INSTITUTE OF PHYSICS PUBLISHING INVERSE PROBLEMS

Inverse Problems **17** (2001) 1847–1864 PII: S0266-5611(01)20474-2

Preconditioned all-at-once methods for large, sparse parameter estimation problems

E Haber[1,2] **and U M Ascher**[1]

[1] Department of Computer Science, University of British Columbia, Vancouver, BC, Canada V6T 1Z4
[2] Department of Earth and Ocean Science, University of British Columbia, Vancouver, BC, Canada V6T 1Z4

E-mail: haber@cs.ubc.ca and ascher@cs.ubc.ca

Received 5 January 2001, in final form 31 July 2001
Published 15 November 2001
Online at stacks.iop.org/IP/17/1847

Abstract

The problem of recovering a parameter function based on measurements of solutions of a system of partial differential equations in several space variables leads to a number of computational challenges. Upon discretization of a regularized formulation a large, sparse constrained optimization problem is obtained. Typically in the literature, the constraints are eliminated and the resulting unconstrained formulation is solved by some variant of Newton's method, usually the Gauss–Newton method. A preconditioned conjugate gradient algorithm is applied at each iteration for the resulting reduced Hessian system.

Alternatively, in this paper we apply a preconditioned Krylov method directly to the KKT system arising from a Newton-type method for the constrained formulation (an 'all-at-once' approach). A variant of symmetric QMR is employed, and an effective preconditioner is obtained by solving the reduced Hessian system approximately. Since the reduced Hessian system presents significant expense already in forming a matrix–vector product, the savings in doing so only approximately are substantial. The resulting preconditioner may be viewed as an incomplete block-LU decomposition, and we obtain conditions guaranteeing bounds for the condition number of the preconditioned matrix.

Numerical experiments are performed for the dc-resistivity and the magnetostatic problems in 3D, comparing the two approaches for solving the linear system at each Gauss–Newton iteration. A substantial efficiency gain is demonstrated. The relative efficiency of our proposed method is even higher in the context of inexact Newton-type methods, where the linear system at each iteration is solved less accurately.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

The problem of recovering a parameter function based on measurements of solutions of a system of partial differential equations (PDEs) in several space variables leads to a number of computational challenges. Such problems arise in many applications, including groundwater flow, dc resistivity, magnetotelluric inversion, diffraction tomography, impedance tomography and more; see [14, 16, 19, 29, 42, 46, 49] and references therein.

For instance, consider the *inverse problem* of recovering an approximation for a model, $m(\boldsymbol{x})$, based on measurement data b on the solution $u(\boldsymbol{x})$ of the *forward problem*

$$\nabla \cdot \left(\mathrm{e}^{m} \nabla u\right) = q, \qquad \boldsymbol{x} \in \Omega \tag{1.1a}$$

$$\nabla u \cdot \boldsymbol{n} = 0, \qquad \boldsymbol{x} \in \partial\Omega \tag{1.1b}$$

$$\int_{\Omega} u \, \mathrm{d}\Omega = 0, \tag{1.1c}$$

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

A typical formulation of this inverse problem would seek $\mathrm{m}(\boldsymbol{x})$ to minimize the sum of a data fitting error term and a regularization term, subject to the forward problem (1.1) being satisfied. The problem is typically ill-posed without regularization and it is ill-conditioned with it, since the regularization term is aimed at removing noise without overshadowing or 'flattening' the data.

To obtain a numerical solution the differential terms must be discretized. This typically leads to an algebraic, constrained optimization problem, where the various Jacobian matrices are very large but sparse. In most works reported in the literature (e.g., [42, 48, 49]), the constraints are eliminated and the resulting unconstrained formulation is solved by some variant of Newton's method, usually the Gauss–Newton method. A preconditioned conjugate gradient (CG) algorithm is applied at each iteration for the resulting reduced Hessian system. The reduced Hessian is a dense matrix, and special care must be taken to form matrix–vector products efficiently.

In this paper we apply instead a preconditioned Krylov method directly to the KKT system arising from a Newton-type method for the constrained formulation [41]. Such a formulation is also referred to as the 'all-at-once' approach [35, 44], since the solution of the forward problem is computed simultaneously with the solution of the inverse problem (cf [38]). In this paper, a variant of symmetric QMR [24, 25] is employed and an effective preconditioner is obtained by solving the reduced Hessian system approximately. Since the reduced Hessian system presents significant expense already in forming a matrix–vector product, the savings in doing so only approximately are substantial. The resulting preconditioner may be viewed as an incomplete block-LU decomposition, and we obtain conditions guaranteeing bounds for the condition number of the preconditioned matrix. An even more impressive improvement is obtained when the KKT system at each nonlinear iteration is solved only approximately, i.e., an *inexact Newton-type* method for the *constrained* formulation is utilized.

The approach is analogous to work for the incompressible Navier–Stokes equations in [20, 21, 45]; however, there are substantial differences in the model equations, their conditioning and the resulting algorithms.

Even closer to our approach, and very recent, are [2, 9, 11–13]. Indeed, Biros and Ghattas proposed a similar preconditioner in the context of optimal fluid flow control. In this paper we supply a new constructive convergence proof which naturally leads to grid-independent convergence rates under reasonable conditions. Furthermore, we study the preconditioner in the context of *distributed* parameter estimation in 3D where (unlike most optimal control problems) the number of model parameters is very large and the algorithm is required to perform

with rather small regularization parameters[3]. These recent references add to our conviction that the approach described here is viable and important for a wide range of applications coupling differential equations and optimization techniques.

To be specific, we introduce some notation next. Let the forward problem be a linear, elliptic differential system,

$$\mathcal{A}(m)\, u \;=\; q, \tag{1.2}$$

where $\mathcal{A}$ refers to a differential operator depending on a model $m(\boldsymbol{x})$, defined on an appropriate domain Ω and equipped with suitable boundary conditions. An instance of (1.2) is given by (1.1).

Let the operator Q indicate the projection onto the locations in Ω to which the data are associated. Thus, the data are viewed as a nonlinear function of the model:

$$b = \mathrm{Q}\mathcal{A}(m)^{-1}q + \epsilon,$$

where ϵ is measurement noise. Since the data are noisy, and the inverse problem of recovering m from it is often ill-posed even without noise [22, 48], there is no unique model which generates the data. Therefore, a process of regularization is used to recover a relatively smooth (or piecewise smooth), locally unique solution to a nearby problem.

A regularization method often utilized in practice minimizes the Tikhonov functional [22, 48, 49][4]

$$\min_m \tfrac{1}{2}\|\mathrm{Q}\mathcal{A}(m)^{-1}q - b\|^2 + \beta \mathsf{R}(m - m_{\mathrm{ref}}), \tag{1.3}$$

where m_{ref} is a reference model and $\beta \geqslant 0$ is the regularization parameter. Typical choices (with $m_{\mathrm{ref}} = 0$) are

$$\mathsf{R} = \int_\Omega k_0 m^2 + \tfrac{1}{2}|\nabla m|^2, \tag{1.4a}$$

or

$$\mathsf{R} = \int_\Omega k_0 m^2 + |\nabla m|, \tag{1.4b}$$

or some combination of the two.

The PDE in (1.2) is subsequently discretized on a grid by a finite volume or finite element method to read

$$A(m)u = q, \tag{1.5}$$

where A is a nonsingular matrix[5], u is the grid function approximating $u(\boldsymbol{x})$ and arranged as a vector, and m and q likewise relate to $m(\boldsymbol{x})$ and $q(\boldsymbol{x})$. The regularization functional is discretized similarly; see, e.g., [6] for a detailed example. The resulting optimization problem is written in constrained form as

$$\begin{array}{ll} \text{minimize} & \tfrac{1}{2}\|Qu - b\|^2 + \beta \mathsf{R}(m - m_{\mathrm{ref}}) \\ \text{subject to} & A(m)u - q = 0. \end{array} \tag{1.6}$$

The following description is a slight generalization of [31]. Introducing the Lagrangian

$$\mathcal{L}(u, m, \lambda) = \tfrac{1}{2}\|Qu - b\|^2 + \beta \mathsf{R}(m - m_{\mathrm{ref}}) + \lambda^T[A(m)u - q], \tag{1.7}$$

[3] Other well known differences from optimal control exist as well.

[4] Throughout this paper the l_2-norm is assumed unless specifically indicated otherwise.

[5] The assumptions of linearity of the forward problem (1.2) with respect to u, as well as the nonsingularity of A, can be relaxed.

a necessary condition for an optimal solution of our problem is

$$\mathcal{L}_u = Q^T(Qu - b) + A^T\lambda = 0, \tag{1.8a}$$

$$\mathcal{L}_m = \beta \mathsf{R}_m + G^T\lambda = 0, \tag{1.8b}$$

$$\mathcal{L}_\lambda = Au - q = 0, \tag{1.8c}$$

where

$$\mathsf{R}_m = \mathsf{R}_m(m) = \frac{\partial \mathsf{R}}{\partial m}, \qquad G = G(u, m) = \frac{\partial (Au)}{\partial m}.$$

If (1.4*a*) is used then

$$\mathsf{R}(m - m_{\text{ref}}) = \tfrac{1}{2}\|W(m - m_{\text{ref}})\|^2, \tag{1.9}$$

where W is a scaled finite difference or finite element matrix which does not depend on m (e.g., [6]). Using (1.8*c*) to eliminate u and then (1.8*a*) to eliminate λ yields a nonlinear least squares data fitting problem. The Gauss–Newton method (e.g. [18, 41]) has been widely employed for the latter problem; however, here we stay with the more general, constrained formulation.

A Newton linearization for solving the nonlinear equations (1.8) leads to the following KKT system to be solved at each iteration:

$$H_{kkt}\begin{pmatrix}\delta u\\ \delta m\\ \delta\lambda\end{pmatrix} = -\begin{pmatrix}\mathcal{L}_u\\ \mathcal{L}_m\\ \mathcal{L}_\lambda\end{pmatrix}, \qquad \text{where} \quad H_{kkt} = \begin{pmatrix}Q^TQ & K & A^T\\ K^T & \beta\mathsf{R}_{mm} + T & G^T\\ A & G & 0\end{pmatrix} \tag{1.10}$$

with

$$K = K(m, \lambda) = \frac{\partial (A^T\lambda)}{\partial m}, \qquad \mathsf{R}_{mm} = \mathsf{R}_{mm}(m) = \frac{\partial \mathsf{R}_m}{\partial m}, \qquad T = T(u, m, \lambda) = \frac{\partial (G^T\lambda)}{\partial m}.$$

This is the large, sparse, linear system on which this paper concentrates.

The only invertible blocks in H_{kkt} are those on the cross-diagonal. In particular, in the upper left corner Q^TQ is normally singular, and the usual Schur complement method [20, 45] will not work. One approach is the augmented Lagrangian method [23, 26, 36]. This adds $\begin{pmatrix}A^T\\ G^T\end{pmatrix}\begin{pmatrix}A & G\end{pmatrix}$ to the upper-left corner of H_{kkt}, making this block invertible and dominant. However, the extra term introduces a significant added complication.

Let us instead permute block rows and columns in (1.10), obtaining

$$H\begin{pmatrix}\delta u\\ \delta\lambda\\ \delta m\end{pmatrix} = -\begin{pmatrix}\mathcal{L}_\lambda\\ \mathcal{L}_u\\ \mathcal{L}_m\end{pmatrix}, \qquad \text{where} \quad H = \begin{pmatrix}A & 0 & G\\ Q^TQ & A^T & K\\ K^T & G^T & \beta\mathsf{R}_{mm} + T\end{pmatrix}. \tag{1.11}$$

If β is not small then the diagonal blocks in H dominate, corresponding to dominant terms in an elliptic PDE system. Several efficient methods for solving such a system are known, and the problem is no longer difficult. But practical values of β are usually 'not large', and the system in (1.11) must be viewed as strongly coupled.

We thus consider block elimination. Specifically, solution methods and their performance for the reduced Hessian are discussed in section 2. In section 3 we then introduce our preconditioned Krylov method for the KKT system. We show that a preconditioner based on the reduced Hessian can be very effective.

Numerical experiments are performed in section 4 for the dc-resistivity problem (1.1) and for the magnetostatic problem (i.e. Maxwell's equations for a zero frequency) in 3D, comparing the two approaches for solving the linear system at each Gauss–Newton iteration. A substantial efficiency gain is demonstrated in sections 4.1.1 and 4.2.1. The relative efficiency

of our proposed method is even higher in the context of inexact Newton-type methods, where the linear system at each nonlinear iteration is solved less accurately, as demonstrated in sections 4.1.2 and 4.2.2.

Conclusions and further remarks are offered in section 5.

2. The reduced Hessian method

Following [31], we proceed to eliminate δu, then $\delta\lambda$, and finally solve for δm in (1.11):

$$\delta u = -A^{-1}[\mathcal{L}_\lambda + G\delta m], \tag{2.12a}$$

$$\begin{aligned}\delta\lambda &= -A^{-T}[\mathcal{L}_u + Q^T Q\delta u + K\delta m] \\ &= -A^{-T}[Q^T J + K]\delta m - A^{-T}[\mathcal{L}_u - Q^T QA^{-1}\mathcal{L}_\lambda],\end{aligned} \tag{2.12b}$$

$$\delta m = -H_{\rm red}^{-1} p, \tag{2.12c}$$

where

$$H_{\rm red} = H_{\rm red}(u, m, \lambda) = J^T J + \beta \mathsf{R}_{mm} + T - S - S^T, \tag{2.13a}$$

$$J = J(u, m) = -QA^{-1}G, \tag{2.13b}$$

$$S = S(u, m, \lambda) = K^T A^{-1} G, \tag{2.13c}$$

and

$$p = p(u, m, \lambda) = \beta \mathsf{R}_m + J^T(QA^{-1}q - b) - K^T(u - A^{-1}q). \tag{2.13d}$$

In the Gauss–Newton approximation, second-order information is ignored by setting $T = 0$, $K = 0$. Hence the reduced Hessian becomes the symmetric positive definite matrix

$$H_{\rm red} = J^T J + \beta \mathsf{R}_{mm}. \tag{2.14}$$

Furthermore, Newton variants for the unconstrained formulation (i.e. for the discretization of (1.3)) are also accommodated here, upon setting u and λ according to (1.8c) and (1.8a) respectively, at the beginning of each iteration. Thus, the most popular, classical Gauss–Newton method can also be implemented more efficiently by using the constrained methodology described in the next section.

To focus our arguments, we consider for the rest of this section the relatively simple Gauss–Newton method. For the solution of (2.12c) we may use a PCG, CGLS or LSQR method [8, 41, 43, 49]. However, a matrix–vector product $H_{\rm red}v$ for some given vector v requires the evaluation of $w = Jv$ and of $J^T w$.

Since the matrix J is large and dense, the evaluation of Jv proceeds by first forming Gv, then solving the forward problem to obtain $A^{-1}Gv$ and finally multiplying the result by $-Q$: all of these are obtained by sparse matrix computations [31, 49]. But even with this, each evaluation of $H_{\rm red}v$ requires the solution of one forward problem and one adjoint problem to a high degree of accuracy, so it is expensive as such.

Moreover, the convergence of these methods without preconditioning may be slow. This is due to the different spectral character of the operators $J^T J$ and R_{mm}. While the former is a compact operator with eigenvalues which cluster at 0, the latter is a discretization of a differential operator with eigenvalues which cluster in the limit at infinity. As discussed in [49], a straightforward idea is to precondition the system using R_{mm}. Especially for the simpler regularization (1.9), this involves solving a Poisson equation which in discretized form yields the operator $\mathsf{R}_{mm} = W^T W$.

Such a preconditioning corresponds to the employment of the system

$$\mathsf{R}_{mm}^{-1}(J^T J + \beta \mathsf{R}_{mm}) = \mathsf{R}_{mm}^{-1} J^T J + \beta I$$

where the first term on the right bounds the largest eigenvalue of the system from above and βI bounds the spectrum from below. Therefore, PCG converges linearly for this problem, independent of the grid [22, 49].

Unfortunately, each PCG iteration can be quite expensive. This is because of the combined effect of (i) having to invert A, A^T and R_{mm}, and (ii) the need to obtain accuracy to a small tolerance in order to maintain sufficient conjugacy of the calculated approximations (see [27] and section 4.1.1).

3. A preconditioned Krylov method for the KKT system

In the previous section we discussed the solution of the preconditioned reduced Hessian system. In this section we consider the solution of the KKT system (1.10) or (1.11) directly.

It is well known [24, 25, 43] that Krylov methods for such systems tend to converge very slowly without appropriate preconditioning. Since H_{kkt} is symmetric but far from being positive definite, we expect to be able to construct preconditioners with similar attributes. The idea of using indefinite preconditioners for the KKT system has been discussed in [24, 25], where it was shown that, although the product of the preconditioner with the KKT system (1.10) is no longer symmetric (which rules out use of the popular MINRES or SYMMLQ methods [45]), it is possible to apply a symmetric QMR variant (which we refer to as SQMR) for the solution of the system. Similar preconditioners are used in [11, 13] for different applications.

The preconditioned version of this method, denoted PSQMR, involves one matrix–vector product and one preconditioning per iteration, which is roughly half the cost of a usual preconditioned QMR or BiCGstab iteration.

In order to devise an effective preconditioner for the KKT system we return to the reduced Hessian method of the previous section. We recall that each iteration in (2.12) involves the solution of linear systems with A, A^T and H_{red}.

Let B be an approximation to A^{-1} and let M_{red} be an approximation to H_{red}^{-1}. These approximate inverses yield an *approximate inverse* M to the permuted KKT matrix H, upon using the reduced Hessian methodology. Furthermore, if we choose B and M_{red} such that their product with a vector can be rapidly calculated then the product of the approximate inverse M with a vector can also be easily and rapidly computed. Such an approximation can be used as a preconditioner for the KKT system.

Thus, suppose that we are to calculate $x = Mv$ for a given vector $v = [v_\lambda^T, v_u^T, v_m^T]^T$. We write x in component form, $x = [x_u^T, x_\lambda^T, x_m^T]^T$, follow (2.12), and obtain the following.

Preconditioning algorithm.

(1) $w_1 = Bv_\lambda$;
(2) $w_2 = B^T(v_u - Q^T Q w_1)$;
(3) $w_3 = v_m - K^T w_1 - G^T w_2$;
(4) $x_m = M_{\text{red}} w_3$;
(5) $x_u = w_1 - BGx_m$;
(6) $x_\lambda = B^T(v_u - Q^T Q x_u - K x_m)$.

This preconditioner is suitable for (1.11). Note the reordering necessary in v and x in order to obtain the corresponding preconditioner for (1.10). Upon such reordering, the corresponding preconditioning matrix for (1.10) becomes symmetric (and indefinite), hence the symmetric QMR algorithm may be used. This yields our PSQMR algorithm.

3.1. Analysis of the reduced Hessian preconditioner

The performance of the preconditioner M is intimately related to the choices of approximate inverses B and M_{red}. In this section we analyse this effect.

For the matrix H in the permuted KKT system (1.11) the reduced Hessian methodology corresponds to a block-LU decomposition,

$$H = LU, \qquad L = \begin{pmatrix} I & 0 & 0 \\ Q^T Q A^{-1} & I & 0 \\ K^T A^{-1} & G^T A^{-T} & I \end{pmatrix}, \quad U = \begin{pmatrix} A & 0 & G \\ 0 & A^T & K + Q^T J \\ 0 & 0 & H_{\text{red}} \end{pmatrix}. \tag{3.15}$$

Note that

$$H^{-1} = U^{-1} L^{-1}, \qquad U^{-1} = \begin{pmatrix} A^{-1} & 0 & -A^{-1} G H_{\text{red}}^{-1} \\ 0 & A^{-T} & -A^{-T}(K + Q^T J) H_{\text{red}}^{-1} \\ 0 & 0 & H_{\text{red}}^{-1} \end{pmatrix},$$

$$L^{-1} = \begin{pmatrix} I & 0 & 0 \\ -Q^T Q A^{-1} & I & 0 \\ -(K^T + J^T Q) A^{-1} & -G^T A^{-T} & I \end{pmatrix}.$$

Defining correspondingly,

$$\bar{J} = -QBG, \qquad \hat{U} = \begin{pmatrix} B & 0 & -BGM_{\text{red}} \\ 0 & B^T & -B^T(K + Q^T \bar{J}) M_{\text{red}} \\ 0 & 0 & M_{\text{red}} \end{pmatrix}, \tag{3.16a}$$

$$\hat{L} = \begin{pmatrix} I & 0 & 0 \\ -Q^T Q B & I & 0 \\ -(K^T + \bar{J}^T Q) B & -G^T B^T & I \end{pmatrix}, \tag{3.16b}$$

it can be directly verified that

$$M = \hat{U} \hat{L}. \tag{3.16c}$$

Now, writing

$$U\hat{U} = I + E_U, \qquad L\hat{L} = I + E_L,$$

we obtain

$$HM = L(U\hat{U})\hat{L} = I + E_U + E_L + L E_U \hat{L},$$

where

$$E_U = \begin{pmatrix} -(I - AB) & 0 & (I - AB) G M_{\text{red}} \\ 0 & -(I - BA)^T & [(I - BA)^T (K + Q^T \bar{J}) - Q^T Q A^{-1} (I - AB) G] M_{\text{red}} \\ 0 & 0 & -(I - H_{\text{red}} M_{\text{red}}) \end{pmatrix}$$

$$E_L = \begin{pmatrix} 0 & 0 & 0 \\ Q^T Q A^{-1} (I - AB) & 0 & 0 \\ (K^T + \bar{J}^T Q) A^{-1} (I - AB) & G^T A^{-T} (I - BA)^T & 0 \end{pmatrix}.$$

Theorem 1. *Assume:*

(1) For the discretizations of the forward problem, the data locator and the regularization term, there are constants $\gamma_1, \gamma_2, \gamma_3$ and γ_4 such that, for all sufficiently fine grids, $\|A^{-1}G\| \leqslant \gamma_1$, $\|A^{-1}Q\| \leqslant \gamma_2$, $\|Q^T Q\| \leqslant \gamma_3$ and $\|G H_{\text{red}}^{-1}\| \leqslant \gamma_4$.

(2) The approximate inverses satisfy $\|I - H_{\text{red}} M_{\text{red}}\| \leqslant \alpha_1$ and $\|I - AB\| \leqslant \alpha_2$ for constants $\alpha_1 < 1$ and $\alpha_2 < 1$.

Then there are constants K_1 and K_2 depending only on the problem such that for all sufficiently fine grids

$$\|I - MH\| \leqslant K_1 \alpha_1 + K_2 \alpha_2.$$

The first of the theorem's two assumptions typically holds under mild conditions on Ω, since A is a discretization of an elliptic operator (see, e.g., [15, 47])[6]. The second assumption specifies a basic requirement of the approximate inverses B and M_{red}. Note that similar bounds hold for $\|I - M_{\text{red}} H_{\text{red}}\|$ and $\|I - BA\|$.

Proof of theorem 1. From the assumptions it follows that

$$\|BG\| \leqslant \|BA\|\|A^{-1}G\| \leqslant (1+\alpha_2)\gamma_1 \leqslant 2\gamma_1,$$
$$\|BQ\| \leqslant \|BA\|\|A^{-1}Q\| \leqslant (1+\alpha_2)\gamma_2 \leqslant 2\gamma_2,$$
$$\|GM_{\text{red}}\| \leqslant \|GH_{\text{red}}^{-1}\|\|H_{\text{red}}M_{\text{red}}\| \leqslant \gamma_4(1+\alpha_1) \leqslant 2\gamma_4.$$

We then obtain

$$\|E_L\| \leqslant \alpha_2 C_1$$
$$\|E_U\| \leqslant \alpha_2 C_2 + \alpha_1 C_3$$
$$\|LE_U\hat{L}\| \leqslant C_4\,(\alpha_2 C_2 + \alpha_1 C_3)$$

where the constants C_i, $i = 1{:}4$, depend on the constants γ_j, $j = 1{:}4$, in an obvious manner. The claimed bound follows directly from the expression previously derived for HM, where K_1 and K_2 depend on the constants C_i, $i = 1{:}4$. □

The constants K_1 and K_2 depend on the problem and are not under our control. However, by choosing B and M_{red} judiciously we can affect α_2 and α_1. For sufficiently small values of these constants,

$$c := K_1\alpha_1 + K_2\alpha_2 < 1.$$

Then, the condition number of MH is bounded by $\frac{1+c}{1-c}$ and rapid convergence of the preconditioned Krylov method may be expected. Note that the convergence rate is expected to be independent of the grid (for all sufficiently fine grids) if α_1 and α_2 are. On the other hand it must be acknowledged that to obtain $c < 1$ the requirements on α_1 and α_2 may be more stringent than what is necessary to obtain satisfactory results in practice.

4. Numerical experiments

For our numerical experiments we use the regularization (1.9) discretizing (1.4*a*), and apply the Gauss–Newton method. This combination of choices is not only the simplest (thus allowing us to avoid various incidental issues), it is also the one which has been most popular amongst practitioners (although it disallows discontinuities in the model) and for which the reduced Hessian method has the most solid track record. For the reduced Hessian method we subsequently use the PCG algorithm.

We apply our proposed method to two typical model problems, the div–grad problem described in (1.1) and the magnetostatic problem. In both cases we use the following setup.

For the regularization we assume a smooth model and choose

$$W = h^{\frac{3}{2}} \begin{pmatrix} -D^T \\ 10^{-3} I \end{pmatrix},$$

that is, the scaled discretization of the operator $\begin{pmatrix} \nabla \\ 10^{-3} I \end{pmatrix}$ with respect to cell centres (see [6]).

For m_{ref} we choose the *constant* $\bar{m}$ which minimizes

$$\|QA^{-1}(\bar{m})q - b\|.$$

[6] Although, both $\|A\|$ and $\|A^{-1}\|$ may be unbounded: recall that A^{-1} corresponds to a discretization of the Green function in more than one space dimension see [33].

As an initial iterate for the nonlinear iteration process we set $m = m_{\text{ref}}$ (i.e., $\mathsf{R}(m - m_{\text{ref}}) = 0$ at first), u as the solution of the corresponding forward problem, and $\lambda = 0$.

4.1. The div–grad inverse problem

To apply our methods to the problem in (1.1), we first write the PDE (1.1a) as a first-order system using fluxes $\boldsymbol{J} = \mathrm{e}^m \nabla u$, and discretize it on a uniform grid in the interval $[-1, 1]$ with grid spacing h. An intermediate staggered grid is used with the fluxes $\boldsymbol{J}$ placed on cell faces while u and m are at cell centres—see [6] for details. This yields the discrete system

$$\begin{aligned} D\,\boldsymbol{J} &= q, \\ \boldsymbol{J} + S(m) D^T u &= 0, \end{aligned}$$

where D is a discretization of the *div* operator plus the boundary conditions (1.1b) and $S(m)$ is a diagonal matrix with harmonic averages of e^m on its diagonal. For a source and a sink we choose

$$q(x, y, z) = \delta(x + 0.4)\,\delta(y + 0.4)\,\delta(z + 0.4) - \delta(x - 0.4)\,\delta(y - 0.4)\,\delta(z - 0.4).$$

Eliminating $\boldsymbol{J}$ we obtain

$$A(m)u = DS(m)D^T u = -q \tag{4.17}$$

and the null space of the operator is handled by requiring a discrete version of (1.1c) to hold.

In order to synthesize data for our numerical experiment we assume that a 'true model' m is given by

$$\begin{aligned} m(x, y, z) = 0.25\,[&3(1 - 9x^2 \exp(-9x^2) - (3y + 1)^2 - 3(3z + 1)^2) \\ &-10(3x/5 - 27x^3 - (3y)^5 - (3z)^5) \exp(-9x^2 - 9y^2 - 27z^2) \\ &-1/3 \exp(-(3x + 1)^2 - 9y^2 - 27z^2)]. \end{aligned}$$

The data are then calculated as $u(m)$-values on a 6^3-grid uniformly spaced in the interval $[-0.6, 0.6]$, corrupted by 1% noise.

The matrix $G = \frac{\partial A(m)u}{\partial m}$ is easily obtained by differentiating the product $S(m)v$ (for an arbitrary vector v) with respect to m. Using the finite volume expressions [6] and (4.17), it is easy to show that G is given by

$$G = DS^2 B_d \operatorname{diag}(\mathrm{e}^{-m}), \qquad B_d = \frac{\partial [S(m)v]}{\partial m}.$$

Note that B_d is a bidiagonal matrix.

For the approximate inverses B and M_{red} we experiment with two choices, as in [3]. We first use incomplete LU-decomposition with threshold (ILU(t)) of A and $W^T W$ to generate approximate inverses. Although practically efficient, the condition number of the preconditioned system deteriorates with this method as the grid is refined. Second, we also use for B a one multigrid W-cycle applied to A, and for M_{red} a one multigrid W-cycle applied to $W^T W$. The multigrid method implemented here is vertex based [50] with one double sweep of symmetric Gauss–Seidel for pre-smoothing and one for post-smoothing. Bilinear interpolation is used for the prolongation, its adjoint for restriction, and the Galerkin coarse grid operator is employed. This ensures grid independence of our preconditioner, and the convergence of PSQMR up to a given tolerance is thus expected in a fixed number of iterations independent of the grid size. Unfortunately, unless we use a better approximation for the reduced Hessian, there is no independence of β, and the number of iterations is expected to rise as β decreases with any of these method variants.

Table 1. Numerical experiment 1: iteration and flopcounts for the reduced Hessian andthe KKT solvers when the linear system is solved accurately.

			PSQMR		PCG	
β	Grid	Preconditioner	Itns	Work	Itns	Work
10^{-2}	9^3	ILU	15	3.5e7	13	9.2e7
	17^3		29	2.3e8	14	1.0e9
	33^3		49	2.8e9	13	3.2e10
	65^3		75	3.1e10	14	4.5e11
10^{-1}	33^3		34	1.7e9	7	1.9e10
10^{-3}			97	5.0e9	23	6.0e10
10^{-4}			211	1.1e10	51	1.3e11
10^{-2}	9^3	MG	22	5.0e8	13	1.6e9
	17^3		24	2.6e9	14	8.1e9
	33^3		24	1.4e10	13	4.7e10
	65^3		24	6.8e10	14	1.9e11
10^{-1}	33^3		11	6.8e9	7	2.1e10
10^{-3}			38	2.4e10	23	7.3e10
10^{-4}			78	4.8e10	51	1.6e11

4.1.1. Experiment 1—solving the linear system accurately. Here we solve the linear system which arises from the first nonlinear constrained Gauss–Newton step. The iterative methods used for the linear system are PCG for the reduced Hessian and PSQMR for (1.10), as described in section 3. These iterative processes are stopped when the residual is below 10^{-5}.

In table 1 we collect counts of iterations ('itns') for various values of grid size, β and preconditioner. Substantially more work may be involved in each PCG iteration than in each PSQMR iteration, even though the same types of matrix–vector products are utilized in both, because one forward problem and one adjoint problem must be solved relatively accurately when using PCG. We therefore display also MATLAB flop counts ('work') as a measure for actual total work.

The ILU(t) preconditioner [43] (denoted 'ILU') is used with threshold 10^{-2}. The multigrid W-cycle (denoted 'MG') is used as discussed above. The same preconditioners are used for both PCG and PSQMR, and the former's inner iteration is stopped when the residual is below 10^{-7}.

The results in table 1 indicate that the number of PSQMR iterations is larger than the number of PCG iterations for comparable runs; however, the overall work estimate for PSQMR is substantially lower than the work estimate for PCG. This is especially true if the ILU preconditioner is used, because more matrix–vector products are needed in order to actually solve the forward problem.

Comparing results for the same value of β when using ILU, both methods require more forward modelling products as the grid is refined[7]. No such dependence on the grid is observed when using a multigrid preconditioner, which agrees with the bounds obtained in section 3. The increase in the number of ILU iterations as the grid is refined is common to both the PSQMR method and the solution of the forward problem. The results in this case show a substantial improvement of PSQMR over PCG.

For smaller values of β the reduced problem becomes more ill-conditioned and the spread of its eigenvalues expands [34]. Both methods deteriorate then, but the cost ratio between

[7] Although the number of PCG iterations remains apparently constant, evaluating matrix–vector products for the reduced Hessian requires more inner iterations for computing $w = Jv$ and $J^T w$ as the grid is refined.

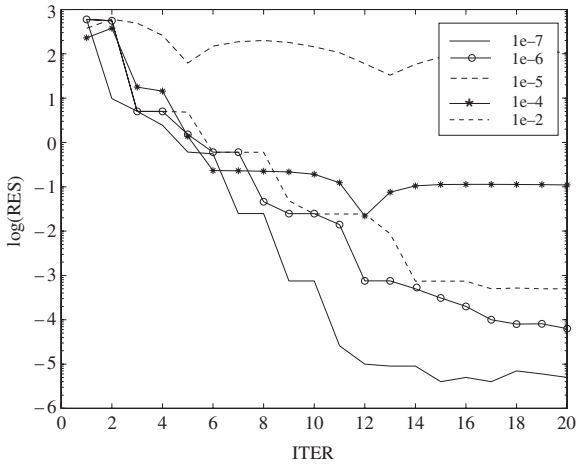

Figure 1. Convergence of the CG method for different tolerance values on the inner solver.

them remains roughly the same.

The stringent tolerance on the inner iteration of the PCG method is central to its relative slowness and results from the necessity to maintain conjugacy of computed directions in the CG algorithm. In figure 1 we plot the log of the residual norm as a function of PCG iterations for several, more relaxed values of this tolerance, employing a 17^3-grid with $\beta = 10^{-2}$. From this figure it is clear that the convergence of PCG is severely effected if the forward problem is not solved well.

4.1.2. Experiment 2—solving the linear system within an inexact Newton-type method. The calculations in section 4.1.1 are performed with a stringent tolerance for the linear system. But within the nonlinear iteration framework for (1.6), the linear KKT system at each iteration may be solved only approximately [37, 41]. Whereas the same procedure may be envisioned utilizing a PCG algorithm for the reduced Hessian approach, there are potential problems of reliability when the directions in a CG algorithm are 'not very conjugate', as mentioned earlier. Therefore, the solution of the forward problem in each PCG iteration must be computed relatively accurately, even if the solution to the whole reduced Hessian system is needed only to a rough accuracy. Thus, we expect our proposed approach to become even more advantageous when employing an inexact-type method.

While there are many variants of Newton-type methods for the nonlinear inverse problem [31], we remain focussed on the Gauss–Newton approximation, where we solve the linear equations at each nonlinear iteration to a rough (relative) accuracy of 10^{-2}. But now we have a few variants of the Gauss–Newton method:

(1) Using the traditional, unconstrained formulation (1.3) we apply an inexact Gauss–Newton method as in [6], where the reduced system for δm is solved to the above liberal tolerance at each iteration. The PCG algorithm is used as before, and the resulting method variant is denoted 'UGN–PCG'. Here, we solve the forward problem in each matrix–vector product to accuracy of 10^{-6}.

(2) If the linear systems at each nonlinear iteration were to be solved 'exactly' then the same Gauss–Newton method could be implemented by applying PSQMR to (1.10) and updating

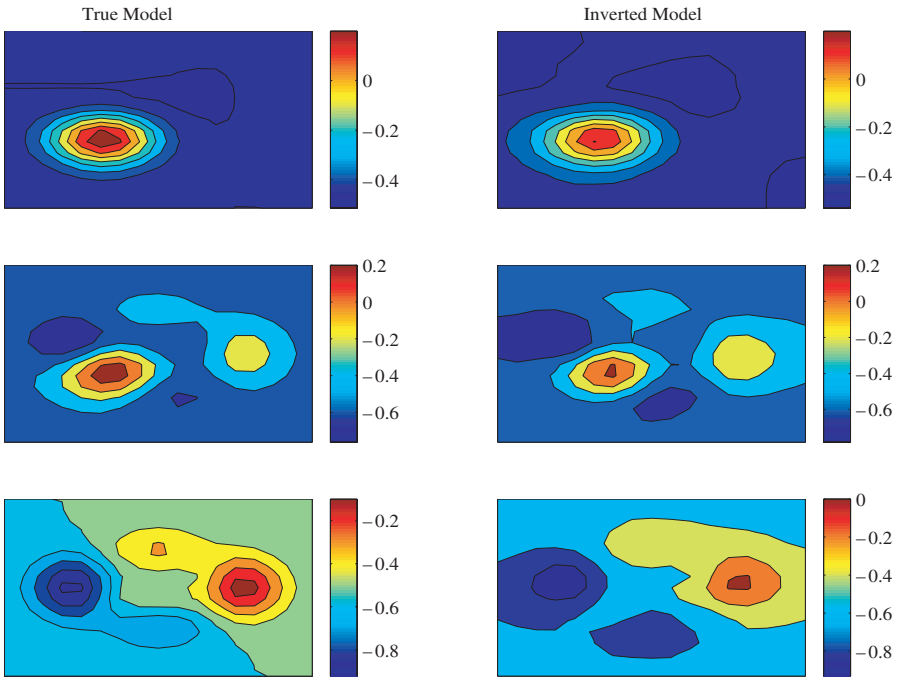

Figure 2. Slices in the 'true model' and the reconstructed model (based on noisy data) for $z = 0.3, 0.55, 0.75$.

for m, followed by updating u and λ to satisfy (1.8c) and (1.8a), respectively [31][8]. But now we solve (1.10) only approximately, so a different variant for the nonlinear solution method is obtained. The idea is reminiscent of post-stabilization [4, 5] and secondary correction in SQP methods [41], and it has advantages both in hastening convergence and in making the solution feasible before reaching optimality; thus this variant adds robustness to the overall method. Denote it by 'UGN–PSQMR'.

(3) A third variant is obtained if, having calculated an update direction δm, δu, $\delta\lambda$ as in the previous variant UGN–PSQMR, we update the entire unknown vector simultaneously. Thus, a line search is used to ensure global convergence with a merit function which weighs objective as well as infeasibility (for details see [18, 31, 41]). This variant corresponds to a constrained Gauss–Newton method when the tolerances are tightened [31]. Denote it by 'CGN–PSQMR'.

For the experiment reported below we use $\beta = 7 \times 10^{-3}$, a value obtained upon employing the discrepancy principle based on our knowledge of the noise level and statistics (see [6] and references therein). The stopping criterion is when the norm of the gradient gets below 10^{-5}. We run the experiment on a 33^3-grid and use both ILU and MG preconditioners as before. Results (which are independent of choice of preconditioner) are plotted in figure 2.

Table 2 records the total number of Gauss–Newton steps needed, the total number of linear iterations needed and the total work in flops for the solution of the nonlinear problem. These results indicate that combining our solver with an inexact Newton-type iteration can be very

[8] Note that the new $\lambda \leftarrow \lambda + \delta\lambda$ can be found directly, as is customary in SQP methods [41].

Table 2. Numerical experiment 2: iteration counts and workestimates for Gauss–Newton variants on a 33^3-grid, with$\beta = 7 \times 10^{-3}$, where the linear systems are solvedinaccurately.

	ILU			MG		
Method	Nonlin-itns	Lin-itns	Work	Nonlin-itns	Lin-itns	Work
UGN–PCG	8	59	1.5e11	8	59	2.8e11
UGN–PSQMR	7	129	7.6e9	7	68	4.5e10
CGN–PSQMR	7	136	7.8e9	7	71	4.6e10

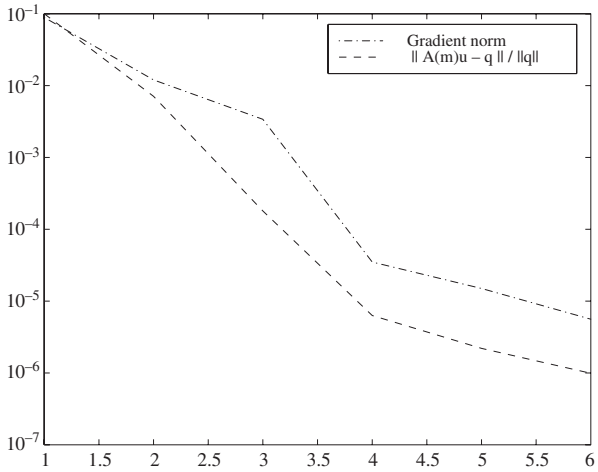

Figure 3. Norms of relative gradient and residual of the constraint as functions of the Gauss–Newton iteration for CGN–PSQMR.

powerful. An order of magnitude efficiency gain is realized over the traditional inexact Gauss–Newton method. In fact, since the number of iterations required for the solution of the forward problem alone using ILU on a 33^3-grid is typically 30–40, it is seen that we manage to solve the entire nonlinear inverse problem in a total cost which is not much larger than that needed for solving a few forward problems.

Further, for CGN–PSQMR the solution to the forward problem is never calculated very accurately; yet, as can be seen in figure 3 we advance towards feasibility (i.e., solving the forward problem more accurately) in tandem as we advance towards optimality (solving the entire constrained optimization problem). Note that the variant CGN–PSQMR is implemented within a procedure which guarantees global convergence under certain conditions, whereas the variant UGN–PSQMR is not.

4.2. The magnetostatic inverse problem

The magnetostatic problem is governed by Maxwell's equations in the steady state,

$$\nabla \times \boldsymbol{E} = \boldsymbol{0}, \tag{4.18a}$$

$$\nabla \times \boldsymbol{H} - \sigma \boldsymbol{E} = \boldsymbol{s}, \tag{4.18b}$$

$$\nabla \cdot (\mu \boldsymbol{H}) = 0. \tag{4.18c}$$

Here, $\boldsymbol{H}$ is the magnetic field, $\boldsymbol{E}$ the electric field, $\sigma \geqslant 0$ the conductivity, $\mu > 0$ the

permeability, and $\boldsymbol{s}$ is a source. The system is closed under the boundary conditions

$$\boldsymbol{H} \times \boldsymbol{n}|_{\partial\Omega} = 0. \tag{4.18d}$$

The inverse problem under consideration is to recover $\sigma(\boldsymbol{x})$ based on measurement data on $\boldsymbol{H}$ at certain locations.

We follow [28–30] in reformulating and discretizing this problem. For this, note that (4.18) corresponds to the frequency domain formulation for a zero frequency. This allows us to avoid certain complications present in the full Maxwell equations which are not central to the theme of this paper.

Defining

$$\mu\boldsymbol{H} = \nabla \times \boldsymbol{\psi}, \qquad \nabla \cdot \boldsymbol{\psi} = 0, \qquad \boldsymbol{E} = \nabla\phi, \qquad \sigma = \mathrm{e}^m, \tag{4.19}$$

applying $\nabla\cdot$ to (4.18b), stabilizing, and substituting (4.19) into (4.18) we obtain the elliptic PDE system

$$\nabla \times \mu^{-1}\nabla \times \boldsymbol{\psi} - \nabla\mu^{-1}\nabla \cdot \boldsymbol{\psi} - \mathrm{e}^m\nabla\phi = \boldsymbol{s}, \tag{4.20a}$$

$$-\nabla \cdot (\mathrm{e}^m\nabla\phi) = \nabla \cdot \boldsymbol{s}, \tag{4.20b}$$

subject to boundary conditions

$$(\nabla \times \boldsymbol{\psi}) \times \boldsymbol{n}|_{\partial\Omega} = \boldsymbol{0}, \tag{4.20c}$$

$$\boldsymbol{\psi} \cdot \boldsymbol{n}\left|_{\partial\Omega} = \frac{\partial\phi}{\partial n}\right|_{\partial\Omega} = 0, \tag{4.20d}$$

$$\int_\Omega \phi \,\mathrm{d}V = \int_\Omega (\mu^{-1}\nabla \cdot \boldsymbol{\psi})\,\mathrm{d}V = 0. \tag{4.20e}$$

To apply the methods presented in this paper to the magnetostatic problem, we must first discretize the PDE system (4.20). For this we use the finite volume method described in [29, 30], employing a uniform staggered grid with the components of $\boldsymbol{\psi}$ placed on cell faces and ϕ and m at cell centres. This yields the discrete system

$$A(m)u = \begin{pmatrix} C^T M^{-1} C + D^T M_c^{-1} D & S(m)D^T \\ 0 & DS(m)D^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\psi} \\ \phi \end{pmatrix} = \begin{pmatrix} \boldsymbol{s} \\ D\boldsymbol{s} \end{pmatrix} = q$$

where D and C are discretizations of the ***div*** and ***curl*** operators plus the boundary conditions (4.20c) and (4.20d), $S(m)$ is a diagonal matrix with harmonic averages of e^m at cell faces on its diagonal, and M and M_c are diagonal matrices whose elements are averages of μ at cell edges and cell centres.

In this work we assume that μ is constant, $\mu = \mu_0 = 4\pi \times 10^{-7}$. Thus, $C^T M^{-1} C + D^T M_c^{-1} D$ corresponds to the staggered, but component-wise usual, discretization of the vector Laplacian times μ_0^{-1}. We also choose $\Omega = [-100, 100]^3$ and employ a grid with 32^3 cells.

For a source and a sink we choose the magnetometric resistivity (MMR) setting of [39, 40], which is an electric line source

$$\boldsymbol{s}(x, y, z) = \begin{pmatrix} (H(x - x_2) - H(x - x_1))\delta(y)\,\delta(z) \\ 0 \\ 0 \end{pmatrix}$$

where H is the Heaviside function and $x_2 = -x_1 = 40$. Thus, $\nabla \cdot \boldsymbol{s}$ consists of two delta functions.

In order to synthesize data for our numerical experiment we add 1% noise to a 'true model' m given by

$$\begin{aligned} m(x, y, z) = -2 \,+\, & \exp(-0.002(x + 25)^2 - 0.002(y + 25)^2 - 0.002(z - 25)^2) \\ & +2\exp(-0.002(x - 25)^2 - 0.002(y - 25)^2 - 0.002(z - 40)^2). \end{aligned}$$

Table 3. Numerical experiment 3: iteration and flop counts forthe reduced Hessian and the KKT solvers when the linear systemis solved accurately.

	PSQMR		PCG	
β	Itns	Work	Itns	Work
10^{-1}	102	6.2e9	7	4.3e10
10^{-2}	104	6.2e9	9	5.4e10
10^{-3}	116	6.9e9	19	1.1e11
10^{-4}	158	9.5e9	28	1.4e11
10^{-5}	236	1.4e10	43	2.9e11

Table 4. Numerical experiment 4: iteration countsand work estimates for Gauss–Newton variantson a 33^3-grid, with $\beta = 5 \times 10^{-3}$,where the linear systems are solved inaccurately.

	SSOR		
Method	Nonlin-itns	Lin-itns	Work
UGN–PCG	3	25	4.1e11
UGN–PSQMR	3	151	1.6e10
CGN–PSQMR	3	157	1.7e10

Upon solving the discretized forward problem, the magnetic field is calculated by taking the discrete curl of[9] ψ. The predicted field values at data locations are then given by $\tilde{Q}C\psi$, where $\tilde{Q}$ is an interpolation matrix chosen here to interpolate the magnetic field on a 30^2-grid uniformly spaced on the plane $z = 0$, $-50 \leqslant x, y \leqslant 50$.

In the experiments reported below we have used one iteration of SSOR [43] as a preconditioner for PCG and PSQMR; this requires less memory than the preconditioners used in section 4.1. The PCG inner iteration (controlling conjugacy) is stopped when the residual is below 10^{-7}.

4.2.1. Experiment 1—solving the linear system accurately. Again we solve the linear system which arises from the first nonlinear constrained Gauss–Newton step and compare the PCG for the reduced Hessian with PSQMR. These iterative processes are stopped when the residual is below 10^{-5}. Table 3 records computational effort estimates for various values of β.

As before, the results in table 3 indicate that the number of PSQMR iterations is larger than the number of PCG iterations for comparable runs; however, the overall work estimate for PSQMR is substantially lower than the work estimate for PCG. In this case we have found that the solution of the forward problem (using the SSOR preconditioner) requires roughly 80–100 iterations. Therefore, there is a substantial cost reduction when solving the whole KKT system. The overall reduction in work is much more substantial than in section 4.1.1; one reason is that the forward problem in this case requires more computational effort to solve, as compared with the problem (1.1) in section 4.1.

4.2.2. Experiment 2—solving the linear system within an inexact Newton-type method. We now repeat the experiments in section 4.1.2 (i.e. using the same grid size, tolerances, etc) for the magnetostatic problem. For the same noise level of 1%, the discrepancy principle yields the value $\beta = 5 \times 10^{-3}$ here. Results analogous to those of table 2 are recorded in table 4.

These results again demonstrate that the combination of our solver with an inexact Newton-type iteration can be very powerful.

[9] It can be shown [29] that the discretization yields $\boldsymbol{H}$ to second-order accuracy.

5. Conclusions and further discussion

In this paper we have developed a preconditioned, symmetric QMR algorithm for the solution of large distributed parameter estimation problems of the form (1.6). The preconditioner is obtained by solving the reduced Hessian system (2.12) approximately. We have analysed the performance of the proposed solver and demonstrated that it can yield significant efficiency gains in practice by applying it to instances of the dc-resistivity and the magnetostatic problems.

Computations were also performed for Maxwell's equations in 3D for moderate frequencies, realizing rather substantial efficiency gains for this problem as well [32].

Let us emphasize that we are not claiming to present 'the best method for all problems': The nonlinear iteration process is different when switching from the unconstrained to the constrained formulation, and it is different again when solving the linear equations only to a rough accuracy. Despite the relevant theory, there may in principle be examples (although none such was encountered in the case studies considered here) when our method would not converge while another nonlinear iteration process would. However, the evidence presented here, complemented by [2, 12, 13] and studied against the background of the major expense per iteration associated with PDE systems in 3D, clearly signals that our approach offers significant advances towards the efficient solution of several important practical problems.

Although we have examined the rapid solution of linear systems at each Newton-type iteration for a nonlinear problem (1.6), our proposed method can be important also for linear problems such as those arising when using the Born approximation [1, 51]. In particular, for a given linear problem of the form $H_{\text{red}}m = -p$ where H_{red} and p are given by (2.13), it may be rather advantageous to write it in constrained form and employ PSQMR (cf [31]).

As mentioned before (see especially section 4.1.1, [27] and figure 1), a major obstacle that our method circumvents is the need to retain conjugacy of direction vectors to a high accuracy when using the CG method. We have noted that when the tolerance is relaxed the algorithm may break down, especially when the norm of the right-hand side in (1.10) becomes small. A way to avoid breakdowns is to solve the forward problem more accurately (by a couple of orders of magnitude) than (1.10). Thus, the tolerance for solving the forward problem when forming the approximation for J in (2.12), (2.13) is gradually tightened, but always more stringent than the tolerance for (1.10). We have had a moderate success with this idea [32] (see also [17] for similar ideas).

Another possibility is to replace J in (2.13) by $\bar{J}$ of (3.16a) and maintain conjugacy for the approximated system. Work along this line will be reported in the future.

The possibility of replacing J by an approximation is relevant also for the PSQMR algorithm, of course, although in a different context: our choice for M_{red} in section 4 was based on ignoring $J^T J$ in H_{red}. Whereas this is doubtlessly cheap, it is also guaranteed to fail as $\beta \to 0$. For small values of $\beta > 0$ we may therefore expect to do better with choices such as a bidiagonalization of $\bar{J} = -QBG$, or an automatic sparse approximate inverse construct for approximating J in M_{red} [10]. We have experimented with such variants and found that, whereas they do indeed improve robustness for small values of β, the overall performance for the examples presented in section 4 is not improved by much.

Finally, the necessary conditions (1.8) can be viewed, as mentioned earlier, as a discretization of a strongly coupled PDE system. In [7] we design a multigrid method for such a system as an alternative for the PSQMR method described in this paper. Generally speaking, the present method is more robust and easier to implement, but the multigrid method performs supremely in adequate circumstances.

Acknowledgments

We are grateful to an anonymous referee for pointing out the references [2, 9, 11, 12]. EH was supported in part under NSERC CRD Grant 80357. UMA was supported in part under NSERC Research Grant 84306.

References

[1] Alumbaugh D and Morrison F 1993 Electromagnetic conductivity imaging with an iterative Born inversion *IEEE Trans. Geosci. Remote Sens.* **31** 758–62
[2] Arian E, Battermann A and Sachs E W 1999 Approximation of the Newton step by a defect correction process *Technical Report* TR 99-12, ICASE, NASA Langley, Hampton, VA
[3] Aruliah D and Ascher U 2000 Multigrid preconditioning for time-harmonic Maxwell's equations in 3D *Preprint*
[4] Ascher U 1997 Stabilization of invariants of discretized differential systems *Numer. Algorithms* **14** 1–23
[5] Ascher U, Chin H, Petzold L and Reich S 1995 Stabilization of constrained mechanical systems with daes and invariant manifolds *J. Mech. Struct. Machines* **23** 135–58
[6] Ascher U and Haber E 2001 Grid refinement and scaling for distributed parameter estimation problems *Inverse Problems* **17** 571–90
[7] Ascher U and Haber E 2001 A multigrid method for distributed parameter estimation problems *Preprint*
[8] Barrett R, Berry M, Chan T F, Demmeland J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C and Van der Vorst H 1994 *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (Philadelphia, PA: SIAM)
[9] Battermann A and Heinkenschloss M 1998 Preconditioners for Karush–Kuhn–Tucker matrices arising in the optimal control of distributed systems *Optimal Control of Partial Differential Equations (Vorau, 1997)* (Basle: Birkhauser) pp 15–32
[10] Benzi M and Tuma M 1999 A comparative study of sparse approximate inverse preconditioners *Trans. Appl. Numer. Math.* **30** 305
[11] Biros G and Ghattas O 1999 Parallel Newton–Krylov methods for PDE-constrained optimization *Proc. CS99, (Portland, OR)*
[12] Biros G and Ghattas O 1999 Parallel preconditioners for KKT systems arising in optimal control of viscous incompressible flows *Proc. Parallel CFD (Williamsburg, VA, May)* (Amsterdam: North-Holland)
[13] Biros G and Ghattas O 2000 Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization Parts I, II *Preprints*
[14] Borcea L, Berryman J G and Papanicolaou G C 1996 High-contrast impedance tomography *Inverse Problems* **12** 835–58
[15] Brezzi F and Fortin M 1991 *Mixed and Hybrid Finite Element Methods* (New York: Springer)
[16] Cheney M, Isaacson D and Newell J C 1999 Electrical impedance tomography *SIAM Rev.* **41** 85–101
[17] Dennis J and Walker H 1984 Local improvement theorems *Math. Prog. Stud.* **22** 70–85
[18] Dennis J E and Schnabel R B 1996 *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Philadelphia: SIAM)
[19] Devaney A J 1989 The limited-view problem in diffraction tomography *Inverse Problems* **5** 510–23
[20] Elman H 1999 Preconditioning for the steady-state Navier–Stokes equations with low viscosity *SIAM J. Sci. Comput.* **20** 1299–316
[21] Elman H and Silvester D 1996 Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations *SIAM J. Sci. Comput.* **17** 33–46
[22] Engl H W, Hanke M and Neubauer A 1996 *Regularization of Inverse Problems* (Dordrecht: Kluwer)
[23] Fortin M and Glowinski R 1983 *Augmented Lagrangian Methods: Applications in the Numerical Solution of Boundary-Value Problems* (Amsterdam: North-Holland)
[24] Freund R W and Jarre F 1996 A QMR-based interior-point algorithm for solving linear programs *Math. Prog.* B **76** 183–210
[25] Freund R W and Nachtigal N M 1994 A new Krylov-subspace method for symmetric indefinite linear systems *Proc. 14th IMACS World Congress on Computational and Applied Mathematics (IMACS)* ed W F Ames pp 1253–6
[26] Golub G and Greif C 2000 Techniques for solving general KKT systems *Technical Report* SCCM, Stanford
[27] Golub G and Ye Q 2000 Inexact preconditioned conjugate gradient method with inner-outer iteration *SIAM J. Sci. Comput* **21** 1305–20

[28] Haber E 2000 A mixed finite element method for the solution of the magnetostatic problem in 3D *Comput. Geosci.* **4** 323–6
[29] Haber E and Ascher U 2001 Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients *SIAM J. Sci. Comput.* **22** 1943–61
[30] Haber E, Ascher U, Aruliah D and Oldenburg D 2000 Fast simulation of 3D electromagnetic using potentials *J. Comput. Phys.* **163** 150–71
[31] Haber E, Ascher U and Oldenburg D 2000 On optimization techniques for solving nonlinear inverse problems *Inverse Problems* **16** 1263–80
[32] Haber E, Oldenburg D and Ascher U 2000 Inversion of 3D electromagnetic data—a constrained optimization approach *SEG (Calgary, August 2000)*
[33] Hackbusch W 1985 *Multi-Grid Methods and Applications* (Berlin: Springer)
[34] Hansen P C 1998 *Rank Deficient and Ill-Posed Problems* (Philadelphia, PA: SIAM)
[35] Heinkenschloss M and Vicente L N 1999 Analysis of inexact trust region SQP algorithms *Technical Report* TR 99-18 Rice University
[36] Ito K and Kunisch K 1990 The augmented Lagrangian method for parameter estimation in elliptic systems *SIAM J. Control Optim.* **28** 113–36
[37] Kelley C T 1999 *Iterartive Methods for Optimization* (Philadelphia, PA: SIAM)
[38] Kleinman R E and van den Berg P M 1992 A modified gradient method for two-dimensional problems in tomography *J. Comput. Appl. Math.* **42** 17–35
[39] McGillivray P R 1992 Forward modelling and inversion of dc resistivity and MMR data *PhD Thesis* University of British Columbia
[40] Nabighian M, Oppliger G, Edwards R, Lo B and Cheesman S 1984 Cross-hole magnetometric resistivity (MMR) *Geophysics* **49** 1313–23
[41] Nocedal J and Wright S 1999 *Numerical Optimization* (New York: Springer)
[42] Parker R L 1994 *Geophysical Inverse Theory* (Princeton, NJ: Princeton University Press)
[43] Saad Y 1996 *Iterartive Methods for Sparse Linear Systems* (Boston, MA: PWS Publishing Company)
[44] Shenoy A R, Heinkenschloss M and Cliff E M 1998 Airfoil design by an all-at-once method *Int. J. Comp. Fluid Mechanics* **11** 3–25
[45] Silvester D, Elman H, Kay D and Wathen A 2001 Efficient preconditioning of the linearized Navier–Stokes equations *J. Comput. Appl. Math.* **128** 261–79
[46] Smith N C and Vozoff K 1984 Two-dimensional dc resistivity inversion for dipole dipole data *IEEE Trans. Geosci. Remote Sens.* **22** 21–8 (special issue on electromagnetic methods in applied geophysics)
[47] Strikwerda J C 1989 *Finite Difference Schemes and Partial Differential Equations* (New York: Wadsworth & Brooks/Cole)
[48] Tikhonov A N and Arsenin V Ya 1977 *Methods for Solving Ill-posed Problems* (New York: Wiley)
[49] Vogel C 1999 Sparse matrix computation arising in distributed parameter identification *SIAM J. Matrix Anal. Appl.* **20** 1027–37
[50] Wesseling P 1992 *An Introduction to Multigrid Methods* (New York: Wiley)
[51] Zhdanov M and Fang S 1996 Three-dimensional quasi-linear electromagnetic inversion *Radio Sci.* **31** 741–8