324 3.2226×10^{-4} | 25/5555 4.2020×10^{-4} |
| 10^{-4} | 15/5608 2.3512×10^{-3} | 15/4942 2.4100×10^{-3} | 16/4387 2.2949×10^{-3} | 16/3561 2.3217×10^{-3} |
| 10^{-3} | 8/3013 2.0926×10^{-2} | 8/2641 2.0927×10^{-2} | 9/2466 1.9737×10^{-2} | 9/2012 1.9835×10^{-2} |
| 10^{-2} | 4/1514 9.3340×10^{-2} | 4/1325 9.3340×10^{-2} | 4/1111 9.3340×10^{-2} | 4/909 9.3363×10^{-2} |
| 10^{-1} | 1/357 4.4498×10^{-1} | 1/313 4.4498×10^{-1} | 1/261 4.4498×10^{-1} | 1/216 4.4498×10^{-1} |

TABLE 6.1

Outer/Inner number of iterations and the relative error

In Figure 6.1 we show the convergence history of the computed “inexact” and “exact” residual. The plot also displays ϵ_{in} as a function of the number of outer iterations. Since H is not available explicitly, for the true solution we use $\epsilon_{in} = 10^{-14}$ to obtain it.

It can be observed from this plot that the difference between the “inexact” and the “exact” residuals is amplified. At the end, the norm of the relative error is 1.821229×10^{-3} .

In Figure 6.2 we show the contour plot of one surface of the true model, the reconstructed surface and their difference. Although the difference between the true model and the reconstructed one is not obvious with the naked eye, the last plot reveals that this difference is of the order of 10^{-3} .

Consider now **relaxing the accuracy of matrix-vector product**: again we set $\epsilon_{out} = 10^{-7}$, while now ϵ_{in} is either taken as in (6.1) or $\epsilon_{in} = 10^{-8} \frac{1}{\|\tilde{r}_{k-1}\|}$. First we test our proposed bound (6.1), i.e., we relax the matrix-vector product. In Figure 6.3 we show the convergence history of the computed residual compared with the “exact” residual with inner tolerance satisfy (6.1). As it can be appreciated, using our proposed criterion (6.1), the computed residual using inexact matrix-vector products (dash-dotted line) stays very close to the “exact” residuals (solid line), with considerably savings in computational effort.

In Figure 6.4 we show the contour plot of one surface of the true model, the reconstructed surface and their difference. The last plot reveals that the difference between the true model and the reconstructed solution is of order of 10^{-6} and more uniform than that of Figure 6.2.

Our next experiments mimic the conditions where it is hard to even compute the problem-dependent quantities in our criteria, such as $\sigma_n(H)$. We do know that relaxing the matrix-vector product is advantageous and we thus use simply $\epsilon_{in} = \ell\epsilon/\|\tilde{r}_j\|$, so here we choose $\ell = 0.1$, (since $\|p_j\|/\|\tilde{r}_j\|$ is bounded,) so that for $\epsilon_{out} = 10^{-7}$, $\epsilon_{in} = 10^{-8}/\|\tilde{r}_j\|$. We report this in Figure 6.5, where it can be seen that the result is still acceptable, but that the inexact run has some delay in convergence when

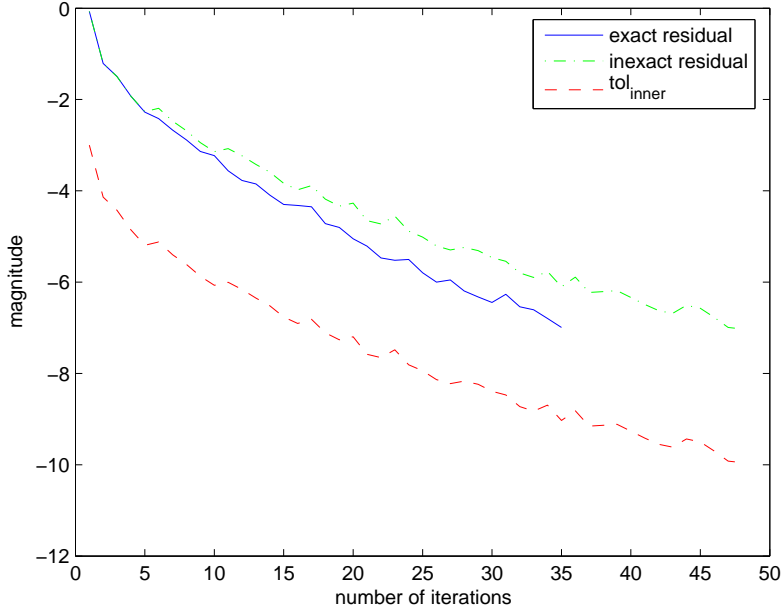


FIG. 6.1. *Relative residual norm of exact and inexact multiplications*

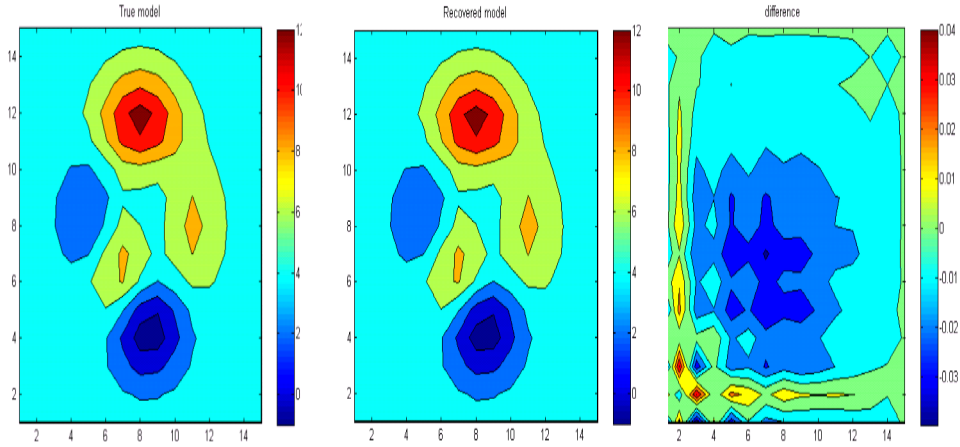


FIG. 6.2. *One surface of the true and the recovered model and their difference*

compared to the Figure 6.3. We show more details on this later in Table 6.2.

Furthermore, we present in Figure 6.6 the contour plot of one surface of the true model, the reconstructed surface and their difference. Note that the difference is of the order 10^{-6} , in fact,

$$\frac{\|x^{true} - x^{comp}\|}{\|x^{comp}\|} = 5.78396 \cdot 10^{-6}.$$

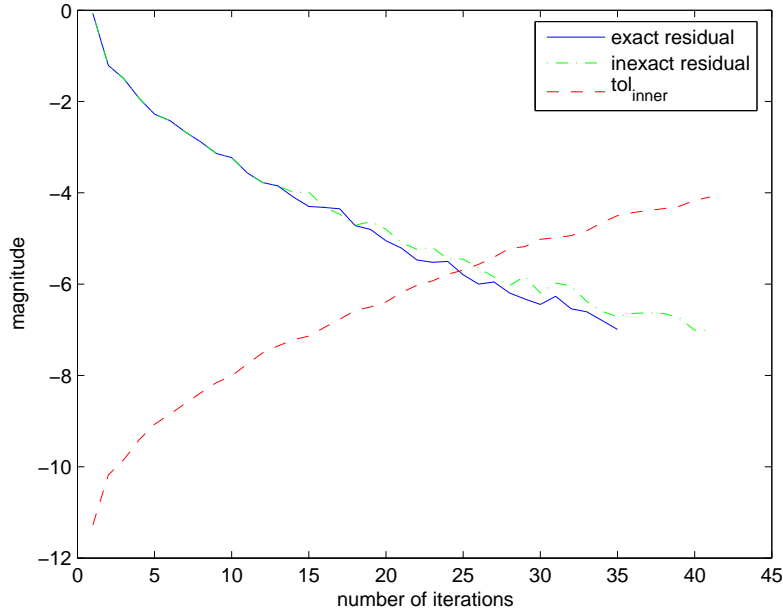
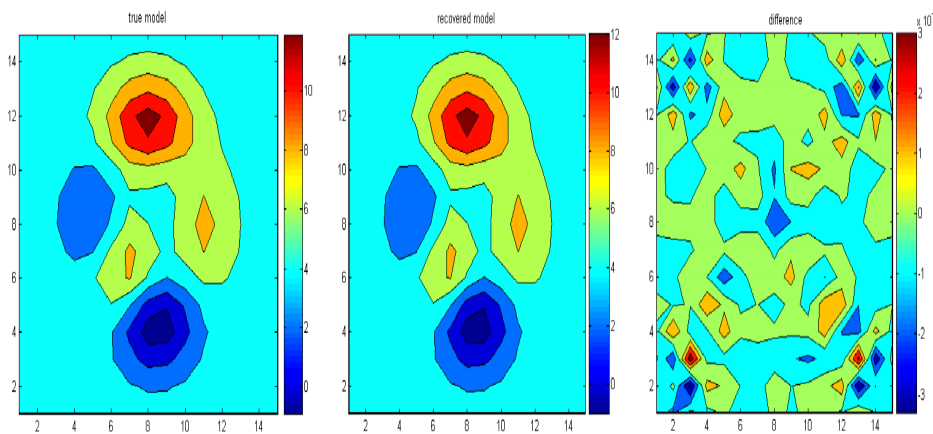
FIG. 6.3. exact and inexact convergence for $n = 16$ with (6.1)

FIG. 6.4. One surface of the true and the recovered model and their difference

To conclude, we present Tables 6.2 and 6.3, where we report total number of outer and inner CG iterations for $16 \times 16 \times 16$ grid and $32 \times 32 \times 32$ grid, respectively, for different values of the outer tolerance ϵ_{out} , and the different inner tolerances discussed, namely fixed with $\epsilon_{in} = 10^{-7}$, increasing tightens $\epsilon_{in} = 10^{-3} \|\tilde{r}_j\| / \|r_0\|$, relaxing the matrix-vector product with using (6.1) or $\epsilon_{in} = 10^{-8} \|r_0\| / \|\tilde{r}_j\|$. The total number of inner iterations represent a good measure of the total work. As before, the first number stands for the number of outer iterations and the second for the number of

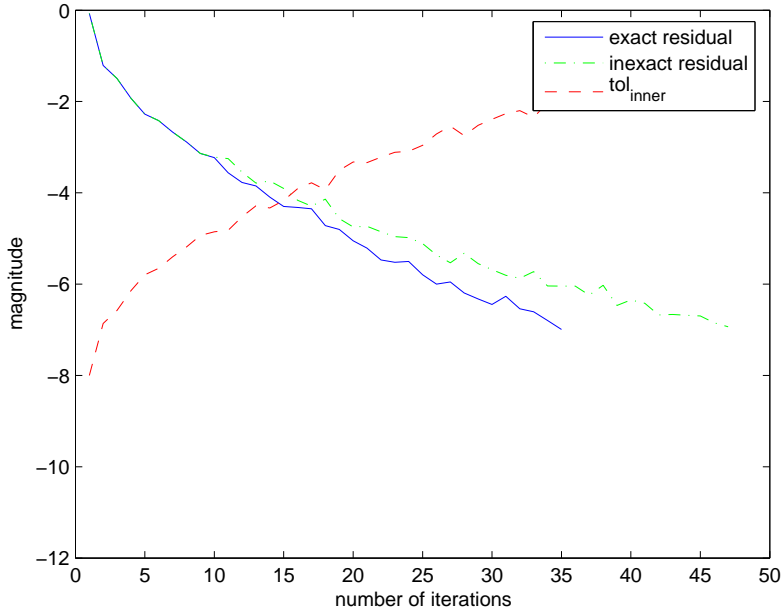


FIG. 6.5. *Relative residual norm of exact and inexact multiplications*

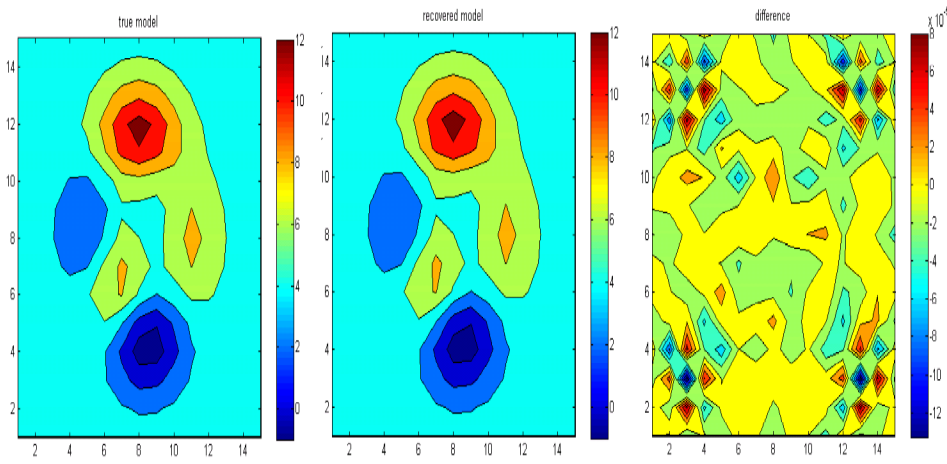


FIG. 6.6. *One surface of the true and the recovered model and their difference*

inner iterations.

We note that since the tolerance $\epsilon_{in} = 10^{-8} / \|\tilde{r}_j\|$ is less restrictive than (6.1) (cf. Figures 6.3 and 6.5), the number of inner iterations to satisfy it is smaller than that needed to (6.1). With either of these criteria, savings of computational time of up to 50% over a fixed inner tolerance can be achieved.

| outer inner | 10^{-7} | 10^{-6} | 10^{-5} | 10^{-4} |
|---------------------------------------|-----------|-----------|-----------|-----------|
| 10^{-7} | 41/15250 | 30/11210 | 20/7496 | 15/5608 |
| $10^{-3} \cdot \ \tilde{r}_{k-1}\ $ | 48/18982 | 34/12266 | 25/8310 | 17/5157 |
| $\frac{10^{-8}}{\ \tilde{r}_{k-1}\ }$ | 47/8689 | 34/7796 | 24/6180 | 15/4453 |
| (6.1) | 41/14242 | 30/9996 | 21/6810 | 15/4590 |

TABLE 6.2

Outer/Inner number of iterations on a $16 \times 16 \times 16$ grid

| outer inner | 10^{-7} | 10^{-6} | 10^{-5} | 10^{-4} |
|---------------------------------------|-----------|-----------|-----------|-----------|
| 10^{-7} | 54/36974 | 40/27589 | 28/19377 | 18/12463 |
| $\frac{10^{-8}}{\ \tilde{r}_{k-1}\ }$ | 62/19609 | 46/17666 | 31/14179 | 19/9941 |
| (6.1) | 54/36404 | 38/25223 | 28/17566 | 18/10850 |

TABLE 6.3

Outer/Inner number of iterations on a $32 \times 32 \times 32$ grid

7. Conclusions. In this paper we have proposed computable tolerances for calculating inexact matrix-vector products based on an optimization problem subject to the time-dependent heat equation. As the experiments show, by dynamically relaxing the stopping criteria of the inner matrix-vector multiplications, i.e., the inner tolerance, we can achieve computational savings up to 50%.

REFERENCES

- [1] George Biros and Omar Gattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-Constrained optimization. Part I: The Krylov-Schur solver. *SIAM Journal on Scientific Computing*, 27:687–713, 2005.
- [2] George Biros and Omar Gattas. Parallel Lagrange-Newton-Krylov-Schur methods for pde-constrained optimization. Part II: the Lagrange-Newton solver and its application to optimal control of steady viscous flows. *SIAM Journal on Scientific Computing*, 27(2):714–739, 2005.
- [3] Anthony J. Devaney. The limited view problem in diffraction tomograph. *Inverse Problems*, 5:510–23, 1989.
- [4] Stan E. Dosso. *Solution of the 1D Magnetotelluric Problem*. PhD thesis, 1990. Department of Earth and Ocean Sciences, University of British Columbia.
- [5] Luc Giraud, Serge Gratton, and Julien Langou. Convergence in backward error of relaxed GMRES. *SIAM Journal on Scientific Computing*, 29:710–728, 2007.
- [6] Gene H. Golub and Qiang Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM Journal on Scientific Computing*, 21:1305–1320, 2000.
- [7] Eldad Haber and Uri M. Ascher. Fast finite volume simulation of 3d electromagnetic problems with highly discontinuous coefficients. *SIAM Journal on Scientific Computing*, 22:1943–1961, 2001.
- [8] Eldad Haber and Uri M. Ascher. Preconditioned all-at-one methods for large, sparse parameter estimation problems. *Inverse Problems*, 17:1847–1864, 2001.
- [9] Eldad Haber, Uri M. Ascher, and Douglas Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16:1263–1280, 2000.
- [10] Carl T. Kelley. *Iterative Methods for Optimization*, Frontiers in Applied Mathematics. SIAM, Philadelphia, 1999.
- [11] Gérard Meurant. *The Lanczos and Conjugate Gradient Algorithms*. SIAM, Philadelphia, 2006.
- [12] Jorge Nocedal and Stephen Wright. *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006. Second Edition.
- [13] Yvan Notay. Flexible Conjugate Gradient. *SIAM Journal on Scientific Computing*, 22:1444–

- 1460, 2000.
- [14] Robert L. Parker. *Geophysical Inverse Theory*. Princeton University Press, 1994.
 - [15] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. The PWS Publishing Company, Boston, 1996. Second edition, SIAM, Philadelphia, 2003.
 - [16] Valeria Simoncini and Daniel B. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, 25:454–477, 2003.
 - [17] Nariida C. Smith and Keeva Vozoff. Two-dimensional DC resistivity inversion for dipole data. *IEEE Trans. Geosci. Remote Sens.*, 22:21-8, 1984.
 - [18] Jasper van den Eshof and Gerard L. G. Sleijpen. Inexact Krylov subspace methods for linear systems. *SIAM Journal of Matrix Analysis Applications*, 26:125–153, 2004.