Cornell Theory Center / ACRI Technical Report CTC97TR268, 2/97

Accurate Solution of Weighted Least Squares by Iterative Methods*

Elena Y. Bobrovnikova† Stephen A. Vavasis‡

February 6, 1997

Abstract. We consider the weighted least-squares (WLS) problem with a very ill-conditioned weight matrix. Weighted least-squares problems arise in many applications including linear programming, electrical networks, boundary value problems, and structures. Because of roundoff errors, standard iterative methods for solving a WLS problem with ill-conditioned weights may not give the correct answer. Indeed, the difference between the true and computed solution (forward error) may be large. We propose an iterative algorithm, called MINRES-L, for solving WLS problems. The MINRES-L method is the application of MINRES, a Krylov-space method due to Paige and Saunders, to a certain layered linear system. Using a simplified model of the effects of roundoff error, we prove that MINRES-L gives answers with small forward error. We present computational experiments for some applications.

*This work has been supported in part by an NSF Presidential Young Investigator grant, with matching funds received from AT&T and Xerox Corp. Research supported in part by NSF through grant DMS-9505155 and ONR through grant N00014-96-1-0050. Support was also received from the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Dept. of Energy, under Contract W-31-109-Eng-38 through Argonne National Laboratory. Support was also received from the J. S. Guggenheim Foundation.

†Formerly of the Center for Applied Mathematics, Cornell University, Ithaca, New York 14853. Part of this work was done while this author was visiting Lucent Bell Laboratories.

‡Department of Computer Science, Cornell University, Ithaca, New York 14853, vavasis@cs.cornell.edu. Part of this work was done while this author was visiting Argonne National Laboratory.

1 Introduction

Consider the *weighted least-squares* (WLS) problem

$$\min \|D^{1/2}(\mathbf{b} - A\mathbf{x})\|^2, \tag{1}$$

where $D \in \mathbf{R}^{m\times m}$, $A \in \mathbf{R}^{m\times n}$, $\mathbf{b} \in \mathbf{R}^m$, and $m \geq n$. In this formula and for the remainder of this article, $\|\cdot\|$ indicates the 2-norm. We make the following assumptions: D is a diagonal positive definite matrix and rank $A = n$. These assumptions imply that (1) is a nonsingular linear system with a unique solution. The normal equations for (1) have the form

$$A^T DA\mathbf{x} = A^T D\mathbf{b}. \tag{2}$$

Weighted least-squares problems arise in several application domains including linear programming, electrical power networks, elliptic boundary value problems and structural analysis, as observed by Strang [21]. This article focuses on the case when matrix D is severely ill-conditioned. This happens in certain classes of electrical power networks. In this case, A is a node-arc adjacency matrix, D is matrix of load conductivities, $\mathbf{b}$ is the vector of voltage sources, and $\mathbf{x}$ is the vector of voltages of the nodes. Ill-conditioning occurs when resistors are out of scale, for instance, when modeling leakage of current through insulators.

Ill-conditioning also occurs in linear programming when an interior-point method is used. To compute the Newton step for an interior-point method, we need to solve a weighted least-squares equation of the form (2). Since some of the slack variables become zero at the solution, matrix D always becomes ill-conditioned as the iterations approach the boundary of the feasible region. In Section 9, we cover this application in more detail. Ill-conditioning also occurs in finite element methods for certain classes of boundary value problems, for example, in the heat equilibrium equation $\nabla \cdot (c\nabla u) = 0$ when thermal conductivity field c varies widely in scale.

An important property of problem (1) or (2) is the norm bound on the solution, which was obtained independently by Stewart [20], Todd [22] and several other authors. See [6] for a more complete bibliography. Here we state this result as in the paper by Stewart.

Theorem 1 *Let $\mathcal{D}$ denote the set of all positive definite $m \times m$ real diagonal matrices. Let A be an $m \times n$ real matrix of rank n. Then there exist constants*

χ_A *and* $\bar{\chi}_A$ *such that for any* $D \in \mathcal{D}$

$$\begin{aligned} \|(A^T D A)^{-1} A^T D\| &\leq \chi_A, \qquad \textit{and} & (3) \\ \|A(A^T D A)^{-1} A^T D\| &\leq \bar{\chi}_A. & (4) \end{aligned}$$

Note that the matrix appearing in (3) is the solution operator for the normal equations (2), in other words, (2) can be rewritten as $\mathbf{x} = (A^T D A)^{-1} A^T D \mathbf{b}$.

Since the bounds (3), (4) exist, we can hope that there exist algorithms for (2) that possess the same property, namely, the forward error bound does not depend on D. We will call these algorithms stable, where *stability*, as defined by Vavasis [23], means that forward error in the computed solution $\hat{\mathbf{x}}$ satisfies

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \epsilon \cdot f(A) \cdot \|\mathbf{b}\|, \tag{5}$$

where ϵ is machine precision and $f(A)$ is some function of A not depending on D. Note that the underlying rationale for this kind of bound is that the conditioning problems in (1) stem from an ill-conditioned D rather than an ill-conditioned A.

This stability property is not possessed by standard direct methods such as QR factorization, Cholesky factorization, symmetric indefinite factorization, range-space and null-space methods, nor by standard iterative methods such as conjugate gradient applied to (2). The only two algorithms in literature that are proved to have this property are the NSH algorithm by Vavasis [23] and the complete orthogonal decomposition (COD) algorithm by Hough and Vavasis [12], both of them direct. See Björck [1] for more information about algorithms for least-squares problems.

We would like to have stable iterative methods for this problem because iterative methods can be much more efficient than direct methods for large sparse problems, which is the common setting in applications.

This article presents an iterative algorithm for WLS problems called MINRES-L. MINRES-L consists of applying the MINRES algorithm of Paige and Saunders [14] to a certain layered linear system. We prove that MINRES-L satisfies (5). This proof of the forward error bound for MINRES-L is based on a simplified model of how roundoff error affects Krylov space methods. This analysis is then confirmed with computational experiments in Section 8.

(The simplified model itself is described in Section 5.) An analysis of roundoff in MINRES-L starting from first principles is not presented here because the effect of roundoff on the MINRES iteration is still not fully understood.

MINRES-L imposes the additional assumption on the WLS problem instance that D is "layered." This assumption is made without loss of generality (i.e., every weighted least-squares problem can be rewritten in layered form), but the MINRES-L algorithm is inefficient for problems with many layers.

This article is organized as follows. In Section 2 we state the layering assumption, and also the layered least-squares (LLS) problem. In Section 3 we consider previous work. In Section 4 we describe the MINRES-L method for two-layered WLS problems. In Section 5 we analyze the convergence in the two-layered case using the simplifying assumptions about roundoff error. In Section 6 and Section 7 we extend the algorithm and analysis to the case of p layers. In Section 8 we present some computational experiments in support of our claims. In Section 9 we consider application of MINRES-L to interior-point methods for linear programming.

2 The Layering Assumption

Recall that we have already assumed that the weight matrix D appearing in (1) is diagonal, positive definite and ill-conditioned. For the rest of this article we impose an additional "layering" assumption: we assume, after a suitable permutation of the rows of $(A, \mathbf{b})$ and corresponding symmetric permutation of D, that D has the structure

$$D = \begin{pmatrix} \delta_1 D_1 & & \\ & \ddots & \\ & & \delta_p D_p \end{pmatrix}, \tag{6}$$

where each D_k is well-conditioned and scaled so that its smallest diagonal entry is 1, and where $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_p > 0$. Let κ denote the maximum diagonal entry among $D_1, \ldots, D_p$. The layering assumption is that κ is not much larger than 1.

Note that this assumption is made without any loss of generality (and we could assume $\kappa = 1$), since we could place each diagonal entry of D in its own layer. Unfortunately, the complexity of our algorithm grows quadratically with p. Furthermore, our upper bound on the forward error degrades as p

increases (see (39) below). Thus, a tacit assumption is that the number of layers p is not too large.

From now on, we write A in partitioned form as

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_p \end{pmatrix}$$

to correspond with the partitioning of D. We partition $\mathbf{b} = [\mathbf{b}_1; \cdots ; \mathbf{b}_p]$ similarly.

Under this assumption, we say that (1) is a "layered WLS" problem. In the context of electrical networks, this assumption means that there are several distinct classes of wires in the circuit, where the resistance of wires in class l is of order $1/\delta_l$. For instance, one class of wires might be transmission lines, whereas the other class might consist of broken wires (open lines) where the resistance is much higher. In the context of the heat equilibrium equation, the layering assumption means that the object under consideration is composed of a small number of different materials. Within each material the conductivity δ_l is constant, but the different materials have very different conductivities. In linear programming, taking $p = 2$ means that the some of the slack variables at the current interior-point iterate are "small" while others are "large."

A limiting case of layered WLS occurs when the gaps between the δ_l's tend to infinity, that is, δ_1 is infinitely larger than δ_2 and so on. As the weight gaps tend to infinity, the solution to (1) tends to the solution of the following problem, which we refer to as *layered least squares* (LLS). Construct a sequence of nested affine subspaces $L_0 \supset L_1 \supset \cdots \supset L_p$ of $\mathbf{R}^n$. These spaces are defined recursively: $L_0 = \mathbf{R}^n$, and

$$L_l = \{\text{minimizers of } \|D_l^{1/2}(A_l\mathbf{x} - \mathbf{b}_l)\| \text{ s.t. } \mathbf{x} \in L_{l-1}\}.$$

Finally, $\mathbf{x}$, the solution to the LLS problem, is the unique element in L_p. The layered least-squares problem was first introduced by Vavasis and Ye [25] as a technique for accelerating the convergence of interior-point methods. They also established the result mentioned above in this paragraph: the solution to the WLS problem in the limit as $\delta_{l+1}/\delta_l \to 0$ for all l converges to the solution of the LLS problem.

Combining this result with Theorem 1 yields the following corollary, also proved by Vavasis and Ye.

Corollary 1 *Let* $\mathbf{x}$ *be the solution to the LLS problem posed with matrix* A *and right-hand side vector* $\mathbf{b}$*. Then* $\|\mathbf{x}\| \leq \chi_A \|\mathbf{b}\|$ *and* $\|A\mathbf{x}\| \leq \bar{\chi}_A \|\mathbf{b}\|$ *for any choice of diagonal positive definite weight matrices* $D_1, \ldots, D_p$.

3 Previous Work

The standard iterative method for least-squares problems, including WLS problems, is conjugate gradient (see Golub and Van Loan [7] or Saad [18]) applied to the normal equations (2). This algorithm is commonly referred to as CGNR, which is how we will denote it here. There are several variants of CGNR in the literature; see, e.g., Björck, Elfving, and Strakoš [2]. Note that in most variants one does not form the triple product $A^T DA$ when applying CG to (2); instead, one forms matrix-vector products involving matrices A^T, D and A. This trick can result in a substantial savings in the running time since $A^T DA$ could be much denser than A alone. The same trick is applicable to our MINRES-L method and was used in our computational experiments.

The difficulty with CGNR is that an inaccurate solution can be returned because $A^T DA$ can be ill-conditioned when D is ill-conditioned. To understand the difficulty, consider the two-layered WLS problem, which is obtained by subtituting (6) in the case $p = 2$ into (2):

$$\delta_1 A_1^T D_1 A_1 \mathbf{x} + \delta_2 A_2^T D_2 A_2 \mathbf{x} = \delta_1 A_1^T D_1 \mathbf{b}_1 + \delta_2 A_2^T D_2 \mathbf{b}_2. \qquad (7)$$

Observe that if $\delta_1 \gg \delta_2$ then Krylov sequence

$$A^T D\mathbf{b}, (A^T DA)A^T D\mathbf{b}, (A^T DA)^2 A^T D\mathbf{b}, \ldots$$

constructed by CGNR is very close to

$$\delta_1 A_1^T D_1 \mathbf{b}_1, \delta_1^2 (A_1^T D_1 A_1) A_1^T D_1 \mathbf{b}_1, \delta_1^3 (A_1^T D_1 A_1)^2 A_1^T D_1 \mathbf{b}_1, \ldots$$

In other words, information about A_2, D_2 and $\mathbf{b}_2$ is lost when forming the Krylov sequence. A different framework for interpreting this difficulty is described in Section 5.

Another iterative method for least-squares problems is LSQR due to Paige and Saunders [15]. This method shares the same difficulty with CGNR because it works in the same Krylov space.

A standard technique for handling ill-conditioning in conjugate gradient is reorthogonalization; see, for example, Paige [16] and Parlett and

Scott [17]. Reorthogonalization, however, cannot solve the difficulty with ill-conditioning in (2) because even the act of forming the first Krylov vector $A^T D\mathbf{b}$ causes a loss of information.

Another technique for addressing ill-conditioned linear systems with iterative methods is called "regularization"; a typical regularization technique modifies the ill-conditioned system with additional terms. See Hanke [10]. Regularization does not appear to be a good approach for solving (1) because (1) already has a well-defined solution (in particular, Theorem 1 implies that solutions are not highly sensitive to perturbation of the data vector $\mathbf{b}$). A regularization technique would compute a completely different solution.

In our own previous work [3], we proposed an iterative method for (2) based on "correcting" the standard CGNR search directions. We have since dropped that approach because we found a case that seemingly could not be handled or detected by that algorithm.

4 MINRES-L for Two Layers

In this section and the next we consider the two-layered case, that is, $p = 2$ in (6). We consider the two-layered case separately from the p-layered case because the two-layered case contains all the main ideas of the general case but is easier to write down and analyze. (In the $p = 1$ case, our algorithm reduces to MINRES applied to (2) and hence is not novel.) Furthermore, the $p = 2$ case is expected to occur commonly in practice. We mention also that the two-layered WLS and LLS problems were considered in §22 of Lawson and Hanson [13].

As noted in the preceding section, the two-layered WLS problem is written in the form (7), in which the diagonal entries of D_1, D_2 on the order of 1 and $\delta_1 \geq \delta_2$. Let us introduce a new variable $\mathbf{v}$ such that

$$A_1^T D_1 A_1 \mathbf{v} = (\delta_1/\delta_2)(A_1^T D_1 A_1 \mathbf{x} - A_1^T D_1 \mathbf{b}_1). \tag{8}$$

Note that this equation always has a solution $\mathbf{v}$ because the right-hand side is in the range of A_1^T. Multiplying (8) by δ_2 and adding to (7) yields

$$A_1^T D_1 A_1 \mathbf{v} = A_2^T D_2 \mathbf{b}_2 - A_2^T D_2 A_2 \mathbf{x}. \tag{9}$$

Putting (8) and (9) together, we get

$$\begin{pmatrix} A_2^T D_2 A_2 & A_1^T D_1 A_1 \\ A_1^T D_1 A_1 & (-\delta_2/\delta_1) A_1^T D_1 A_1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} A_2^T D_2 \mathbf{b}_2 \\ A_1^T D_1 \mathbf{b}_1 \end{pmatrix}. \tag{10}$$

Our algorithm, which we call MINRES-L (for MINRES "layered"), is the application of the MINRES iteration due to Paige and Saunders [14] to (10). Note that (10) is a symmetric linear system.

In general, this linear system is rank deficient because if $(\mathbf{x}; \mathbf{v})$ is a solution and $\mathbf{v}'$ satisfies $A_1\mathbf{v}' = A_1\mathbf{v}$, then $(\mathbf{x}; \mathbf{v}')$ is also a solution. Thus, (10) is rank deficient whenever the rank of A_1 is less than n. This means we must address existence and uniqueness of a solution. Existence follows because the original WLS problem (7) is guaranteed to have a solution. Uniqueness of $\mathbf{x}$ is established as follows: if we add δ_2 times the first row of (10) to δ_1 times the second row, we recover the original WLS problem (7). Since (7) has a unique solution, (10) must uniquely determine $\mathbf{x}$. Since $\mathbf{x}$ is uniquely determined, so is $A_1\mathbf{v}$.

The question arises whether MINRES (in exact arithmetic) will find a solution of (10). MINRES can find a solution only if it lies in the Krylov space, which (because of rank deficiency) is not necessarily full dimensional. This question was answered affirmatively by Theorem 2.4 of Brown and Walker [4]. (Their analysis concerns GMRES, but the same result applies to MINRES in exact arithmetic.) Furthermore, their result states that, assuming the initial guess is $\mathbf{0}$, the computed solution $(\mathbf{x}; \mathbf{v})$ will have minimum norm over all possible solutions. Since $\mathbf{x}$ is uniquely determined, their result implies that $\mathbf{v}$ will have minimum norm.

Recall from Section 3 that the problem with applying conjugate gradient directly to (7) is that the linear system may be ill-conditioned when $\delta_1 \gg \delta_2$, and hence conjugate gradient may return an inaccurate answer. Thus, it may seem paradoxical that we remedy a problem caused by ill-conditioning with an iterative method based on a truly rank-deficient system. One explanation of this paradox concerns the limiting behavior as $\delta_1/\delta_2 \to \infty$. In this case, (7) tends to the linear system $A_1^T D_1 A_1 \mathbf{x} = A_1^T D_1 \mathbf{b}_1$. This system will, in general, not have a unique solution (because A_1 is not assumed to have rank n), so CGNR will compute some solution that may have nothing to do with A_2, D_2, or $\mathbf{b}_2$. Thus, the CGNR solution is not expected to have the forward accuracy that we demand.

On the other hand, as $\delta_1/\delta_2 \to \infty$, we see that (10) tends to

$$\begin{pmatrix} A_2^T D_2 A_2 & A_1^T D_1 A_1 \\ A_1^T D_1 A_1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} A_2^T D_2 \mathbf{b}_2 \\ A_1^T D_1 \mathbf{b}_1 \end{pmatrix}.$$

This system is easily seen to be the Lagrange multiplier conditions for the two-layered LLS problem: recall from Section 2 that the two-layered LLS problem is

$$\begin{array}{rl} \text{minimize} & \|D_2^{1/2}(A_2\mathbf{x}-\mathbf{b}_2)\|^2 \\ \text{subject to} & A_1^T D_1 A_1 \mathbf{x} = A_1^T D_1 \mathbf{b}_1. \end{array}$$

This is the correct limiting behavior: the WLS solution tends to the LLS solution as $\delta_2/\delta_1 \to 0$. An in-depth explanation of MINRES-L's convergence behavior follows.

5 Convergence Analysis for Two Layers

In this section we consider convergence of MINRES-L in the presence of roundoff error for the case $p = 2$. As mentioned in the introduction, we make a simplifying assumption concerning the effect of roundoff error in Krylov space methods. The assumption concerns either CG or MINRES applied to the symmetric linear system $M\mathbf{x} = \mathbf{c}$. In our use of these algorithms, there is no preconditioner, and the initial guess is $\mathbf{x}^{(0)} = \mathbf{0}$. Further, in our use of MINRES, $\mathbf{c}$ lies in the range-space of M (i.e., the system is consistent). In our use of CG, M is positive definite. With these restrictions in mind, our assumption about the effect of roundoff is that after a sufficient number of iterations, either method will compute an iterate $\hat{\mathbf{x}}$ satisfying

$$\|\mathbf{c} - M\hat{\mathbf{x}}\| \leq C\epsilon \cdot \|M\| \cdot \|\mathbf{x}\| \tag{11}$$

where C is a modest constant, ϵ is machine epsilon, and $\mathbf{x}$ is the true solution. (If multiple solutions exist, we take $\mathbf{x}$ to be the minimum-norm solution.)

As far as we know, this bound has not been rigorously proved, but it is related to a bound proved by Greenbaum [9] in the case of conjugate gradient. In particular, Greenbaum's result implies that (11) would hold for CG if we were guaranteed that the recursively updated residual drops to well below machine precision, which always happens in our test cases.

As for MINRES, less is known, but a bound like (11) is known to hold for GMRES implemented with Householder transformations [5]. GMRES is equivalent to MINRES augmented with a full reorthogonalization process. We are content to assert (11) for MINRES, with evidence coming from our computational experiments.

This bound sheds light on why MINRES-L can attain much better accuracy than CGNR. For CGNR, the error bound (11) implies that $\|A^T D\mathbf{b} - A^T DA\hat{\mathbf{x}}\|$ gets very small, where $\hat{\mathbf{x}}$ is the computed solution. This latter quantity is the same as $\|(A^T DA)(\mathbf{x} - \hat{\mathbf{x}})\|$. But recall that we are seeking a bound on the forward error, that is, on $\|\mathbf{x} - \hat{\mathbf{x}}\|$. In this case, the factor $(A^T DA)$ can greatly skew the norm when δ_2/δ_1 is close to zero, so there is no bound on $\|\mathbf{x} - \hat{\mathbf{x}}\|$ independent of δ_1/δ_2, that is, (5) is not expected to be satisfied by CGNR. This is confirmed by our computational experiments.

In contrast, an analysis of MINRES-L starting from (11) does yield the accuracy bound (5). We need the following preliminary lemma.

Lemma 1 *Let A be an $m \times n$ matrix of rank n and $\bar{A}$ an $r \times n$ submatrix of A. Suppose the linear system $\bar{A}^T \bar{D} \bar{A}\mathbf{x} = A^T\mathbf{c}$ is consistent. Here, $\mathbf{c}$ is a given vector, and $\bar{D}$ is a given diagonal positive definite matrix. Then for any solution $\mathbf{x}$,*

$$\|\bar{A}\mathbf{x}\| \leq \|\bar{D}^{-1}\| \cdot \bar{\chi}_A \cdot \|\mathbf{c}\| \tag{12}$$

and

$$\|\bar{A}\mathbf{x}\| \leq \|\bar{D}^{-1}\| \cdot \chi_A \cdot \|A^T\mathbf{c}\|. \tag{13}$$

Furthermore, there exists a solution $\mathbf{x}$ satisfying

$$\|\mathbf{x}\| \leq \|\bar{D}^{-1}\| \cdot \chi_A \bar{\chi}_A \cdot \|\mathbf{c}\|. \tag{14}$$

PROOF. First, note the following preliminary result. Let H, K be two symmetric $n \times n$ matrices such that H is positive semidefinite and K is positive definite. Let $\mathbf{b}$ be an n-vector in the range space of H. Then $(H + \epsilon K)^{-1}\mathbf{b}$ converges to a solution of $H\mathbf{x} = \mathbf{b}$ as $\epsilon \to 0^+$. This is proved by reducing to the diagonal case using simultaneous diagonalization of H, K.

Let D be the extension of $\bar{D}$ to an $m \times m$ diagonal matrix obtained by filling in zeros, so that $A^T DA = \bar{A}^T \bar{D} \bar{A}$. Since $A^T DA\mathbf{x} = A^T\mathbf{c}$ is consistent, the limit of $(A^T(D + \epsilon I)A)^{-1}A^T\mathbf{c}$ as $\epsilon \to 0^+$ is some solution $\mathbf{x}$ of $\bar{A}^T \bar{D} \bar{A}\mathbf{x} = A^T\mathbf{c}$, as noted in the preceding paragraph. Let M be an $m \times m$ diagonal matrix with 1's in diagonal positions corresponding to $\bar{D}$ and zeros elsewhere.

We have

$$\begin{aligned}
\|\bar{A}\mathbf{x}\| &= \|MA\mathbf{x}\| \\
&= \lim_{\epsilon\to 0^+} \|MA(A^T(D+\epsilon I)A)^{-1}A^T\mathbf{c}\| \\
&= \lim_{\epsilon\to 0^+} \|M(D+\epsilon I)^{-1}(D+\epsilon I)A(A^T(D+\epsilon I)A)^{-1}A^T\mathbf{c}\| \qquad (15) \\
&\le \lim_{\epsilon\to 0^+} \|M(D+\epsilon I)^{-1}\| \cdot \sup_{\epsilon>0} \|(D+\epsilon I)A(A^T(D+\epsilon I)A)^{-1}A^T\| \cdot \|\mathbf{c}\| \\
&\le \|\bar{D}^{-1}\| \cdot \bar{\chi}_A \cdot \|\mathbf{c}\|.
\end{aligned}$$

The last line was obtained by the transpose of (4). This proves (12). Note that this holds for all $\mathbf{x}$ satisfying $\bar{A}^T\bar{D}\bar{A}\mathbf{x} = A^T\mathbf{c}$, since this latter equation uniquely determines $\bar{A}\mathbf{x}$. Similarly, to demonstrate (13), we start from (15):

$$\begin{aligned}
\|\bar{A}\mathbf{x}\| &\le \lim_{\epsilon\to 0^+} \|M(D+\epsilon I)^{-1}(D+\epsilon I)A(A^T(D+\epsilon I)A)^{-1}A^T\mathbf{c}\| \\
&\le \lim_{\epsilon\to 0^+} \|M(D+\epsilon I)^{-1}\| \cdot \sup_{\epsilon>0} \|(D+\epsilon I)A(A^T(D+\epsilon I)A)^{-1}\| \cdot \|A^T\mathbf{c}\| \\
&\le \|\bar{D}^{-1}\| \cdot \chi_A \cdot \|A^T\mathbf{c}\|.
\end{aligned}$$

For the second part of the proof, observe by the first part that $A^T\mathbf{c} = \bar{A}^T\bar{D}\bar{A}\mathbf{x} = A^TDA\mathbf{x} = A^TDMA\mathbf{x}$. Hence,

$$\begin{aligned}
\mathbf{x} &= \lim_{\epsilon\to 0^+} (A^T(D+\epsilon I)A)^{-1}A^T\mathbf{c} \\
&= \lim_{\epsilon\to 0^+} (A^T(D+\epsilon I)A)^{-1}A(D+\epsilon I)MA\mathbf{x}
\end{aligned}$$

and thus

$$\begin{aligned}
\|\mathbf{x}\| &\le \sup_{\epsilon>0} \|(A^T(D+\epsilon I)A)^{-1}A(D+\epsilon I)\| \cdot \|MA\mathbf{x}\| \\
&\le \chi_A\|\bar{A}\mathbf{x}\|.
\end{aligned}$$

Combining this with (12) proves (14). ∎

To resume the analysis of MINRES-L, we define

$$\mathbf{r}_1 = A_2^TD_2A_2\hat{\mathbf{x}} + A_1^TD_1A_1\hat{\mathbf{v}} - A_2^TD_2\mathbf{b}_2, \text{ and} \qquad (16)$$

$$\mathbf{r}_2 = A_1^TD_1A_1\hat{\mathbf{x}} - (\delta_2/\delta_1)A_1^TD_1A_1\hat{\mathbf{v}} - A_1^TD_1\mathbf{b}_1, \qquad (17)$$

where $(\hat{\mathbf{x}};\hat{\mathbf{v}})$ is the solution computed by MINRES-L. Then (11) applied to (10) yields the bounds

$$\|\mathbf{r}_1\|, \|\mathbf{r}_2\| \le C\epsilon \cdot \|H_2\| \cdot \|(\mathbf{x};\mathbf{v})\|. \qquad (18)$$

In this formula, H_2 is shorthand for the coefficient matrix of (10).

We can extract another equation from (16) and (17); in particular, if we multiply (16) by δ_2, multiply (17) by δ_1 and then add, we eliminate the terms involving $\hat{\mathbf{v}}$:

$$\delta_2\mathbf{r}_1 + \delta_1\mathbf{r}_2 = \delta_1 A_1^T D_1 A_1\hat{\mathbf{x}} + \delta_2 A_2^T D_2 A_2\hat{\mathbf{x}} - \delta_1 A_1^T D_1\mathbf{b}_1 - \delta_2 A_2^T D_2\mathbf{b}_2.$$

Let $\mathbf{x}$ be the exact solution to the WLS problem. The last two terms of this equation can be replaced with terms involving $\mathbf{x}$ by using (7). Interchanging the left- and right-hand sides yields

$$\delta_1 A_1^T D_1 A_1(\hat{\mathbf{x}} - \mathbf{x}) + \delta_2 A_2^T D_2 A_2(\hat{\mathbf{x}} - \mathbf{x}) = \delta_2\mathbf{r}_1 + \delta_1\mathbf{r}_2. \tag{19}$$

The goal is to derive an accuracy bound like (5) from (18) and (19). We start by bounding the quantity on the right-hand side of (18). Note that $\|H_2\|$ can be bounded by $2\kappa\|A\|^2$ because the largest entries in D_1, D_2 are bounded by κ. We can bound $\|\mathbf{x}\|$ by $\chi_A\|\mathbf{b}\|$ using Theorem 1. Next we turn to bounding $\|\mathbf{v}\|$ in (18). Recall that, as mentioned in the preceding section, $\mathbf{v}$ is not uniquely determined, but MINRES will find the minimum-norm $\mathbf{v}$ satisfying (10). Recall that $\mathbf{v}$ is determined by the constraint

$$A_1^T D_1 A_1\mathbf{v} = A_1^T D_1\mathbf{b}_1 - A_1^T D_1 A_1\mathbf{x}.$$

One way to pick such a $\mathbf{v}$ is to make it minimize $\|A_2\mathbf{v}\|$ subject to the above constraint. In this case, $\mathbf{v}$ is a layered least-squares solution with right-hand side data $(\mathbf{b}_1 - A_1\mathbf{x}; \mathbf{0})$. Thus, Corollary 1 yields the bound

$$\begin{aligned} \|\mathbf{v}\| &\le \chi_A \cdot \|\mathbf{b}_1 - A_1\mathbf{x}\| \\ &\le \chi_A(\|\mathbf{b}\| + \bar{\chi}_A\|\mathbf{b}\|) \\ &= \chi_A(\bar{\chi}_A + 1)\|\mathbf{b}\| \end{aligned}$$

for this choice of $\mathbf{v}$. (The factor $\bar{\chi}_A + 1$ can be improved to $\bar{\chi}_A$ by using the analysis of Gonzaga and Lara [8].) Combining the $\mathbf{x}$ and $\mathbf{v}$ contributions means that we have bounded the right-hand side of (18); let us rewrite (18) with the new bound:

$$\|\mathbf{r}_1\|, \|\mathbf{r}_2\| \le 2C\epsilon \cdot \|A\|^2 \cdot \kappa \cdot \chi_A(\bar{\chi}_A + 2)\|\mathbf{b}\|. \tag{20}$$

Next, we write new equations for $\mathbf{r}_1, \mathbf{r}_2$. Observe that $\mathbf{r}_1$ lies in the range of A_1^T and A_2^T, so we can find $\mathbf{h}_1$ satisfying

$$\mathbf{r}_1 = A_1^T D_1 A_1\mathbf{h}_1 + A_2^T D_2 A_2\mathbf{h}_1. \tag{21}$$

Similarly, by (17) there exists $\mathbf{h}_2$ satisfying

$$\mathbf{r}_2 = A_1^T D_1 A_1 \mathbf{h}_2. \tag{22}$$

By applying (13) to $\mathbf{r}_1$ and $\mathbf{r}_2$ separately, with "$A^T\mathbf{c}$" in the lemma taken to be first $\mathbf{r}_1$ and then $\mathbf{r}_2$, we conclude from (21) and (22) that

$$\begin{aligned} \|[A_1; A_2]\mathbf{h}_1\| &\leq \chi_A \|\mathrm{diag}(D_1^{-1}, D_2^{-1})\| \cdot \|\mathbf{r}_1\|, \text{ and} \qquad (23) \\ \|A_1\mathbf{h}_2\| &\leq \chi_A \|D_1^{-1}\| \cdot \|\mathbf{r}_2\|. \qquad (24) \end{aligned}$$

Substituting (21) and (22) into (19) yields

$$\begin{aligned} \delta_1 A_1^T D_1 A_1(\hat{\mathbf{x}} - \mathbf{x}) + \delta_2 A_2^T D_2 A_2(\hat{\mathbf{x}} - \mathbf{x}) &= \delta_1 A_1^T D_1 A_1 \mathbf{h}_2 + \delta_2 A_1^T D_1 A_1 \mathbf{h}_1 \\ &\quad + \delta_2 A_2^T D_2 A_2 \mathbf{h}_1 \\ &= \delta_1 A_1^T D_1 (A_1\mathbf{h}_2 + (\delta_2/\delta_1) A_1 \mathbf{h}_1) \\ &\quad + \delta_2 A_2^T D_2 A_2 \mathbf{h}_1. \end{aligned}$$

Notice (by analogy with (7)) that the preceding equation is exactly a weighted least-squares computation where the "unknown" is $\hat{\mathbf{x}} - \mathbf{x}$ and the right-hand side data is $(A_1\mathbf{h}_2 + (\delta_2/\delta_1)A_1\mathbf{h}_1; A_2\mathbf{h}_1)$. Thus, by Theorem 1,

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \leq \chi_A \|(A_1\mathbf{h}_2 + (\delta_2/\delta_1)A_1\mathbf{h}_1; A_2\mathbf{h}_1)\|.$$

We now build a chain of inequalities: the right-hand side of the preceding inequality is bounded by (23) and (24), and the right-hand side of (23) and (24) is bounded by (20). Combining all of this yields

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \leq 4C\epsilon \cdot \chi_A^3 \|A\|^2 \cdot \kappa \cdot (\bar{\chi}_A + 2) \cdot \|\mathbf{b}\|. \tag{25}$$

To obtain the preceding inequality, we used the facts that $\delta_2/\delta_1 \leq 1$ (by assumption) and that $\|\mathrm{diag}(D_1^{-1}, D_2^{-1})\| \leq 1$ (also by assumption, since the smallest entry in each D_i is taken to be 1).

Thus, we have an error bound of the form (5) as desired; in particular, there is no dependence of the error bound on δ_2/δ_1. Note that this bound depends on κ. Recall that κ is defined to be the maximum entry in $D_1, \ldots, D_p$ and is assumed to be small. Indeed, as noted in Section 2, we can always assume that $\kappa = 1$ if we are willing to divide the problem into many layers.

6 MINRES-L for p Layers

In this section we present the MINRES-L algorithm for the p-layered WLS problem. The algorithm is the application of MINRES to the symmetric linear system $H_p\mathbf{w} = \mathbf{c}_p$, where H_p is a square matrix of size $(1 + p(p-1)/2)n \times (1+p(p-1)/2)n$, $\mathbf{c}_p$ is a vector of that order, and $\mathbf{w}$ is the vector of unknowns. Matrix H_p is partitioned into $(1+p(p-1)/2) \times (1+p(p-1)/2)$ blocks each of size $n \times n$. Vectors $\mathbf{c}_p$ and $\mathbf{w}$ are similarly partitioned. The WLS solution vector is the first subvector of $\mathbf{w}$.

In more detail, the vector $\mathbf{w}$ is composed of $\mathbf{x}$ concatenated with $p(p-1)/2$ n-vectors that we denote $\mathbf{v}_{i,j}$, where i lies in $2,\ldots,p$ and j lies in $1,\ldots,i-1$. Recall that the p-layered WLS problem may be written

$$\delta_1 A_1^T D_1 A_1 \mathbf{x} + \cdots + \delta_p A_p^T D_p A_p \mathbf{x} = \delta_1 A_1^T D_1 \mathbf{b}_1 + \cdots + \delta_p A_p^T D_p \mathbf{b}_p. \tag{26}$$

Let $\mathbf{x}$ be the solution to this equation. Then we see from this equation that $A_p^T D_p A_p \mathbf{x} - A_p^T \mathbf{b}_p$ lies in the span of $[A_1^T,\ldots,A_{p-1}^T]$. Therefore, there exists a solution $[\mathbf{v}_{p,p-1};\ldots;\mathbf{v}_{p,1}]$ to the equation

$$A_p^T D_p A_p \mathbf{x} + A_{p-1}^T D_{p-1} A_{p-1} \mathbf{v}_{p,p-1} + \cdots + A_1^T D_1 A_1 \mathbf{v}_{p,1} = A_p^T D_p \mathbf{b}. \tag{27}$$

This equation is the first block-row of $H_p\mathbf{w} = \mathbf{c}_p$. In other words, the first block row of H_p contains one copy of each of the matrices $A_i^T D_i A_i$, and the first block of $\mathbf{c}_p$ is $A_p^T D_p \mathbf{b}$.

In general, the $(p-i+1)$th block-row of $H_p\mathbf{w} = \mathbf{c}_p$, for $i = 1,\ldots,p$, is the equation

$$A_i^T D_i A_i \mathbf{x} + \sum_{j=1}^{i-1} A_j^T D_j A_j \mathbf{v}_{i,j} - \sum_{j=i+1}^{p} \frac{\delta_j}{\delta_i} A_i^T D_i A_i \mathbf{v}_{j,i} = A_i^T D_i \mathbf{b}_i. \tag{28}$$

This completes the description of block-rows $1,\ldots,p$ of $H_p\mathbf{w} = \mathbf{c}_p$. We now establish some properties of these block-rows, and we postpone the description of block-rows $p+1,\ldots,1+p(p-1)/2$.

Lemma 2 *Suppose* $\mathbf{w}$ *is a solution to the linear equation* (28) *for each* $i = 1,\ldots,p$, *where* $\mathbf{w}$ *denotes the concatenation of* $\mathbf{x}$ *and all of the* $\mathbf{v}_{i,j}$*'s. Then* $\mathbf{x}$ *is the solution to the WLS problem* (26).

Proof. For each i, multiply (28) by δ_i and then sum all p equations obtained in this manner. Observe that all the $\mathbf{v}_{i,j}$ terms cancel out and we end up exactly with (26). ∎

We also need the converse to be true.

Lemma 3 *Suppose* $\mathbf{x}$ *is the solution to* (26). *Then there exist vectors* $\mathbf{v}_{i,j}$ *for* $1 \le j < i \le p$ *such that* (28) *is satisfied for each* $i = 1, \ldots, p$.

Proof. The proof is by induction on (decreasing) $k = p, \ldots, 1$. We assume that we have already determined $\mathbf{v}_{i,j}$ for all $i = k+1, \ldots, p$ and all $j = 1, \ldots, i-1$ so that (28) is satisfied for $i = k+1, \ldots, p$, and now we must determine $\mathbf{v}_{k,j}$ for $j = 1, \ldots, i-1$ to satisfy (28) for the particular value $i = k$. The base case of the induction is that we can select $\mathbf{v}_{p,1}, \ldots, \mathbf{v}_{p,p-1}$ to satisfy (28) in the case $i = p$ because, as noted above, $A_p^T D_p A_p \mathbf{x} - A_p^T \mathbf{b}_p$ lies in the range of $[A_1^T, \ldots, A_{p-1}^T]$ because of (26).

Now for the induction case of $k < p$. Rewrite (28) for the case $k = i$, and multiply through by δ_k:

$$\delta_k A_k^T D_k A_k \mathbf{x} + \delta_k \sum_{j=1}^{k-1} A_j^T D_j A_j \mathbf{v}_{k,j} - \sum_{j=k+1}^{p} \delta_j A_k^T D_k A_k \mathbf{v}_{j,k} = \delta_k A_k^T D_k \mathbf{b}_k. \tag{29}$$

Recall that our goal is to choose $\mathbf{v}_{k,j}$ for $j = 1, \ldots, k-1$ to make this equation valid.

Multiply (28) for each $i = k+1, \ldots, p$ by δ_i and add this to (29). After rearranging and summations and cancelling common terms on the left-hand side, we end up with

$$\sum_{i=k}^{p} \delta_i A_i^T D_i A_i \mathbf{x} + \sum_{i=k}^{p} \sum_{j=1}^{k-1} \delta_i A_j^T D_j A_j \mathbf{v}_{i,j} = \sum_{i=k}^{p} \delta_i A_i^T D_i \mathbf{b}_i. \tag{30}$$

Dividing through by δ_k and separating out the $\mathbf{v}_{k,j}$ terms from the second summation yields:

$$\begin{aligned} &A_1^T D_1 A_1 \mathbf{v}_{k,1} + \cdots + A_{k-1}^T D_{k-1} A_{k-1} \mathbf{v}_{k,k-1} \\ &\qquad = \sum_{i=k}^{p} \frac{\delta_i}{\delta_k} A_i^T D_i (\mathbf{b}_i - A_i \mathbf{x}) - \sum_{i=k+1}^{p} \sum_{j=1}^{k-1} \frac{\delta_i}{\delta_k} A_j^T D_j A_j \mathbf{v}_{i,j}. \end{aligned} \tag{31}$$

But from (26) we know that $\sum_{i=k}^{p} \delta_i A_i^T D_i (\mathbf{b}_i - A_i \mathbf{x})$ lies in the range of $[A_1^T, \ldots, A_{k-1}^T]$. Clearly the rightmost summation of (31) also lies in the same range. Therefore, there exist $\mathbf{v}_{k,j}$ for $j = 1, \ldots, k-1$ to make (31) valid. But then these same choices will make (29) valid because the algebraic steps used to derive (31) from (29) can be reversed. This proves the lemma. ∎

Note that the preceding proof actually demonstrates a strengthened version of the lemma. The strengthened version states that if we are given $\mathbf{x}$ satisfying (26) and, for some k, vectors $\mathbf{v}_{i,j}$ for $k \leq j < i \leq p$ that satisfy (28) for all $i = k, \ldots, p$, then we can extend the given data to a solution of (28) for all $i = 1, \ldots, p$. This strengthened version is needed below.

We now explain the remaining $p(p-1)/2$ block-rows of H_p. These rows exist solely for the purpose of making H_p symmetric. First, we have to order the variables and equations correctly. The variables will be listed in the order $(\mathbf{x}; \mathbf{v}_{p,p-1}; \mathbf{v}_{p,p-2}; \ldots; \mathbf{v}_{p,1}; \mathbf{v}_{p-1,p-2}; \ldots; \mathbf{v}_{p-1,1}; \ldots; \mathbf{v}_{2,1})$. The first p equations will be listed in the order (28) for $i = p, p-1, \ldots, 1$. This means that the first p rows of H_p have the format $[S_p, T_p]$, where S_p is a $p \times p$ matrix and T_p is a $p \times (p-1)(p-2)/2$ matrix. Furthermore, it is easily checked that S_p is symmetric: its first block-row and first block-column both consist of $A_i^T D_i A_i$ listed in the order $i = p, \ldots, 1$; the $(p-i+1)$st entry of its main diagonal is $-(\delta_p/\delta_i) A_i^T D_i A_i$ for $i = 1, \ldots, p-1$; and all its other blocks are zeros. Then we define H_p to be

$$H_p = \begin{pmatrix} S_p & T_p \\ T_p^T & 0 \end{pmatrix}.$$

We define $\mathbf{c}_p$ as

$$\mathbf{c}_p = \begin{pmatrix} A_p^T D_p \mathbf{b}_p \\ \vdots \\ A_1^T D_1 \mathbf{b}_1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix},$$

where there are $p(p-1)/2$ blocks of zeros. For example, the following linear system is $H_3\mathbf{w} = \mathbf{c}_3$:

$$\begin{pmatrix} A_3^T D_3 A_3 & A_2^T D_2 A_2 & A_1^T D_1 A_1 & 0 \\ A_2^T D_2 A_2 & -\frac{\delta_3}{\delta_2} A_2^T D_2 A_2 & 0 & A_1^T D_1 A_1 \\ A_1^T D_1 A_1 & 0 & -\frac{\delta_3}{\delta_1} A_1^T D_1 A_1 & -\frac{\delta_2}{\delta_1} A_1^T D_1 A_1 \\ 0 & A_1^T D_1 A_1 & -\frac{\delta_2}{\delta_1} A_1^T D_1 A_1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{v}_{3,2} \\ \mathbf{v}_{3,1} \\ \mathbf{v}_{2,1} \end{pmatrix} = \begin{pmatrix} A_3^T D_3 \mathbf{b}_3 \\ A_2^T D_2 \mathbf{b}_2 \\ A_1^T D_1 \mathbf{b}_1 \\ \mathbf{0} \end{pmatrix}.$$

We now must consider whether $H_p\mathbf{w} = \mathbf{c}_p$ has any solutions; in particular, we must demonstrate that the new group of equations $T_p^T\mathbf{w}' = \mathbf{0}$ is consistent with the first p rows. Here $\mathbf{w}'$ denotes the first p blocks of $\mathbf{w}$, that is, $\mathbf{w}' = (\mathbf{x}; \mathbf{v}_{p,p-1}; \ldots ; \mathbf{v}_{p,1})$. Studying the structure of T_p, we see that there are $(p-1)(p-2)/2$ block-rows of T_p^T indexed by (i,j) for $1 \le j < i \le p-1$ (in correspondence with the columns of T_p, which correspond to variables $\mathbf{v}_{i,j}$ for i,j in that range). The row indexed by (i,j) has exactly two nonzero block entries that yield the equation

$$A_j^T D_j A_j \mathbf{v}_{p,i} - \frac{\delta_i}{\delta_j} A_j^T D_j A_j \mathbf{v}_{p,j} = \mathbf{0}. \tag{32}$$

Our task is therefore to show that we can simultaneously satisfy (28) for $i = 1, \ldots, p$ and (32) for (i,j) such that $1 \le j < i \le p-1$.

Our approach is to select the $\mathbf{v}_{p,j}$'s in the order $\mathbf{v}_{p,p-1}, \mathbf{v}_{p,p-2}, \ldots, \mathbf{v}_{p,1}$. In particular, assuming $\mathbf{v}_{p,j+1}, \ldots, \mathbf{v}_{p,p-1}$ are already selected, we define $\mathbf{v}_{p,j}$ to be any solution to

$$\sum_{k=1}^{j} \delta_k A_k^T D_k A_k \mathbf{v}_{p,j} = \delta_j \left(A_p^T D_p \mathbf{b}_p - A_p^T D_p A_p \mathbf{x} - \sum_{k=j+1}^{p-1} A_k^T D_k A_k \mathbf{v}_{p,k} \right). \tag{33}$$

The following lemma shows that this linear system is consistent.

Lemma 4 *If the $\mathbf{v}_{p,j}$'s are chosen in reverse order to satisfy* (33)*, then at each step the linear system is consistent, and* (32) *is satisfied.*

PROOF. The proof is by reverse induction on j. The base case is $j = p-1$, in which case (33) has a solution because, as noted above, $A_p^T D_p \mathbf{b}_p - A_p^T D_p A_p \mathbf{x}$

lies in the span of $[A_1^T, \ldots, A_{p-1}^T]$. In the case $j = p-1$, (32) is vacuously true: there is no i in the specified range.

Now consider the case $j < p-1$. Pick any i in the range $j+1, \ldots, p-1$. Start with the version of (33) satisfied by $\mathbf{v}_{p,i}$, which holds by the induction hypothesis:

$$\sum_{k=1}^{i} \delta_k A_k^T D_k A_k \mathbf{v}_{p,i} = \delta_i \left(A_p^T D_p \mathbf{b}_p - A_p^T D_p A_p \mathbf{x} - \sum_{k=i+1}^{p-1} A_k^T D_k A_k \mathbf{v}_{p,k} \right).$$

Move the terms $k = j+1, \ldots, i$ of the first summation to the right-hand side:

$$\begin{aligned}
\sum_{k=1}^{j} \delta_k A_k^T D_k A_k \mathbf{v}_{p,i} &= \delta_i \left(A_p^T D_p \mathbf{b}_p - A_p^T D_p A_p \mathbf{x} - \sum_{k=j+1}^{i} \frac{\delta_k}{\delta_i} A_k^T D_k A_k \mathbf{v}_{p,i} \right. \\
&\qquad \left. - \sum_{k=i+1}^{p-1} A_k^T D_k A_k \mathbf{v}_{p,k} \right) \\
&= \delta_i \left(A_p^T D_p \mathbf{b}_p - A_p^T D_p A_p \mathbf{x} - \sum_{k=j+1}^{i} A_k^T D_k A_k \mathbf{v}_{p,k} \right. \\
&\qquad \left. - \sum_{k=i+1}^{p-1} A_k^T D_k A_k \mathbf{v}_{p,k} \right) \\
&= \delta_i \left(A_p^T D_p \mathbf{b}_p - A_p^T D_p A_p \mathbf{x} - \sum_{k=j+1}^{p-1} A_k^T D_k A_k \mathbf{v}_{p,k} \right).
\end{aligned}$$

The second line was obtained from the first by applying (32) inductively (with "j" in (32) taken to be k). The third line was obtained by merging the two summations on the right.

But notice that the preceding equation means that $\mathbf{v}_{p,i}$ satisfies the same linear system as $\mathbf{v}_{p,j}$, that is (33), except with the right-hand side scaled by δ_i/δ_j. This proves that (33) is consistent for the j case since we have constructed a solution to it. Although this linear system does not necessarily have a unique solution, a linear system of the form $A^T A\mathbf{x} = \mathbf{b}$ uniquely determines $A\mathbf{x}$. Thus, we have also proved that $\delta_j A_k^T D_k A_k \mathbf{v}_{p,i} = \delta_i A_k^T D_k A_k \mathbf{v}_{p,j}$ for all $k = 1, \ldots, j$. This result is actually a strengthening of (32) for j; for that equation we need only the specific case of $k = j$. ∎

The reader may have noticed that the preceding proof is apparently too complicated and that we could establish the result more simply by solving for $\mathbf{v}_{p,p-1}$ in (33) with $j = p-1$, and then setting $\mathbf{v}_{p,j} = (\delta_j/\delta_{p-1})\mathbf{v}_{p,p-1}$ for $j = 1, \ldots, p-2$. This simpler approach does not yield the bounds on $\|\mathbf{v}_{p,j}\|$ needed in the next section.

This proof shows that the above method for selecting $\mathbf{v}_{p,1}, \ldots, \mathbf{v}_{p,p-1}$ is consistent and satisfies (32). We also see that (27) is satisfied; this follows immediately from taking $j = 1$ in (33). To complete the proof that there is a solution to $H_p\mathbf{w} = \mathbf{c}_p$, we need only verify (28) in the case $i = p-2, \ldots, 1$. But recall from the proof of Lemma 3 that the remaining $\mathbf{v}_{i,j}$'s for $i = p-2, \ldots, 1$ can be determined sequentially by using the construction in the proof. Thus, the arguments of this section have established the following theorem.

Theorem 2 *There exists at least one solution* $\mathbf{w}$ *to* $H_p\mathbf{w} = \mathbf{c}_p$*, and furthermore, any such solution has as its first* n *entries the vector* $\mathbf{x}$ *that solves* (26).

7 Convergence Analysis for p Layers

The convergence analysis for p layers follows the same basic outline as the convergence analysis for two layers. In particular, we use (11) as the starting point for the error analysis. Observe that (11) has the norm of the true solution on the right-hand side. Thus, to apply that bound, we must get a norm bound on $\mathbf{v}_{i,j}$ for all i, j satisfying $1 \leq j < i \leq p$.

We start with bounds on $\mathbf{v}_{p,j}$ for $j = p-1, p-2, \ldots, 1$. Apply Lemma 1 to (33) in the case $j = p-1$. In the lemma, take $\bar{A} = [A_1; \ldots; A_{p-1}]$ and $\bar{D} = \operatorname{diag}(\delta_1 D_1, \ldots, \delta_{p-1}D_{p-1})$. As noted above, $A_p^T D_p \mathbf{b}_p - A_p^T D_p A_p \mathbf{x}$ lies in the range of $[A_1^T, \ldots, A_{p-1}^T]$ so (33) is consistent. The right-hand side of (33) in the $j = p-1$ case has the form $A^T\mathbf{c}$ with $\mathbf{c} = \delta_{p-1}[\mathbf{0}; \cdots; \mathbf{0}; D_p(\mathbf{b}_p - A_p\mathbf{x})]$. Note that $\|D_p(\mathbf{b}_p - A_p\mathbf{x})\|$ is bounded by $\kappa(\bar{\chi}_A + 1)\|\mathbf{b}\|$. Thus, from (12),

$$\begin{aligned}
\|[A_1; \ldots; A_{p-1}]\mathbf{v}_{p,p-1}\| &\leq \|\operatorname{diag}(\delta_1 D_1, \ldots, \delta_{p-1}D_{p-1})^{-1}\| \\
&\quad \cdot \bar{\chi}_A \cdot \delta_{p-1}\kappa(\bar{\chi}_A + 1)\|\mathbf{b}\| \\
&= \|\operatorname{diag}((\delta_{p-1}/\delta_1)D_1^{-1}, \ldots, (\delta_{p-1}/\delta_{p-1})D_{p-1}^{-1})\| \\
&\quad \cdot \kappa\bar{\chi}_A(\bar{\chi}_A + 1)\|\mathbf{b}\| \\
&\leq \kappa\bar{\chi}_A(\bar{\chi}_A + 1)\|\mathbf{b}\|. \qquad (34)
\end{aligned}$$

To derive the third line from the second, we used the facts that $\|D_i^{-1}\| \leq 1$ for each i and $\delta_i/\delta_j \leq 1$ for $i \geq j$.

Now we use the same line of reasoning to get a bound on $\mathbf{v}_{p,p-2}$ based on (33) for the case $j = p - 2$. In this case, the right-hand side of (33) has the form $A^T\mathbf{c}$, where

$$\mathbf{c} = \delta_{p-2}[\mathbf{0}; \cdots ; \mathbf{0}; -D_{p-1}A_{p-1}\mathbf{v}_{p,p-1}; D_p(\mathbf{b}_p - A_p\mathbf{x})].$$

Thus, $\|\mathbf{c}\|$ is bounded by $\delta_{p-2}(\kappa(\bar{\chi}_A + 1)\|\mathbf{b}\| + \kappa^2\bar{\chi}_A(\bar{\chi}_A + 1)\|\mathbf{b}\|)$, which is at most $2\delta_{p-2}\kappa^2\bar{\chi}_A(\bar{\chi}_A + 1)\|\mathbf{b}\|$.

We continue this argument inductively. Each time the bound grows by a factor $2\kappa\bar{\chi}_A$ to take into account the fact that $\mathbf{v}_{p,i}$ appears on the right-hand side for the equation determining $\mathbf{v}_{p,i-1}$. In the end we conclude that

$$\|[A_1; \cdots ; A_i]\mathbf{v}_{p,i}\| \le (2\kappa\bar{\chi}_A)^{p-i}(\bar{\chi}_A + 1)\|\mathbf{b}\|. \tag{35}$$

Next we must bound $\mathbf{v}_{i,j}$ for $1 \le j < i \le p - 1$. These vectors are determined by (28). We can find a solution to (28) by first solving

$$\bar{A}^T\bar{D}\bar{A}\mathbf{z}_i = A_i^T D_i\left(\mathbf{b}_i - A_i\mathbf{x} + \sum_{j=i+1}^{p}\frac{\delta_j}{\delta_i}A_i\mathbf{v}_{j,i}\right),$$

for $\mathbf{z}_i$, where $\bar{A} = [A_1; \ldots ; A_{i-1}]$, $\bar{D} = \mathrm{diag}(D_1, \ldots , D_{i-1})$. This equation is already known to be consistent. Furthermore, $\delta_j/\delta_i \le 1$ in the preceding equation. We set $\mathbf{v}_{i,1} = \cdots = \mathbf{v}_{i,i-1} = \mathbf{z}_i$. Using (12), we conclude that

$$\|\bar{A}\mathbf{v}_{i,k}\| \le \bar{\chi}_A\kappa\left(\|\mathbf{b}\| + \bar{\chi}_A\|\mathbf{b}\| + \sum_{j=i+1}^{p}\|A_i\mathbf{v}_{j,i}\|\right) \tag{36}$$

for each $k = 1, \ldots , j$.

We now claim that

$$\|[A_1; \cdots ; A_j]\mathbf{v}_{i,j}\| \le (4\kappa\bar{\chi}_A)^{p-i}(\bar{\chi}_A + 1)\|\mathbf{b}\|$$

for $1 \le j < i < p$. This is proved by induction on decreasing i using recurrence (36). The $j = p$ term on the right-hand side of (36) is bounded by (35), and the remaining terms are bounded by the induction hypothesis. We omit the details.

For the right-hand side of (11) we need a bound on $\|\mathbf{v}_{i,j}\|$. Note that up to now we have not uniquely determined $\mathbf{v}_{i,j}$ itself. Recall that in each case

Lemma 1 was used to bound $\|A_k \mathbf{v}_{i,j}\|$. We can force unique determination by choosing the $\mathbf{v}_{i,j}$ as in the proof of Lemma 1, yielding

$$\|\mathbf{v}_{i,j}\| \leq (4\kappa\bar{\chi}_A)^{p-i}(\bar{\chi}_A + 1)\chi_A\|\mathbf{b}\| \tag{37}$$

by (14). Note that MINRES does not necessarily select this $\mathbf{v}_{i,j}$, but because of its minimization property (that is, Theorem 2.4 of Brown and Walker [4] described in Section 4), it will select $\mathbf{v}_{i,j}$ whose norm is no larger than in the preceding bound.

We now can apply (11). The other factor on the right-hand side, namely, $\|H_p\|$, is easily seen to be bounded by $p^2\|A\|^2\kappa$. Let $\hat{\mathbf{w}}$ be the solution computed by MINRES-L, and let $\mathbf{r} = H_p\hat{\mathbf{w}} - \mathbf{c}_p$, i.e., $\mathbf{r} = H_p\hat{\mathbf{w}} - H_p\mathbf{w}$. Then, substituting (37) on the right-hand side of (11) yields

$$\|\mathbf{r}\| \leq C\epsilon p^4\|A\|^2 \cdot \kappa(\bar{\chi}_A + 1)\chi_A \cdot (4\kappa\bar{\chi}_A)^{p-1} \cdot \|\mathbf{b}\|. \tag{38}$$

Let $\mathbf{r}_p, \ldots, \mathbf{r}_1$ be the first p block-entries of $\mathbf{r}$. Note that $\mathbf{r}_j$ must lie in the span of $[A_1^T, \ldots, A_j^T]$ in order for the equation $H_p(\hat{\mathbf{w}} - \mathbf{w}) = \mathbf{r}$ to have a solution, because it can be seen from (28) that the $(p-i+1)$st block-row of H_p involves only $A_1, \ldots, A_i$. Thus, let us find $\mathbf{h}_i$ that solves $\mathbf{r}_i = A_1^T D_1 A_1 \mathbf{h}_i + \cdots + A_i^T D_i A_i \mathbf{h}_i$ for each i. By (13) we know that $\|[A_1; \ldots; A_i]\mathbf{h}_i\| \leq \chi_A\|\mathbf{r}_i\|$.

Let $\hat{\mathbf{x}}$ be the first n entries of $\hat{\mathbf{w}}$, that is, the computed WLS solution. If we multiply the $(p-i+1)$st block row of $H_p(\hat{\mathbf{w}} - \mathbf{w}) = \mathbf{r}$ by δ_i for $i = 1, \ldots, p$ and add these p rows, we obtain

$$\begin{aligned}
\sum_{i=1}^{p} \delta_i A_i^T D_i A_i(\hat{\mathbf{x}} - \mathbf{x}) &= \sum_{i=1}^{p} \delta_i \mathbf{r}_i \\
&= \sum_{i=1}^{p} \delta_i \left(\sum_{j=1}^{i} A_j^T D_j A_j \mathbf{h}_i\right) \\
&= \sum_{i=1}^{p} \delta_i A_i^T D_i A_i \sum_{j=i}^{p} \frac{\delta_j}{\delta_i}\mathbf{h}_j.
\end{aligned}$$

The third line was obtained from the second by interchanging the order of summation. Thus, we see from the third line above that $\hat{\mathbf{x}} - \mathbf{x}$ solves a WLS problem in which the ith entry of the data vector is $A_i \sum_{j=i}^{p} \frac{\delta_j}{\delta_i}\mathbf{h}_j$. Since $\delta_j/\delta_i \leq 1$ for i, j in this range, we conclude that the data vector is bounded in norm by $p^2 \max_{i<j} \|A_i \mathbf{h}_j\|$, that is, by $p^2\chi_A \max_i \|\mathbf{r}_i\|$. Then Theorem 1 implies that

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \leq p^2\chi_A^2 \max_i \|\mathbf{r}_i\|.$$

Substituting (38) yields

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \leq C\epsilon p^6 \|A\|^2 \cdot \kappa(\bar{\chi}_A + 1)\chi_A^3 \cdot (4\kappa\bar{\chi}_A)^{p-1}\|\mathbf{b}\|. \tag{39}$$

This is a bound of the form (5) as desired.

8 Computational Experiments

In this section we present computational experiments on MINRES-L and CGNR to compare their accuracy and efficiency. The first few tests involve a small node-arc adjacency matrix. The remaining tests are on matrices arising in linear programming and boundary value problems. All tests were conducted in Matlab 4.2 running on an Intel Pentium under Microsoft Windows NT 4.0. Matlab is a software package and programming language for numerical computation written by The Mathworks, Inc. All computations are in IEEE double precision with machine epsilon approximately $2.2 \cdot 10^{-16}$. Matlab sparse matrix operations were used in all tests.

Our implementation of CGNR is based on CGLS1 as in (3.2) of Björck, Elfving and Strakoš [2]. These authors conclude that CGLS1 is a good way to organize CGNR. There are two matrix-vector products per CGLS1 iteration, one with matrix $A^TD^{1/2}$ and one with $D^{1/2}A$. In our implementation, the CGNR iteration terminates when the scaled computed residual $\|\mathbf{s}_k\|/\|A^TD\mathbf{b}\|$ drops below 10^{-13}. Our implementation of MINRES is based on [14], except Givens rotations were used instead of 2×2 Householder matrices (so that there are some inconsequential sign differences). The MINRES-L iteration terminates when the scaled computed residual $\|\mathbf{r}_k\|/\|[A_1^TD_1\mathbf{b}_1; \ldots ; A_p^TD_p\mathbf{b}_p]\|$ drops below 10^{-13}. e

The first matrix A used in the following tests is the reduced node-arc adjacency matrix of the graph depicted in Figure 1. A "node-arc adjacency" matrix contains one column for each node of a graph and one row for each edge. Each row contains exactly two nonzero entries, a $+1$ and a -1 in the columns corresponding to the endpoints of the edge. (The choice of which endpoint is assigned $+1$ and which is assigned -1 induces an orientation on the edge, but often this orientation is irrelevant for the application.) A reduced node-arc incidence (RNAI) matrix is obtained from a node-arc incidence matrix by deleting one column. RNAI matrices arise in the analysis of an electrical network with batteries and resistors; see [23]. They also arise in network flow problems. In the case of Figure 1, the column corresponding to

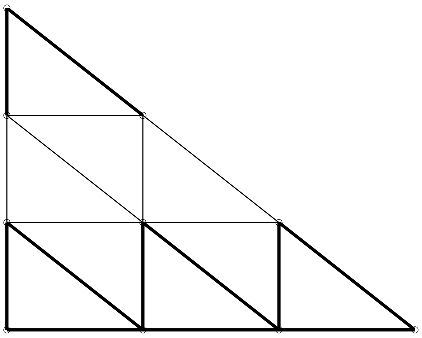


Figure 1: An 18×9 RNAI matrix based on this graph was used for the first group of tests. The column corresponding to the top node is deleted. Edges marked with heavy lines are weighted 1, and edges marked with light lines are weighted δ_2, where δ_2 varies from test to test.

the top node was deleted. Thus, A is an 18×9 matrix. It is well known that the RNAI matrix for a connected graph always has full rank. RNAI matrices are known to have small values of χ_A and $\bar{\chi}_A$ [23].

In all these tests, the weight matrix has two layers. We took $D_1 = I$, $D_2 = I$, and $\delta_1 = 1$, while we let δ_2 vary from experiment to experiment. The rows of A in correspondence with D_2 are drawn as thinner lines in Figure 1. Finally, the right-hand side $\mathbf{b}$ was chosen to be the first 18 prime numbers.

The results are displayed in Table 1, and the cases when $\delta_2 = 10^{-6}$ and $\delta_2 = 10^{-12}$ are plotted in Figure 2. The *scaled error* that is tabulated and plotted in all cases is defined to be $\|\hat{\mathbf{x}} - \mathbf{x}\|/\|\mathbf{b}\|$. We choose this particular scaling for the error because our goal is to investigate stability bound (5). The true solution $\mathbf{x}$ is computed using the COD method [12]. Note that the accuracy of CGNR decays as δ_2 gets smaller, whereas MINRES-L's accuracy stays constant. MINRES-L requires many more flops than CGNR because the system matrix is larger. The running time of CGNR is about the same for the first four rows of the table as the ill-conditioning increases. In the last two rows the running time of CGNR drops because the matrix $A^T DA$ masquerades as a low-rank matrix for small values of δ_2, causing early termination of the Lanczos process.

Besides returning an inaccurate solution, CGNR has the additional difficulty that its residual (the quantity normally measured in practical use of this algorithm) does not reflect the forward error, so there is no simple way

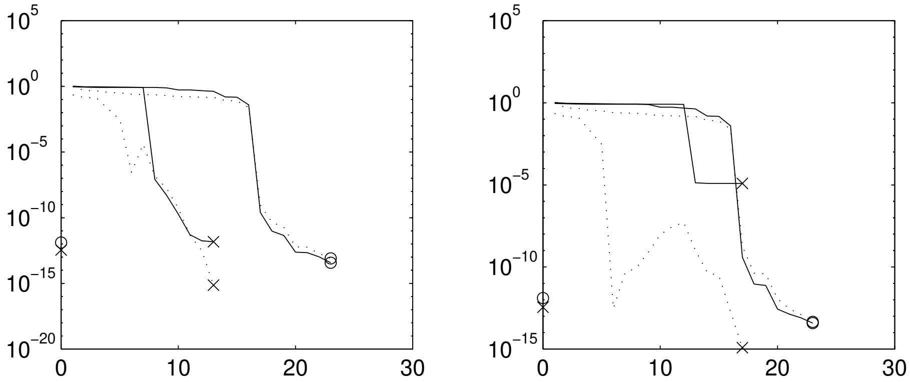
Table 1: Behavior of the two-layered MINRES-L algorithm compared to CGNR for decreasing values of δ_2. The error reported is the scaled error defined in the text. Note that the CG accuracy degrades while the MINRES-L accuracy stays about the same.

δ_2	MINRES-L Flops	MINRES-L Iterations	MINRES-L Error	CGNR Flops	CGNR Iterations	CGNR Error
10^{-3}	15032	23	1.9e-14	3479	11	3.0e-14
10^{-6}	15032	23	3.8e-14	4085	13	1.5e-12
10^{-9}	14387	22	2.7e-14	4085	13	7.9e-9
10^{-12}	15032	23	3.8e-14	5297	17	1.2e-5
10^{-15}	15032	23	3.7e-14	1964	6	8.2e-1
10^{-18}	15032	23	4.2e-14	1064	6	8.2e-1

to determine whether CGNR is computing good answers. In contrast, the error and residual in MINRES-L are closely correlated. This correlation is predicted by our theory.

The next computational test involved a larger matrix A taken from the Netlib linear programming test set, namely, the matrix in problem AFIRO, which is 51×27. We used a matrix D with 1's in its first 27 diagonal positions and 10^{-12} in its remaining 24 positions (i.e., $D_1 = I$, $D_2 = I$, $p = 2$, $\delta_1 = 1$, $\delta_2 = 10^{-12}$). The right-hand side vector $\mathbf{b}$ was chosen to contain the first 51 primes. MINRES-L required 137 iterations and 250 kflops and yielded a solution $\hat{\mathbf{x}}$ with scaled error $3.0 \cdot 10^{-12}$ with respect to the true solution computed by the COD method. For this matrix, χ_A and $\bar{\chi}_A$ are not known. CGNR on this problem required 69 iterations and 61 kflops and returned an answer with scaled error $2.2 \cdot 10^{-3}$. The convergence plots are depicted in Figure 3.

The excessive number of iterations required by MINRES is apparently caused by a loss of orthogonality in the Lanczos process. To verify this hypothesis, we ran GMRES on the same layered matrix. GMRES [19] on a symmetric matrix is equivalent to MINRES with full reorthogonalization. (In exact arithmetic the two algorithms are identical.) We call this algorithm GMRES-L. The same termination tests were used. The result is depicted in Figure 4. In this case, GMRES-L ran for 50 iterations (fewer than $(1 + p(p-1)/2)n = 54$) and returned a more accurate answer, one with forward error $1.2 \cdot 10^{-14}$. However, the number of flops was higher, 350 k, because of the



Legend: —× = CGNR scaled error —○ = MINRES-L scaled error
⋯× = CGNR scaled residual ⋯○ = MINRES-L scaled residual

Figure 2: Convergence behavior of CGNR and MINRES-L for the 18×9 RNAI test case. The plots are for $\delta_2 = 10^{-6}$ (left) and $\delta_2 = 10^{-12}$ (right). In these plots and all that follow, the x-axis is the iteration number. For both algorithms the computed (i.e., recursively updated) residual is plotted rather than the true residual. Other experiments (not reported here) indicate that these are usually indistinguishable. The $\times$ on the y-axis indicates the cutoff below which the CGNR scaled residual must drop in order for (11) to be true with $\epsilon = 10^{-13}$. The $\circ$ on the y-axis is the analog for MINRES-L.

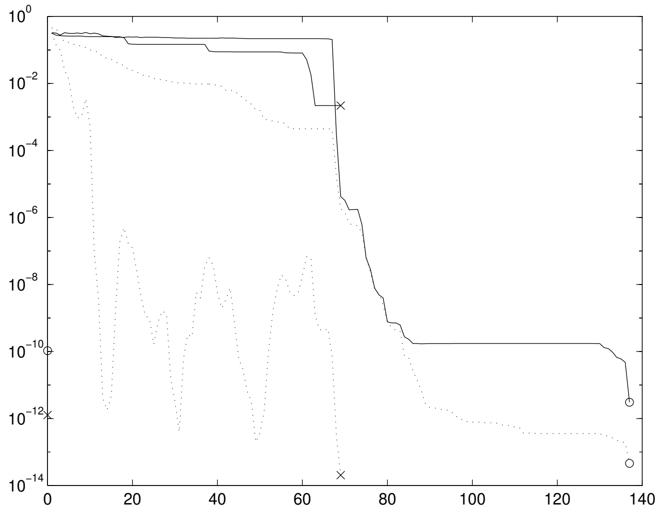


Figure 3: Convergence behavior of CGNR and MINRES-L for AFIRO. The curves are labeled as in Figure 2.

Gram-Schmidt process in the GMRES main loop.

The next computational test involves a larger matrix A arising from finite-element analysis. The application is the solution of the boundary value problem $\nabla \cdot (c\nabla u) = 0$ on the polygonal domain depicted in Figure 5 with Dirichlet boundary conditions. The conductivity field c is 1 on the outer part of the domain and is 10^{12} on the darker triangles. As discussed in [24], this type of problem gives rise to a weighted least-squares problem in which A encodes information about the geometry and D encodes the ill-conditioned conductivity field. The values of χ_A and $\bar{\chi}_A$ for this matrix are not known, although bounds are known for variants of these parameters. The particular matrix A is 652×136. The right-hand side vector $\mathbf{b}$ was chosen according to the Dirichlet boundary conditions described in [24]. The MINRES-L method for this problem gave scaled error of $1.3 \cdot 10^{-13}$ after 382 iterations and 6.5 mflops. To compute the true solution, we used the NSHI method in [24]. In this case, surprisingly, CGNR gave almost as accurate an answer, but the termination test was never activated. (We cut off CGNR after $10n$ iterations.) The residual of CGNR is quite oscillatory as depicted in Figure 6. In the finite-element literature, CGNR would be referred to as conjugate gradient on the *assembled stiffness matrix*, which is $A^T DA$.

A cause of this odd behavior of CGNR is as follows. Note that the region of high conductivity is not incident on the boundary of the domain so $\mathbf{b}_1 = \mathbf{0}$.

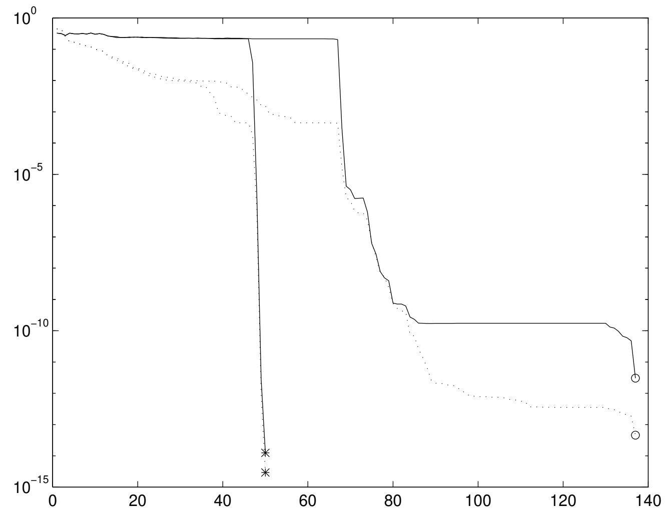


Figure 4: Convergence behavior of GMRES-L (—∗ and ⋯∗) and MINRES-L (—○ and ⋯○) for AFIRO.

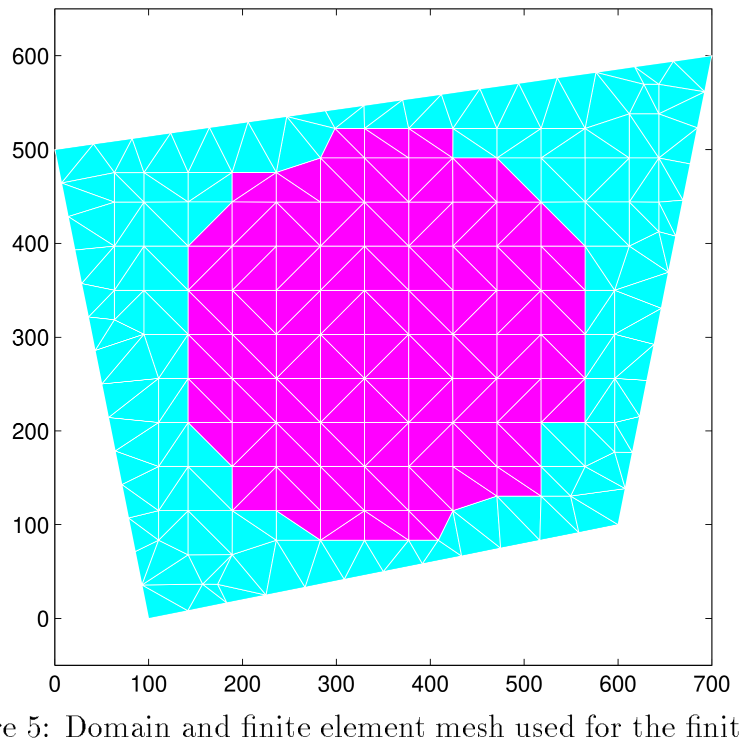


Figure 5: Domain and finite element mesh used for the finite element experiment. Conductivity in the dark triangles is 10^{12} and in the light triangles is 1.

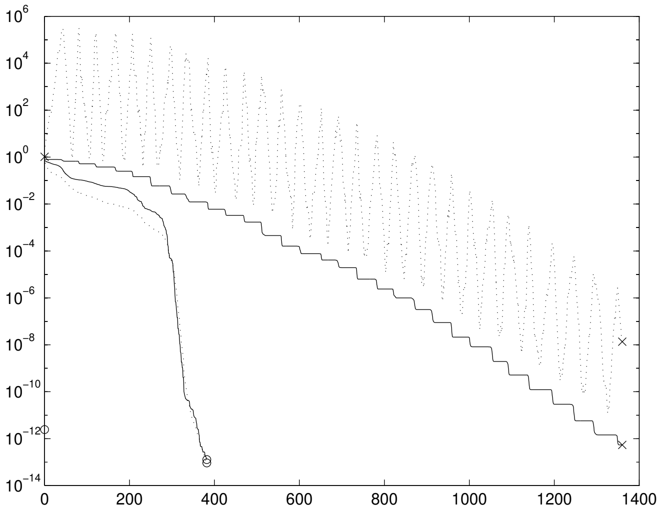


Figure 6: Convergence of CGNR and MINRES-L for the finite element test problem. The curves are labeled as in Figure 2.

Thus, $A^T D\mathbf{b} = \delta_2 A_2^T D_2 \mathbf{b}_2$ for this problem. Since δ_2 is $O(10^{-12})$, CGNR starts from a right-hand side that is already almost zero. Furthermore, this right-hand side is nearly orthogonal to the span of $A_1^T D_1 A_1$, which dominates the stiffness matrix $A^T DA$. Thus, CGNR has trouble making progress. The surprisingly accurate answer from CGNR in this example is not so useful in practice because there is no apparent way to detect that convergence is underway.

The final test is a three-layered problem based on the matrix A from ADLITTLE of the Netlib test set, a 138×56 matrix. Matrix D has as its first 28 diagonal entries 1, its next 28 diagonal entries 10^{-8} and its last 82 entries 10^{-16}. The right-hand side vector is the first 138 prime numbers. The convergence is depicted in Figure 7. As expected, the scaled error of MINRES-L decreased to $2 \cdot 10^{-10}$, while the scaled error of CGNR was 0.3. Note the excessive number of iterations required by MINRES-L. Again, this is apparently due to loss of orthogonality because the number of iterations was only 118 for GMRES-L to achieve a scaled error of $9.4 \cdot 10^{-13}$. In fact, for this test GMRES-L was more efficient than MINRES-L in terms of flop count.

In most cases we see that the MINRES-L algorithm performs essentially as expected, except for the two cases in which a loss of orthogonality causes many more iterations than expected. In every case, MINRES-L's running time is higher than CGNR's, but CGNR can produce bad solutions as measured by forward error.

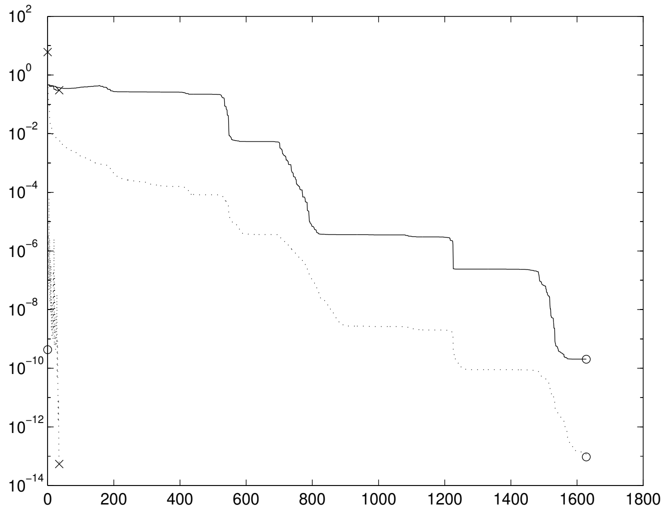


Figure 7: Convergence of CGNR and MINRES-L for ADLITTLE. The curves are labeled as in Figure 2. Note the excessive number of iterations for MINRES-L caused by a loss of orthogonality.

9 An Issue for Interior-Point Methods

In this section we describe an issue that arises when using the MINRES-L algorithm in an interior-point method for linear programming. Full consideration of this matter is postponed to future work.

It is well known that the system of equations for the Newton step in an interior-point method can be expressed as a weighted least-squares problem. To be precise, consider the linear programming problem

$$\begin{array}{rl} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A^T\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

whose dual is

$$\begin{array}{rl} \text{maximize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & A\mathbf{y} + \mathbf{s} = \mathbf{c}, \\ & \mathbf{s} \geq \mathbf{0} \end{array}$$

(which is standard form, except we have transposed A to be consistent with least-squares notation). A primal-dual method starting at a feasible interior point $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ for this problem computes an update $\Delta\mathbf{y}$ to $\mathbf{y}$ satisfying

$$A^T DA\Delta\mathbf{y} = A^T D(\mathbf{s} - \sigma\mu X^{-1}\mathbf{e}), \tag{40}$$

where $X = \mathrm{diag}(\mathbf{x})$, $S = \mathrm{diag}(\mathbf{s})$, $D = XS^{-1}$, σ is an algorithm-dependent parameter usually in $[0, 1]$, μ is the duality gap, and $\mathbf{e}$ is the vector of all 1's.

See Wright [26]. Since (40) has the form of a WLS problem, we can obtain $\Delta\mathbf{y}$ using the MINRES-L algorithm.

One way to compute $\Delta\mathbf{s}$ is via $\Delta\mathbf{s} := -A\Delta\mathbf{y}$. This method is not stable because $\Delta\mathbf{s}$ has very small entries in positions where $\mathbf{s}$ has very small entries; these small entries must be computed accurately with respect to the corresponding entry of $\mathbf{s}$. In contrast, the error in all components of $\Delta\mathbf{s}$ arising from the product $A\Delta\mathbf{y}$ is on the order of $\epsilon \cdot \|\mathbf{s}\|$ (where ϵ is machine-epsilon). A direct method for accurately computing all components of $\Delta\mathbf{s}$ was proposed by Hough [11], who obtains a bound of the form

$$|\Delta s_i - \widehat{\Delta s_i}|/s_i \leq f(A) \cdot \epsilon \tag{41}$$

for each i. We will consider methods for extending MINRES-L to accurate computation of $\Delta\mathbf{s}$ in future work. As noted by Hough, $\Delta\mathbf{x}$ is easily computed from $\Delta\mathbf{s}$ with a similar accuracy bound assuming $\Delta\mathbf{s}$ satisfies (41).

10 Conclusions

We have presented an iterative algorithm MINRES-L for solving weighted least squares. Theory and computational experiments indicate that the method is more accurate than CGNR when the weight matrix is highly ill-conditioned. This work raises a number of questions.

1. Is there an iterative method that does not require the layering assumption?

2. If layering is indeed required, can we get a more parsimonious layered linear system when $p \geq 3$? In particular, is there a $3n \times 3n$ system of equations with all the desired properties for the 3-layered case (instead of the $4n \times 4n$ system that we presented)?

3. What is the best way to handle loss of orthogonality in MINRES that was observed in Section 8?

4. Can this work be extended to stable computation of $\Delta\mathbf{x}$ and $\Delta\mathbf{s}$ in an interior-point method? (This question was raised in Section 9.)

5. What about preconditioning? In most of our computational tests, we ran both MINRES and CG for more than n iterations because our aim

was to compute the solution vector as accurately as possible. In practice, one hopes for convergence in much fewer than n iterations. What are techniques for preconditioning WLS problems? Note that the analysis of MINRES-L's accuracy in Section 5 and Section 7 presupposes that no preconditioner is used.

Acknowledgments

We had helpful discussions of this work with Anne Greenbaum and Mike Overton of NYU; Roland Freund, David Gay, and Margaret Wright of Bell Labs; Patty Hough of Sandia; Rich Lehoucq and Steve Wright of Argonne; Homer Walker of Utah State; and Zdeněk Strakoš of the Czech Academy of Sciences. We thank Patty Hough and Gail Pieper for carefully reading an earlier draft of this paper. In addition, we received the Netlib linear programming test cases in Matlab format from Patty Hough.

References

[1] A. Björck. *Numerical methods for least squares problems.* SIAM Press, Philadelphia, 1996.

[2] A. Björck, T. Elfving, and Z. Strakoš. Stability of conjugate gradient and Lanczos methods for linear least squares problems. Preprint, 1997.

[3] E. Bobrovnikova and S. Vavasis. Iterative methods for weighted least squares. Appears in *Proc. Copper Mountain Conference on Iterative Methods* distributed to conference participants, 1996.

[4] P. Brown and H. Walker. GMRES on (nearly) singular systems. Technical Report UCRL-JC-115882, Numerical Mathematics Group, Center for Computational Sciences and Engineering, Lawrence Livermore National Laboratory, 1994. To appear in *SIAM J. Matrix Anal. App.*

[5] J. Drkošová, A. Greenbaum, M. Rozložník, and Z. Strakoš. Numerical stability of GMRES. *BIT*, 25:309–330, 1995.

[6] A. L. Forsgren. On linear least-squares problems with diagonally dominant weight matrices. Report TRITA-MAT-1995-OS2, Optimization

and Systems Theory, Department of Mathematics, Royal Institute of Technology, S-100 44 Stockholm, Sweden, 1995.

[7] G. Golub and C. Van Loan. *Matrix Computations, 3rd edition.* Johns Hopkins University Press, Baltimore, 1996.

[8] C. Gonzaga and H. Lara. A note on properties of condition numbers. Preprint, 1996.

[9] A. Greenbaum. Estimating the attainable accuracy of recursively computed residual methods. Technical Report TR95-1515, Department of Computer Science, Cornell University, 1995. To appear in *SIAM J. Matrix Anal. App.*

[10] M. Hanke. *Conjugate gradient type methods for ill-posed problems.* Longman, Harlow, Essex, 1995.

[11] P. Hough. Stable computation of search directions for near-degenerate linear programming problems. Unpublished manuscript, 1997.

[12] P. Hough and S. Vavasis. Complete orthogonal decomposition for weighted least squares. Technical Report 94TR203, Advanced Computing Research Institute, Cornell Theory Center, 1994. To appear in *SIAM J. Matrix Anal. Appl.*

[13] C. Lawson and R. Hanson. *Solving Least Squares Problems.* Prentice Hall, Englewood Cliffs, New Jersey, 1974. Republished by SIAM Press, Philadelphia, 1995.

[14] C. Paige and M. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617–629, 1975.

[15] C. Paige and M. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8:43–71, 1982.

[16] C. C. Paige. Practical use of the symmetric Lanczos process with reorthogonalization. *BIT*, 10:183–195, 1970.

[17] B. Parlett and D. Scott. The Lanczos algorithm with selective reorthogonalization. *Math. Comp.*, 33:217–238, 1979.

[18] Y. Saad. *Iterative methods for sparse linear systems.* PWS Publishing Company, Boston, 1996.

[19] Y. Saad and M. H. Schultz. GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[20] G. W. Stewart. On scaled projections and pseudoinverses. *Linear Algebra and Its Applications*, 112:189–193, 1989.

[21] G. Strang. A framework for equilibrium equations. *SIAM Review*, 30:283–297, 1988.

[22] M. J. Todd. A Dantzig-Wolfe-like variant of Karmarkar's interior-point linear programming algorithm. *Operations Research*, 38:1006–1018, 1990.

[23] S. A. Vavasis. Stable numerical algorithms for equilibrium systems. *SIAM J. Matrix Anal. Appl.*, 15:1108–1131, 1994.

[24] S. A. Vavasis. Stable finite elements for problems with wild coefficients. *SIAM J. Numer. Anal.*, 33:890–916, 1996.

[25] S. A. Vavasis and Y. Ye. A primal-dual interior point method whose running time depends only on the constraint matrix. *Mathematical Programming*, 74:79–120, 1996.

[26] S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM Press, Philadelphia, 1997.