JOURNAL OF COMPUTATIONAL PHYSICS **123,** 330–340 (1996)
ARTICLE NO. 0027

A Preconditioned Krylov Subspace Method for the Solution of Least Squares Problems in Inverse Scattering

KEES VUIK, AGUR G. J. SEVINK, AND GÉRARD C. HERMAN

Faculty of Technical Mathematics and Informatics, Delft University of Technology, Mekelweg 4, NL 2628 CD Delft, The Netherlands

Received June 7, 1994; revised June 26, 1995

We present an iterative method of preconditioned Krylov type for the solution of large least squares problems. We prove that the method is robust and investigate its rate of convergence. For an important application, originating from seismic inverse scattering, we derive a suitable preconditioner using asymptotic theory. Numerical experiments are used to compare the method with other iterative methods. It appears that the preconditioned Krylov method can be much more efficient than CG applied to the normal equations.

1. INTRODUCTION

In this paper we consider the solution of underdetermined least squares problems by iterative methods. Section 2 contains the description of the problem. In Section 3 we give a short survey of existing iterative methods for least squares problems, such as SIRT, ART, and CG. After this survey we present two different variants of a preconditioned Krylov subspace method. The first one is robust, whereas the second one can be considerably faster for underdetermined systems. In Section 4 we present results concerning the convergence of the Krylov methods. Since the eigenvalues of the preconditioned matrix play an important role we also give an algorithm to approximate these eigenvalues. Section 5 contains an application originating from seismic inverse scattering. For this application we give a suitable preconditioner. The convergence results given in Section 4 are compared with the convergence observed in the numerical examples. Finally, we note that for this application the preconditioned Krylov method may be three times faster than the CG method applied to the normal equations.

2. STATEMENT OF THE PROBLEM

We consider the least squares problem: given the matrix $L \in \mathbb{C}^{m\times n}$ and right-hand side $b \in \mathbb{C}^m$, find a solution vector $x \in \mathbb{C}^n$ such that

$$\|Lx - b\|_2 = \min_{y\in\mathbb{C}^n} \|Ly - b\|_2. \tag{1}$$

In this paper we consider an underdetermined system of equations $Lx = b$ so $m < n$. However, the methods given in Section 3 can also be used for overdetermined systems. It is easy to show that x is a solution of (1) if and only if

$$L^*Lx = L^*b. \tag{2}$$

The equations given in (2) are called the normal equations.

The l^2-norm condition number $\kappa_2(L)$ is the ratio of the largest and the smallest singular value (see [11, p. 223]): $\kappa_2(L) = \sigma_{\max}/\sigma_{\min}$.

3. ITERATIVE METHODS FOR LEAST SQUARES PROBLEMS

Before we start to describe our iterative solution method we give a short survey of existing methods. For the theory of least squares problems we refer to [11, Chap. 5]. Direct methods to solve underdetermined linear equations are given in [5].

3.1. *Survey of Well-Known Iterative Methods*

The following iterative methods are known:

SIRT. The SIRT (simultaneous iterative reconstruction technique) method is described in [10]. It can be shown that SIRT is equivalent to Richardson iteration applied to the system (see [19])

$$C^{1/2}L^*RLC^{1/2}y = C^{1/2}L^*Rb \quad x = C^{-1/2}y.$$

In these equations R is a row-scaling matrix and C is a column-scaling matrix.

ART. The ART (algebraic reconstruction technique) method is described in [12]. ART is equivalent to SOR (successive over relaxation) applied to the system [3]

$$LL^*y = b, \quad \text{where } x = L^*y.$$

0021-9991/96 $18.00

CG. A common approach to solve least squares problems is to apply CG (conjugate gradients) to the normal equations

$$L^*Lx = L^*b.$$

This method is called CGNR and is already proposed in [13]. Drawbacks of this method are: it can suffer from rounding errors and a slow rate of convergence. Both phenomena depend on $\kappa_2(L)^2$ which is in general very large. More stable variants, with respect to rounding errors, are given in [3, 16]. In [3], also, preconditioned variants of CGNR are considered. A comparable method CGNE is proposed by [6], which applies CG to the equations

$$LL^*y = b, \quad \text{where } x = L^*y.$$

Comparisons of SIRT, ART, and CG-like methods are given in [19, 20, 8]. For the tomographic problems considered in [19, 20] the CG method has a better rate of convergence than the SIRT method. The comparison given in [8] leads to the conclusion that ART is only more efficient than CG for overdetermined systems, where $m \gg n$.

3.2. *Preconditioned Krylov Subspace Methods*

For square linear systems with a nonsymmetric coefficient matrix $A \in \mathbb{R}^{n\times n}$, it is well known that Krylov subspace methods based on the matrix A have in general a much better convergence behaviour than the CGNR and CGNE methods. One reason for this is the fact that CGNR and CGNE are based on the Krylov subspace

$$K_k(A^TA, A^Tr_0) = \text{span}\{A^Tr_0, (A^TA)A^Tr_0, ..., (A^TA)^{k-1}A^Tr_0\},$$

whereas the nonsymmetric Krylov methods are based on

$$K_k(A, r_0) = \text{span}\{r_0, Ar_0, ..., (A)^{k-1}r_0\}.$$

This motivates us to look for Krylov methods based on $K_k(L, r_0)$ for the solution of the least squares problems given by (1). However, it is impossible to form $K_k(L, r_0)$ if $n \neq m$. The vector $L(Lr_0)$ is not defined because $L \in \mathbb{C}^{m\times n}$ and $Lr_0 \in \mathbb{C}^m$. Our idea to circumvent this is to construct a preconditioner $T \in \mathbb{C}^{n\times m}$ such that T is an approximation of the generalized inverse of L. Note that there are two reasons to use a preconditioner T: first to obtain a square matrix TL and second to have fast convergence.

As usual, as for square matrices, the difficulty is to find a "good" approximation T. An important application, where it is possible to find such a T, is presented in [18] and summarized in Section 5. Note that for a given $T \in \mathbb{C}^{n\times m}$ the product matrix TL is in $\mathbb{C}^{n\times n}$. So the Krylov subspace $K_k(TL, Tr_0)$ is defined and equal to span$\{Tr_0, (TL)Tr_0, ..., (TL)^{k-1}Tr_0\}$. This space is equivalent to the Krylov subspace associated with the left-preconditioned system: $TLx = Tb$. We look for a solution $x_k \in x_0 + K_k(TL, Tr_0)$ such that the residual $r_k = b - Lx_k$ satisfies a certain optimality property.

In our method we choose $x_k \in x_0 + K_k(TL, Tr_0)$ such that

$$\|b - Lx_k\|_2 = \min_{\xi \in x_0 + K_k(TL, Tr_0)} \|b - L\xi\|_2. \tag{3}$$

The algorithm to obtain x_k is given by (compare GCR given in [9]) the following.

ALGORITHM 1.

select x_0, *eps*;
$r_0 = b - Lx_0$, $k = 0$;
while $\|r_k\|_2 > eps$ do
  $k := k + 1$, $u_k^{(1)} = Tr_{k-1}$, $c_k^{(1)} = Lu_k^{(1)}$;
  {c_k is orthogonalized with the modified Gram–Schmidt method}
  for $i = 1, ..., k - 1$ do
    $\alpha_i = c_i^* c_k^{(i)}$;
    $c_k^{(i+1)} = c_k^{(i)} - \alpha_i c_i$, $u_k^{(i+1)} = u_k^{(i)} - \alpha_i u_i$;
  endfor
  {The norm of c_k is made equal to one. Note that $c_k = Lu_k$}
  $c_k = c_k^{(k)}/\|c_k^{(k)}\|_2$, $u_k = u_k^{(k)}/\|c_k^{(k)}\|_2$;
  {The residual is made orthogonal to span $\{c_1, ..., c_k\}$ and x_k is changed accordingly}
  $x_k = x_{k-1} + u_k c_k^* r_{k-1}$;
  $r_k = r_{k-1} - c_k c_k^* r_{k-1}$;
endwhile

Remarks. 1. When GCR or GMRES [17] is applied to $TLx = Tb$ then $\|Tb - TLx_k\|_2$ instead of $\|b - Lx_k\|_2$ is minimized. This may lead to wrong results.

2. The vectors $u_k^{(i)}$ and $c_k^{(i)}$ are only used to describe the algorithm. In a computer implementation they can be replaced by u_k and c_k.

3. The vectors u_k are called search directions. The companion vectors $c_1, ..., c_k$ form an orthonormal basis for the Krylov subspace $K_k(LT, LTr_0)$. Both vector sequences should be stored in memory. If many iterations are needed, then the memory requirements can be bounded by restarting or truncating the algorithm. In our application the number of iterations is small, so all vectors could be kept in memory.

4. The residual r_k is perpendicular to $K_k(LT, LTr_0)$. Note that the residual r_k is obtained by updating r_{k-1}. Due to rounding errors it is possible that r_k and $b - Lx_k$ become different. For this reason it is a good idea to compare the

updated residual r_k and $b - Lx_k$ after every 10 or 20 iterations. If the difference is too large one can use the following strategies:

(a) restart the algorithm,

(b) calculate $r_k = b - Lx_k$, and change x_k and r_k as follows:

for $i = 1, ..., k$ do
$x_k = x_k + u_i c_i^* r_k; r_k = r_k - c_i c_i^* r_k;$
endfor

and continue the algorithm.

In the following theorem we show that the calculated x_k satisfies (3).

THEOREM 3.1. *If $\|c_i^{(i)}\|_2 \neq 0$ for every $i \leq k$, then the approximation x_k obtained from Algorithm* 1 *satisfies*

$$\|b - Lx_k\|_2 = \min_{\xi \in x_0 + \text{span}\{u_1, ..., u_k\}} \|b - L\xi\|_2. \tag{4}$$

Proof. It is clear from $x_k = x_{k-1} + u_k c_k^* r_{k-1}$ that $x_k \in x_0 + \text{span}\{u_1, ..., u_k\}$. Using the equations $r_k = b - Lx_k$ and $c_i = Lu_i$, (4) is rewritten

$$\|r_k\|_2 = \min_{\eta \in \text{span}\{c_1, ..., c_k\}} \|r_0 - \eta\|_2. \tag{5}$$

It is well known that (5) holds if and only if

$$r_k \perp \text{span}\{c_1, ..., c_k\} \tag{6}$$

We prove this relation by induction. Since $r_1 = r_0 - c_1 c_1^* r_0$, relation (6) holds for $k = 1$. Suppose that (6) holds for $i \leq k - 1$. The new residual is formed by $r_k = r_{k-1} - c_k c_k^* r_{k-1}$. Note that r_{k-1} and c_k are perpendicular to $\text{span}\{c_1, ..., c_{k-1}\}$, the first by the induction hypothesis and the second by construction. This together with $c_k^* r_k = c_k^* r_{k-1} - c_k^* c_k c_k^* r_{k-1} = 0$ implies that r_k is perpendicular to $\text{span}\{c_1, ..., c_k\}$, which proves the theorem. ∎

Remarks. 1. Note that this theorem is valid for every choice of $u_k^{(1)}$. If $u_k^{(1)} = Tr_{k-1}$ is used (as in Algorithm 1) it follows that x_k satisfies (3).

2. As long as $\|c_k^{(k)}\|_2 \neq 0$ Algorithm 1 converges and the quantity $\|r_k\|_2$ decreases monotonically. Furthermore, relation (6), together with the fact that $r_k \in \mathbb{C}^m$, imply that $\|r_m\|_2 = 0$.

The case $c_k^{(1)^*} r_{k-1} = 0$ will lead to a stagnation of r at the kth iteration and a break down at the $(k + 1)$th iteration, because $LTr_k \in \text{span}\{c_1, ..., c_k\}$, so $c_{k+1}^{(k+1)} = 0$, and $c_{k+1} = c_{k+1}^{(k+1)} / \|c_{k+1}^{(k+1)}\|_2$ is not defined. In order to circumvent breakdown we define the search direction $u_k^{(1)}$ as follows.

DEFINITION 1. If $(LTr_{k-1})^* r_{k-1} \neq 0$ take $u_k^{(1)} = Tr_{k-1}$; else take $u_k^{(1)} = L^* r_{k-1}$.

THEOREM 3.2. *When we use $u_k^{(1)}$ as defined in Definition* 1, *the proposed method has no breakdown.*

Proof. Using $u_k^{(1)} = Tr_{k-1}$ there are two possibilities: $c_k^{(1)^*} r_{k-1} \neq 0$ and $c_k^{(1)^*} r_{k-1} = 0$. If $c_k^{(1)^*} r_{k-1} \neq 0$ it follows that $c_k^{(1)}$ is not an element of $\text{span}\{c_1, ..., c_{k-1}\}$, since $r_{k-1} \perp \text{span}\{c_1, ..., c_{k-1}\}$. So $c_k^{(k)} \neq 0$ and thus u_k and c_k exist. If $c_k^{(1)^*} r_{k-1} = 0$ the original search direction $u_k^{(1)} = Tr_{k-1}$ is replaced by $u_k^{(1)} = L^* r_{k-1}$. Again two different cases occur:

$\|L^* r_{k-1}\|_2 \neq 0$. In this case $c_k^{(1)^*} r_{k-1} = (LL^* r_{k-1})^* r_{k-1} = \|L^* r_{k-1}\|_2^2 \neq 0$, which implies that u_k and c_k exist (compare the first part of the proof).

$\|L^* r_{k-1}\|_2 = 0$. In this case $L^* r_{k-1} = L^*(b - Lx_{k-1}) = 0$. This implies that x_{k-1} is a solution of the normal equations (2). Thus x_{k-1} is a solution of the least squares problem (1). This is called a lucky breakdown. ∎

Remark. Theorem 3.1 has been proved for every choice of $u_k^{(1)}$, so it also holds for the choice of $u_k^{(1)}$ given in Definition 1. Since for this choice $\|c_k^{(k)}\|_2 \neq 0$ (unless $\|r_{k-1}\|_2 = 0$), the proposed method is convergent.

In our application it never happened that the choice $u_k^{(1)} = Tr_{k-1}$ leads to $c_k^{(1)^*} r_{k-1} = 0$. So the preconditioner T is the same in every iteration. In such a case it is better to use Algorithm 2 given below.

ALGORITHM 2. Apply Algorithm 1 to the right preconditioned system

$$LTy = b. \tag{7}$$

If the norm of the residual $\|b - LTy_k\|_2$ is small enough, form $x_k = Ty_k$, which is an approximation of the solution of the least squares problem (1).

Remarks. 1. Algorithm 2 gives the same results as GCR (or GMRES) applied to (7). In principle all Krylov subspace methods for square complex nonsymmetric matrices can be used (for a recent survey see [2]).

2. The iterate y_k used in Algorithm 2 is an element of the Krylov subspace $K_k(LT, r_0)$.

3. The main difference between both algorithms is that $u_k^{(1)} = Tr_{k-1}$ and $c_k^{(1)} = Lu_k^{(1)}$ in Algorithm 1 are replaced by $u_k^{(1)} = r_{k-1}$ and $c_k^{(1)} = LTu_k^{(1)}$ in Algorithm 2. Note that in Algorithm 1 $c_k \in \mathbb{C}^m$ and $u_k \in \mathbb{C}^n$, whereas in Algorithm 2 $u_k, c_k \in \mathbb{C}^m$. So when $m \ll n$, Algorithm 2 needs much less memory and work than Algorithm 1 to obtain the same iterate x_k.

4. THE CONVERGENCE BEHAVIOUR OF THE PRECONDITIONED KRYLOV METHOD

It is well known that the convergence behaviour of Krylov subspace methods for square linear systems depends on the eigenvalues of the (preconditioned) matrix. In this section we generalize some of these results for square linear matrices to the Krylov method proposed in Section 3. To approximate the eigenvalues of LT we describe the Arnoldi method [1] applied to the square matrix LT, where we assume that the matrix T is the same in every iteration. This method provides so-called Ritz values, which are approximations of the eigenvalues of LT.

The Arnoldi method [1] applied to $LT \in \mathbb{C}^{m\times m}$ can be described as follows.

ALGORITHM 3.

$r_0 = b$, $v_1 = r_0/\|r_0\|_2$;
for $j = 1, ..., k$ do
    $v_{j+1} = LTv_j$;
    for $i = 1, ..., j$ do
        $h_{ij} = v_{j+1}^* v_i$;
        $v_{j+1} = v_{j+1} - h_{ij}v_i$;
    endfor
    $h_{j+1,j} = \|v_{j+1}\|_2$;
    $v_{j+1} = v_{j+1}/h_{j+1,j}$;
endfor.

After the algorithm is completed the upper Hessenberg matrix $H_k \in \mathbb{C}^{k\times k}$ can be formed:

$$H_k = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \dots & \dots & h_{1k} \\ h_{21} & h_{22} & h_{23} & \dots & \dots & h_{2k} \\ 0 & h_{32} & h_{33} & \dots & \dots & h_{3k} \\ 0 & 0 & h_{43} & \ddots & \dots & h_{4k} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & h_{k-1,k} & h_{kk} \end{pmatrix}.$$

For the choice $T = L^*$ the matrix H_k is tridiagonal and the Arnoldi method leads to the same results as the Lanczos method [15]. In general, k is much smaller than m, or n, so H_k is a relatively small matrix. The eigenvalues of H_k can be calculated using MATLAB or LAPACK subroutines. These eigenvalues are called Ritz values and are approximations of the eigenvalues of LT. The Ritz values have the following properties:

— they converge fast to the extreme eigenvalues,

— they only converge to the eigenvalues for which the corresponding eigenvectors have a nonzero component in the right-hand side b.

Below we give some results concerning the rate of convergence of Algorithm 1. Since Algorithm 1 and 2 lead to the same iterates x_k the convergence results are also applicable to Algorithm 2. The convergence behaviour is investigated using the fact that the iterate x_k obtained from Algorithm 1 satisfies the equation:

$$\|b - Lx_k\|_2 = \min_{\xi\in K_k(TL,Tr_0)} \|r_0 - L\xi\|_2. \tag{8}$$

Note that $r_k = b - Lx_k$ can also be written as $r_k = \hat{p}_k(LT)r_0$, where $\hat{p}_k$ is a polynomial of degree k such that $\hat{p}_k(0) = 1$. The class of polynomial of degree at most k and constant term 1 is denoted by Π_k^1. Furthermore, Eq. (8) implies that

$$\|r_k\|_2 \le \|p_k(LT)r_0\|_2,$$

for every polynomial $p_k \in \Pi_k^1$. In the remainder of this paper we assume that the matrix LT is diagonalizable [11, p. 338].

DEFINITION 2. For the diagonalizable matrix LT there is an $S \in \mathbb{C}^{m\times m}$ such that $S^{-1}(LT)S = D$, where

$$D = \begin{pmatrix} \lambda_1 & & \varnothing \\ & \ddots & \\ \varnothing & & \lambda_m \end{pmatrix}$$

and $\lambda_i \in \mathbb{C}$. We have ordered λ_i and s_i, the ith column of S, such that r_0 can be written as $r_0 = \sum_{i=1}^{t} \alpha_i s_i$, where $\alpha_i \neq 0$ for $i = 1, ..., t$.

THEOREM 4.1. *If $\varepsilon^{(k)}$ is defined as*

$$\varepsilon^{(k)} = \min_{p_k\in\Pi_k^1} \max_{1\le i\le t} |p_k(\lambda_i)|,$$

then the residual r_k satisfies the inequality

$$\|r_k\|_2 \le \varepsilon^{(k)}\|S\|_2\|S^{-1}\|_2\|r_0\|_2.$$

For the proof we refer to Theorem 3.3 of [9].

The condition number $\kappa_2(S) = \|S\|_2\|S^{-1}\|_2$ can be large. However, for a square linear system we see that the quantity $\varepsilon^{(k)}$ gives a good indication of the rate of convergence of Algorithm 1 (which is equal to GCR [9] for square systems). For the choice $T = L^*$, the CG method, the condition number $\kappa_2(S)$ is equal to 1.

Many bounds for $\varepsilon^{(k)}$ are given in the literature. We only give two of them:

— if all eigenvalues λ_i are on the real axis and the effective condition number κ_{eff} of LT is defined as $\kappa_{\text{eff}} =$

$\max_{1\le i\le t}|\lambda_i|/\min_{1\le i\le t}|\lambda_i|$ then the following inequality for $\varepsilon^{(k)}$ holds (see Theorem 10.2.5 in [11]):

$$\varepsilon^{(k)} \le 2\left(\frac{\sqrt{\kappa_{\text{eff}}}-1}{\sqrt{\kappa_{\text{eff}}}+1}\right)^k,$$

— if all eigenvalues are enclosed in a circle centered at $C \in \mathbb{R}$ with $C > 0$ and having radius R with $C > R$ then $\varepsilon^{(k)} \le (R/C)^k$ (compare Theorem 5 given in [17]).

Finally, for square linear systems originating from discretized PDE problems one frequently observes superlinear convergence behaviour. This means that the reduction factor $\|r_{k+1}\|_2/\|r_k\|_2$ decreases after some iterations. In [21] an explanation of superlinear convergence is given, based on the convergence of the Ritz values to the extreme eigenvalues. It is straightforward to generalize this explanation to the convergence behaviour of Algorithm 1.

5. EXAMPLE: SEISMIC INVERSE SCATTERING

We now consider a least-squares inverse problem from exploration seismology. The aim of exploration seismology is to determine structures in the earth's subsurface from seismic measurements at the surface. This type of inverse problem is underdetermined and can therefore serve as an example for the preconditioned Krylov subspace method discussed in Section 3. With the aid of asymptotic methods we compute a preconditioner T and subsequently analyse the performance of the resulting method by studying the rate of convergence as compared to other methods and by applying the Arnoldi method discussed in Section 4.

5.1. *Formulation of the Problem*

We consider the scattering of acoustic waves by a bounded two-dimensional object V embedded in a homogeneous, ideal fluid material and differing from its surroundings in its modulus of compression κ, i.e.,

$$\kappa(x, z) = \kappa^{(0)}(1 + \Delta\kappa(x, z)). \tag{9}$$

In Eq. (9), $\kappa^{(0)}$ is the modulus of compression of the surrounding material and $\Delta\kappa$ is a small contrast. We have chosen Cartesian coordinates x and z, denoting horizontal coordinate and depth, respectively. In order to estimate $\Delta\kappa$, a number of experiments have been performed at the surface (Fig. 1). In each experiment, the configuration is probed with the field generated by a monochromatic line-source with angular frequency ω_k, located at $x = x_s$ and $z = 0$. This field is recorded by a receiver at $x = x_s + o$ and $z = 0$. In all experiments, the source-receiver offset o is kept constant. The midpoint between source and receiver is denoted by x_j. This implies that each measurement is

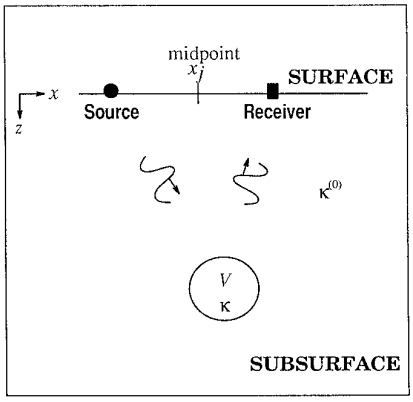

FIG. 1. Configuration of the seismic inverse problem.

labeled by the two indices k (frequency ω_k) and j (midpoint x_j), respectively.

In order to obtain an integral equation relating the (unknown) contrast $\Delta\kappa$ to the measured field, we decompose the total pressure wavefield into the incident field p^{inc} (the field in the absence of contrast $\Delta\kappa$) and the scattered wavefield p^{sc}. We can then derive an integral representation for the scattered field in terms of the contrast $\Delta\kappa$. For small values of the contrast, this representation can be linearized around $\kappa^{(0)}$ and we obtain the linear integral equation of the first kind [7],

$$p^{\text{sc}}(x_j, \omega_k) = \int_V \Delta\kappa(x', z')\mathbf{L}(x', z', x_j, \omega_k)\, dx'\, dz'. \tag{10}$$

In this equation, $x_j = j\,\Delta x$ are the midpoints for subsequent experiments, $\omega_k = k\,\Delta\omega$ are discrete frequencies, whereas $\mathbf{L}(x', z', x_j, \omega_k)$ is given by

$$\mathbf{L}(x', z', x_j, \omega_k) = -\frac{\omega_k^2 s(\omega_k)}{16(c^{(0)})^2} H_0^{(2)}(\omega_k \tau_{up}(x', z', x_j)) \cdot H_0^{(2)}(\omega_k \tau_{\text{down}}(x', z', x_j)). \tag{11}$$

In Eq. (11), $c^{(0)}$ denotes the velocity of the background medium, $s(\omega)$ is the amplitude spectrum of the source, and $H_0^{(2)}$ is the Hankel function of order zero and the second kind. Writing $\mathbf{x}' = (x', z')$, $\mathbf{x}_j = (x_j, 0)$ and $\mathbf{o} = (o, 0)$, we can express the traveltime of waves from the source downwards to any subsurface point $\mathbf{x}'$, τ_{down}, in the form

$$\tau_{\text{down}}(x', z', x_j) = \frac{\|\mathbf{x}' - \mathbf{x}_j + \tfrac{1}{2}\mathbf{o}\|_2}{c^{(0)}}. \tag{12}$$

(In Eq. (12), $\|\cdot\|_2$ denotes physical length of the vector.) In a similar way, we can express τ_{up}, the traveltime of waves from the subsurface point upwards to the receiver as

$$\tau_{\text{up}}(x', z', x_j) = \frac{\|\mathbf{x}' - \mathbf{x}_j - \frac{1}{2}\mathbf{o}\|_2}{c^{(0)}}. \quad (13)$$

Since both $\omega\tau_{\text{down}}$ and $\omega\tau_{\text{up}}$ are large for the cases of interest in exploration seismology, we replace the Hankel functions by their asymptotic approximations, i.e.,

$$H_0^{(2)}(\lambda) \sim \sqrt{2/\pi\lambda}\,\exp\{-i(\lambda - \tfrac{1}{4}\pi)\} \quad (\lambda \to \infty). \quad (14)$$

After inserting the asymptotic approximation (14) into (11), we obtain

$$\mathbf{L}(x', z', x_j, \omega_k) = -\frac{is(\omega_k)\omega_k}{8\pi c^{(0)}} \cdot \frac{\exp\{-i\omega_k\tau(x', z', x_j)\}}{R(x', z', x_j)}, \quad (15)$$

where τ $(=(\tau_{\text{down}} + \tau_{\text{up}}))$ can be interpreted as the total travel time and

$$R = (\|\mathbf{x}' - \mathbf{x}_j - \tfrac{1}{2}\mathbf{o}\|_2\|\mathbf{x}' - \mathbf{x}_j + \tfrac{1}{2}\mathbf{o}\|_2)^{1/2} \quad (16)$$

is related to the geometrical spreading of waves, propagating through the background medium.

Writing (10) in operator notation finally results in

$$L\,\Delta\kappa = p^{\text{sc}}, \quad (17)$$

with

$$L\,\Delta\kappa = \int_V \Delta\kappa(x', z')\mathbf{L}(x', z', x_j, \omega_k)\,dx'\,dz', \quad (18)$$

where the function $\mathbf{L}$ is given by Eq. (15). The quantity $L\,\Delta\kappa$ of Eq. (18) represents a matrix–vector product, since we have to discretize the integral over V. The discretization interval Δ_V has to be chosen small enough in order to sample the oscillatory integrand accurately.

The inverse problem given in Eq. (17) can be formulated as a least-squares problem (Section 2), relating the calculated data p^{sc} for a model x to the measured data d: given the matrix $L \in \mathbb{C}^{m\times n}$ and right-hand side $d \in \mathbb{C}^m$, find a solution vector $x \in \mathbb{R}^n$ such that

$$\|Lx - d\|_2 = \min_{y\in\mathbb{R}^n} \|Ly - d\|_2, \quad (19)$$

with the l^2-norm now given by

$$\|Lx - d\|_2 = \left(\sum_{j,k} |Lx(x_j, \omega_k) - d(x_j, \omega_k)|^2\right)^{1/2}. \quad (20)$$

The elements of the matrix L are given by Eq. (15) and the data d are given by measurements at the surface. The dimension m of the data space is determined by the number of measurements and is equal to the number of (discrete) frequencies ω_k times the number of midpoints x_j. The dimension n of the model space can be chosen and is equal to the number of cells, into which the subsurface is discretized. The dimension n of the model space can be much larger than the dimension m of the data space $(m \ll n)$.

5.2. *Choice of the Preconditioning Operator T*

In order to accelerate the convergence, the preconditioning operator T should be chosen as close as possible to L^{-1}, the generalized inverse of L. In [7], a Born inversion method is discussed, where this inverse is determined approximately (with the aid of high-frequency asymptotics) and where $\Delta\kappa$ is expressed in terms of the data d by the relation

$$\Delta\kappa(x', z') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A(x', z', x, \omega)d(x, \omega)\,dx\,d\omega. \quad (21)$$

The weight function $A(x', z', x, \omega)$ is derived in the Appendix. Equation (21) requires the knowledge of d for $-\infty < x < \infty$ and $-\infty < \omega < \infty$. If we take the preconditioning operator T equal to T^B, given by

$$T^B(x', z', x_j, \omega_k) = A(x', z', x_j, \omega_k)\,\Delta x\,\Delta\omega, \quad (22)$$

we observe that this choice of T approximates the generalized inverse of L if the sampling is "good enough," in the sense that $\{x_j : j = 1, ..., J\}$ and $\{\omega_k : k = 1, ..., K\}$ represent adequate discretizations of the infinite intervals occurring in Eq. (21). For sparsely sampled data, this is not the case, but T^B can still be a fair approximation of L^{-1} and can therefore be used as preconditioner. The use of high frequency asymptotic methods for computing preconditioning operators for square systems has been suggested by [14].

5.3. *Comparison of the Various Methods*

We consider the estimation of two different subsurface models that only differ in the size of V (dimension n), referred to as the small and the large models. For both problems we use data gathered for nine midpoints (nine source-receiver pairs), spaced 45 m apart, with an offset of 150 m for all midpoints. The discretization-interval Δ_V is chosen equal to 15 m. This is small enough for these problems, since we need about five samples per dominant wavelength (equal to 80 m here) for a reasonable sampling. The data d have been calculated from Eq. (17) for a model consisting of two pointscatterers (V consists of two one-cell scatterers).

First, we have considered the estimation of the small subsurface model, discretized into 2500 cells. For this problem $m = 738$ and $n = 2500$. The pointscatterers were situated at a depth of 450 and 900 m. Second, we have also considered the estimation of the larger subsurface model, discretized into 10,000 cells. Here $m = 738$ again, but now $n = 10{,}000$. The pointscatterers were situated at a depth of 600 and 1350 m.

At the end of Subsection 3.1 we note that comparisons given in the literature suggest that CG is more efficient than ART and SIRT for underdetermined systems. This motivates us to compare the preconditioned Krylov subspace methods only with the CG method.

In Fig. 2, $\log_{10}(\|r_k\|_2/\|r_0\|_2)$ of the small model problem is displayed as a function of the iteration index for four different schemes:

G. The gradient scheme.

CG. The conjugate gradient scheme (Algorithm 2 with $T = L^*$).

PKNO. The nonorthogonalized preconditioned Krylov subspace scheme (Algorithm 2 with $T = T^B$ of Eq. (22) and no orthogonalisation). In [18] this method has been referred to as PSOR (preconditioned successive overrelaxation scheme).

PK. The preconditioned Krylov subspace scheme (Algorithm 2 with choice $T = T^B$ of Eq. (22)).

The G and PKNO methods are included, because in our application a low accuracy is required. In such a case PKNO can be faster than PK, because the number of iterations is comparable, but the work per iteration is less for PKNO (see Table IV and V).

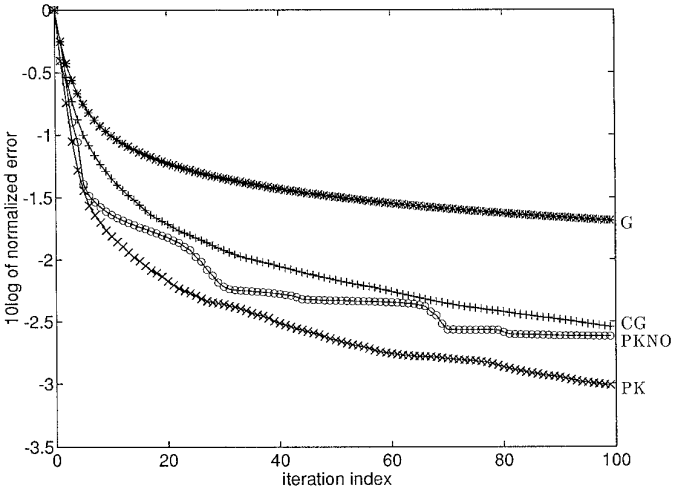

FIG. 2. Rate of convergence for the small problem (2500 unknowns) for the following iterative schemes: gradient scheme (G), conjugate gradient scheme (CG), preconditioned Krylov scheme without orthogonalization (PKNO) and preconditioned Krylov scheme (PK). The error norm has been normalized with respect to the error of the initial model u_0 (=0).

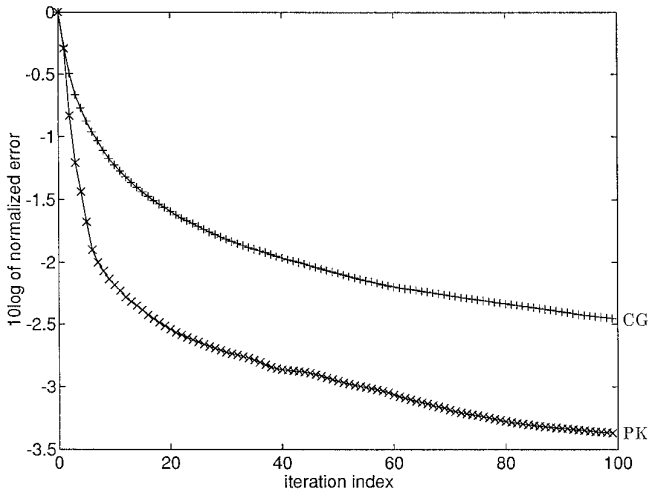

FIG. 3. Rate of convergence for the large problem (10,000 unknowns) for the following iterative schemes: conjugate gradient scheme (CG) and preconditioned Krylov scheme (PK). The error norm has been normalized with respect to the error of the initial model u_0 (=0).

The starting model x_0 is taken equal to zero for all subsurface points. The scaling is carried out with respect to the l^2-norm of the data (which is equal to the residual of the start iteration). Apparently, the choice T^B given by Eq. (22) is closer to the generalized inverse L^{-1} than the choice L^*, used in the gradient (G) and conjugate gradient (CG) methods, and gives rise to a much faster convergence, especially in the first iterations. The amount of work per iteration is more or less the same for all methods. Only for the PK method the amount of work per iteration increases noticeably for large numbers of iterations. Note that the rate of convergence deteriorates for all methods. Furthermore, PK uses much fewer iterations than CG and G to obtain the same accuracy. Initially the convergence of PK and PKNO is comparable; however, after 10 iterations the convergence of PKNO stagnates at certain intervals.

In Fig. 3, $\log_{10}(\|r_k\|_2/\|r_0\|_2)$ of the large model-problem is displayed as a function of the iteration index for two different schemes: the CG and the PK schemes. The scaling is again with respect to the l^2-norm of the data. From a comparison of Figs. 2 and 3 we conclude that the preconditioning operator T^B seems to be even more effective in accelerating the rate of convergence for large-scale inverse problems containing 10,000 unknowns.

Having compared the rate of convergence of the different schemes for different sizes of n, we now want to consider more quantitatively the two different choices of T, i.e., L^* and T^B. We do this by considering the small problem ($n = 2500$) and calculating the Ritz-values for the different T, using the scheme given in Algorithm 3. In

TABLE I

The Ritzvalues for the Small Problem (CG)

k	θ_1	θ_2	θ_3	θ_{k-3}	θ_{k-2}	θ_{k-1}	θ_k
5	0.00217	0.00717	0.01378	0.00717	0.01378	0.02255	0.03010
10	0.00052	0.00230	0.00484	0.01996	0.02390	0.02622	0.03014
15	0.00023	0.00105	0.00242	0.02413	0.02576	0.02691	0.03014
20	0.00012	0.00056	0.00130	0.02445	0.02631	0.02706	0.03014
25	0.00007	0.00035	0.00087	0.02555	0.02633	0.02706	0.03014
30	0.00005	0.00026	0.00060	0.02568	0.02633	0.02706	0.03014

Table I some Ritz-values are shown for the CG method for different numbers of iterations k. In Table II the same is shown for the preconditioned Krylov subspace method (PK). For both methods the largest Ritz values converge fast to the corresponding eigenvalues, whereas the smallest Ritz value becomes closer and closer to zero. This observation is comparable to observations made in [19]. The largest Ritz values of LT^B are better clustered around 1 than those of LL^* which illustrates (also quantitatively) that T^B is a better approximation of the generalized inverse of L.

We compare the theoretical convergence results with the experimental results. The effective condition number is approximated by θ_k/θ_1 using Tables I and II. The observed rate of convergence $\|r_{i+1}\|_2/\|r_i\|_2$ and the (approximated) theoretical rate of convergence $(\sqrt{\theta_k/\theta_1} - 1)/(\sqrt{\theta_k/\theta_1} + 1)$ are presented in Fig. 4 for CG and in Fig. 5 for PK. Note that there is a qualitatively close correspondence between theory and experiments (generalizing the analysis of CG given in [20] to PK may even lead to better theoretical results). Note that instead of a faster rate of convergence observed in PDE problems the rate of convergence of CG and PK deteriorates when the number of iterations increase. To understand this behaviour we note that the initial residual has large components in eigenvectors corresponding to the large eigenvalues. After a number of iterations these components decrease considerably and become comparable with the components corresponding to small eigenvalues. So the effective condition number increases which explains the deterioration of the rate of convergence.

In Table III we summarize the theoretical rates of convergence of CG and PK. From this table we see that the theory predicts (correctly) that PK converges faster than CG. This implies that the Ritz values may be used to measure the quality of a given preconditioner.

In practice we only require low accuracy, since seismic data contain a relatively large noise component. Therefore we also present in Tables IV and V the CPU time and number of iterations that are required for reducing the residual to 5% of its initial value (i.e., by choosing eps = $0.05\ \|r_0\|_2$ in Algorithm 2). The CPU time is measured on an HP 720 workstation. We see that for this accuracy the methods PK and PKNO are comparable. Note that for the large model-problem PK and PKNO are approximately three times faster than CG. Note that there are only 10 extra memory vectors needed for the PK method. This amount of memory is negligible to the amount of memory to store the full matrix L.

6. CONCLUSIONS

Two related preconditioned Krylov subspace methods have been presented to solve underdetermined least squares problems. We show that the first variant has no breakdown and that the second variant is more efficient than the first one. To analyse the convergence behaviour a relation between the residual and the eigenvalues of the precon-

TABLE II

The Ritzvalues for the Small Problem (PK)

k	θ_1	θ_2	θ_3	θ_{k-3}	θ_{k-2}	θ_{k-1}	θ_k
5	0.42149	1.03663	1.97072	1.03663	1.97072	2.95177	3.63582
10	0.08392	0.36642	0.72695	2.64811	3.15636	3.42586	3.65609
15	0.02992	0.15581	0.34739	3.16691	3.33419	3.48742	3.65584
20	0.02007	0.08263	0.20526	3.23081	3.41193	3.49093	3.65589
25	0.01332	0.03896	0.11507	3.38580	3.43911	3.49392	3.65586
30	0.00698	0.02819	0.07936	3.41293	3.44489	3.49280	3.65586

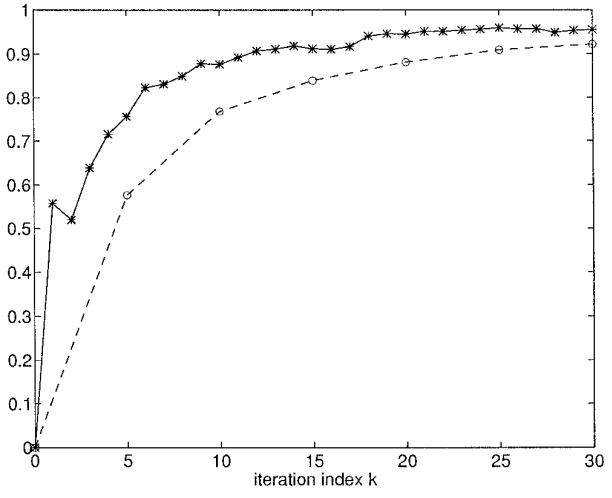

FIG. 4. Comparison between the observed rate of convergence $\|r_{k+1}\|_2/\|r_k\|_2$ (solid line), and the one estimated from Table I (dashed line) for the CG scheme (small problem).

ditioned matrix has been given. Furthermore, the Arnoldi method is described to approximate the eigenvalues.

The CG method and the preconditioned Krylov method have been applied to a seismic inverse scattering problem. For this problem a suitable preconditioner is obtained by using asymptotic theory. From the experiments it follows that the preconditioned Krylov methods are three times as fast as the CG method for large problems. Finally, we note a good qualitative correspondence between the theoretical and experimental rate of convergence.

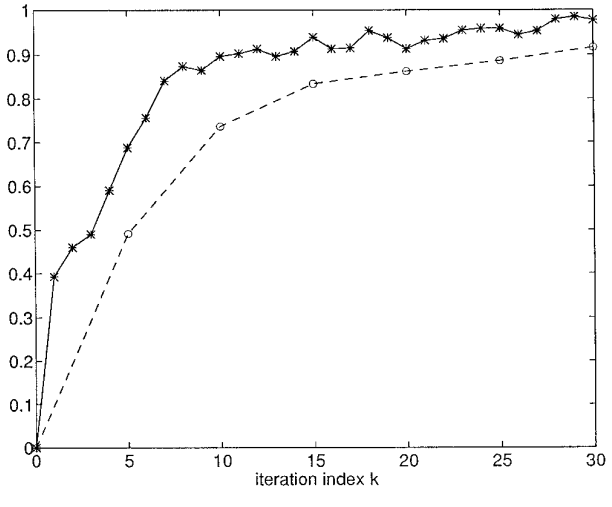

FIG. 5. Comparison between the observed rate of convergence $\|r_{k+1}\|_2/\|r_k\|_2$ (solid line), and the one estimated from Table II (dashed line) for the PK scheme (small problem).

TABLE III

The Theoretical Rate of Convergence of CG and PK (Small Problem)

k	5	10	15	20	25	30
CG	0.576	0.767	0.839	0.881	0.908	0.921
PK	0.492	0.736	0.834	0.862	0.886	0.916

APPENDIX A: COMPUTATION OF THE PRECONDITIONING OPERATOR *T*

In order to derive a relation between the contrast $\Delta\kappa$ and the wavefield d of the form (21), we first insert L, given by Eq. (15), into Eq. (10). We then obtain

$$p^{sc}(x,\omega) = -\int_V \Delta\kappa(\mathbf{x}')\,\frac{\exp\{-i\omega\tau(\mathbf{x}',x)\}}{R(\mathbf{x}',x)}\,W(\omega)\,dx'\,dz', \tag{A1}$$

with $\mathbf{x}' = (x', z')$ and where we have introduced the variable $W(\omega) = i\omega s(\omega)/8\pi c^{(0)}$.

In our derivation, we assume that $s(\omega)$ is not equal to zero and the data d are equal to p^{sc} given by Eq. (A1).

We now write Eq. (21) in the form

$$\Delta\kappa(\mathbf{x}') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\{i\omega\tau(\mathbf{x}',x)\}Q(\mathbf{x}',x,\omega)p^{sc}(x,\omega)\,dx\,d\omega, \tag{A2}$$

where Q is a weight function to be determined and the integral is taken on the midpoints between sources and receivers (x) and frequencies ω. We now want to determine Q such, that Eq. (A2) gives the correct estimate of $\Delta\kappa(\mathbf{x}')$ if it consists of a point scatterer at (arbitrary) location $\mathbf{x}_0'$. This implies that the contrast function can be written as

$$\Delta\kappa(\mathbf{x}') = \Delta\kappa(\mathbf{x}_0')\delta(\mathbf{x}' - \mathbf{x}_0'), \tag{A3}$$

where δ denotes the two-dimensional Dirac delta function. Substituting this relation into Eq. (A1) and the resulting

TABLE IV

Number of Iterations and CPU Time (in Seconds) for the Small Problem

50×50	CG	PK	G	PKNO
k	10	5	26	5
CPU	68.45	40.5	177.37	40.4

TABLE V

Number of Iterations and CPU Time (in Seconds) for the Large Problem

100×100	CG	PK	G	PKNO
k	12	4	42	4
CPU	331	132	1146	127

expression for p^{sc} into Eq. (A2), we obtain

$$\delta(\mathbf{x}' - \mathbf{x}_0') = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\exp[i\omega\{\tau(\mathbf{x}', x) - \tau(\mathbf{x}_0', x)\}]}{R(\mathbf{x}_0', x)} Q(\mathbf{x}', x, \omega) W(\omega)\, dx\, d\omega. \tag{A4}$$

Since the left-hand side of Eq. (A4) involves a Dirac delta function acting at $\mathbf{x}' = \mathbf{x}_0'$, it seems intuitively reasonable to approximate the traveltime $\tau(\mathbf{x}', x)$ by the first two terms of its Taylor expansion around $\mathbf{x}' = \mathbf{x}_0'$

$$\tau(\mathbf{x}', x) \approx \tau(\mathbf{x}_0', x) + \nabla_{\mathbf{x}'}\tau(\mathbf{x}', x)|_{\mathbf{x}_0'} \cdot (\mathbf{x}' - \mathbf{x}_0'), \tag{A5}$$

to use the approximation

$$Q(\mathbf{x}', x, \omega) \approx Q(\mathbf{x}_0', x, \omega) \tag{A6}$$

and neglect higher order terms [7]. Then, Eq. (A4) can be rewritten as

$$\delta(\mathbf{x}' - \mathbf{x}_0') = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\exp[i\omega\, \nabla_{\mathbf{x}'}\tau(\mathbf{x}', x)|_{\mathbf{x}_0'} \cdot (\mathbf{x}' - \mathbf{x}_0')]}{R(\mathbf{x}_0', x)} Q(\mathbf{x}_0', x, \omega) W(\omega)\, dx\, d\omega. \tag{A7}$$

Following [7], we make a transformation of integration variables,

$$(\mathrm{x}, \omega) \mapsto (k_1, k_2) = \mathbf{k}, \tag{A8}$$

where

$$\mathbf{k}(x, \omega) = -\omega \nabla_{\mathbf{x}'}\tau(\mathbf{x}', x)|_{\mathbf{x}_0'}. \tag{A9}$$

The Jacobian of this transformation is given by

$$J = \frac{\partial(x, \omega)}{\partial(k_1, k_2)} = \left[\frac{\partial(k_1, k_2)}{\partial(x, \omega)}\right]^{-1} = -\frac{1}{\omega} \cdot \begin{vmatrix} \tau_{x'x} & \tau_{x'} \\ \tau_{z'x} & \tau_{z'} \end{vmatrix}^{-1}, \tag{A10}$$

where $\tau_{x'}$ denotes differentiation of τ with respect to x' (and similar notations for $\tau_{x'x}$, $\tau_{z'}$, and $\tau_{z'x}$).

If we now choose $Q(\mathbf{x}_0', x, \omega)$ to be given by

$$Q(\mathbf{x}_0', x, \omega) = -\frac{R(\mathbf{x}_0', x)}{(2\pi)^2 W(\omega)} \cdot |J|^{-1} = -\frac{|\omega| R(\mathbf{x}_0', x)}{(2\pi)^2 W(\omega)} \cdot |\tau_{z'x}\tau_{x'} - \tau_{x'x}\tau_{z'}|, \tag{A11}$$

Equation (A7) can be written

$$\delta(\mathbf{x}' - \mathbf{x}_0') = \frac{1}{(2\pi)^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\{-i\mathbf{k} \cdot (\mathbf{x}' - \mathbf{x}_0')\}\, dk_1\, dk_2, \tag{A12}$$

which is an identity.

From Eqs. (21) and (A2), we now conclude that the weight function A is given by

$$A(\mathbf{x}', x, \omega) = \exp\{i\omega\tau(\mathbf{x}', x)\} Q(\mathbf{x}', x, \omega), \tag{A13}$$

with Q given by Eq. (A11). This concludes the determination of A. For inhomogeneous background media, with material properties that are sufficiently smooth functions of the spatial coordinates, the functions τ and Q can be determined using ray-tracing methods [4]. For the case of a homogeneous background medium, τ and its spatial derivatives can be computed explicitly from Eqs. (12)–(13) using the relation $\tau = \tau_{\mathrm{down}} + \tau_{\mathrm{up}}$ after which Q follows from Eq. (A11). (The determination of R for a homogeneous background is discussed below Eq. (15).

REFERENCES

1. W. E. Arnoldi, *Quart. Appl. Math.* **9,** 17 (1951).
2. R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der horst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (SIAM, Philadelphia, 1993).
3. Å. Björck and T. Elving, *BIT* **19,** 145 (1979).
4. N. Bleistein, *Mathematical Methods for Wave Phenomena* (Academic Press, New York, 1984).
5. R. E. Cline and R. J. Plemmons, *SIAM Rev.* **18,** 92 (1976).
6. E. J. Craig, *J. Math. Phys.* **34,** 64 (1955).
7. J. K. Cohen, F. G. Hagin, and N. Bleistein, *Geophysics* **51,** 1552 (1986).
8. M. C. A. van Dijke, thesis, State University Utrecht, Utrecht, 1992.
9. S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *SIAM J. Numer. Anal.* **20,** 345 (1983).
10. P. Gilbert, *J. Theor. Biol.* **36,** 105 (1972).
11. G. H. Golub and C. F. Van Loan, *Matrix Computations,* 2nd ed. (Johns Hopkins Univ. Press, Baltimore, 1989).
12. R. Gordon, R. Bender, and G. T. Herman, *J. Theor. Biol.* **29,** 471 (1970).
13. M. R. Hestenes and E. Stiefel, *J. Res. Nat. Bur. Stand.* **49,** 409 (1952).

14. R. E. Kleinman and P. M. van den Berg, *Radio Sci.* **26,** 175 (1991).
15. C. Lanczos, *J. Res. Nat. Bur. Stand.* **45,** 255 (1950).
16. C. C. Paige and M. A. Saunders, *ACM Trans. Math. Software* **8,** 43 (1982).
17. Y. Saad and M. H. Schultz, *SIAM J. Sci. Statist. Comput.* **7,** 856 (1986).
18. A. G. J. Sevink and G. C. Herman, ''Fast Iterative Solution of Sparsely Sampled Seismic Inverse Problems,'' in *Proceedings, Second International Conference on Mathematical and Numerical Aspects of Wave Propagation,* (SIAM, Philadelphia, 1993).
19. A. van der Sluis and H. A. van der Vorst, ''Numerical Solution of Large, Sparse Linear Algebraic Systems Arising from Tomographic Problems, in *Seismic Tomography, with Applications in Global Seismology and Exploration Geophysics,* edited by G. Nolet (Reidel, Dordrecht, 1987).
20. A. van der Sluis and H. A. van der Vorst, *Linear Alg. Appl.* **130,** 257 (1990).
21. H. A. van der Vorst and C. Vuik, *J. Comput. Appl. Math.* **48,** 327 (1993).