On the Method Of Weighting for Equality Constrained Least Squares Problems

Charles Van Loan

TR 84-604
March 1984
(Revised August 1984)

Department of Computer Science
Cornell University
Ithaca, New York 14853

*The research for this paper was supported under the auspices of ONR contract N00014-83-K-0640.

N -10035

ON THE METHOD OF WEIGHTING FOR EQUALITY-CONSTRAINED LEAST-SQUARES PROBLEMS

Charles Van Loan
Department of Computer Science
Cornell University
Ithaca, New York 14853

Abstract

The generalized singular value decomposition is used to analyze the problem of minimizing $\| Ax - b \|_2$ subject to the constraint $Bx = d$. A by-product of the analysis is a new iterative procedure that can be used to improve an approximate solution obtained via the method of weights. All that is required to implement the procedure is a single QR factorization. These developments turn out to be of interest when A and B are sparse and for the case when systolic architectures are used to carry out the computations.

The research for this paper was supported under the auspices of ONR contract N00014-83-K-0640 .

1. Introduction

The problem we consider is how to find a vector $x \in R^n$ that solves the "LSE" problem

$$(1.1) \qquad \min_{Bx = d} \|Ax - b\|_2$$

where $A \in R^{m\times n}$ $(m \geq n)$, $b \in R^m$, $B \in R^{p\times n}$ $(p \leq n)$ and $d \in R^p$. We will assume that

$$(1.2) \qquad \operatorname{rank}(B) = p$$

and that the null spaces of A and B intersect only trivially:

$$(1.2) \qquad N(A) \cap N(B) = \{0\} \iff \operatorname{rank}\left[\begin{pmatrix} A \\ B \end{pmatrix}\right] = n$$

This condition ensures that (1.1) has a unique solution which we designate by x_{LSE}.

Important settings where the LSE problem arises include constrained surface fitting, constrained optimization, geodetic least-squares adjustment, and beam-forming.

Several methods for solving the LSE problem are discussed in Lawson and Hanson [17, Ch.20-22]. In one approach QR factorizations are used to compute the projections of x_{LSE} onto $N(B)$ and its orthogonal complement $N(B)^{\perp}$:

Algorithm 1.1

(a) Compute an orthogonal Q such that

$$Q^T B^T = \begin{bmatrix} R_B \\ 0 \end{bmatrix} \begin{array}{l} p \\ n-p \end{array}$$

is upper triangular.

(b) Solve the pxp system $R_B^T y_1 = d$ and set $x_1 = Q_1 y_1$ where

$$Q = [\underset{p}{Q_1}, \underset{n-p}{Q_2}]$$

(c) Compute an orthogonal U such that

$$U^T(AQ_2) = \begin{bmatrix} R_A \\ 0 \end{bmatrix} \begin{array}{l} n-p \\ m-n+p \end{array}$$

is upper triangular.

(d) Solve $R_A y_2 = U_1^T(b - Ax_1)$ and set $x_2 = Q_2 y_2$ where

$$U = [\underset{n-p}{U_1}, \underset{m-n+p}{U_2}]$$

(e) $x = x_1 + x_2$

This algorithm is particularly simple to implement using LINPACK subroutines [7] . In MATLAB [18] it requires five lines of code.

Unfortunately, Algorithm 1.1 is not viable when A is large and sparse, a situation that occurs with some frequency. The trouble lies with the fill-in that can be expected during the formation of the product AQ_2. If B is sparse then recent sparse null space techniques could be used to generate a sparse Q_2;

see [15,20] . However, the sparsity of AQ_2 is unpredictable with such a process.

Another shortcoming of Algorithm 1.1 surfaces when one seeks to implement it on systolic arrays tailored to perform fast QR factorizations. In an important beam forming problem, one must solve the LSE problem for many different B matrices. (A is fixed.) Ideally, one would like to "pipeline" the solution process in Algorithm 1.1 to achieve maximum concurrency. Unfortunately, this turns out to be impossible since the QR factorization of each AQ_2 must be calculated from scratch.

This prompts us to solve the LSE problem by the well known method of weights. In this approach a positive weight μ is chosen and the <u>un</u>constrained least squares problem

$$\min \left\| \begin{pmatrix} \mu B \\ A \end{pmatrix} x - \begin{pmatrix} \mu d \\ b \end{pmatrix} \right\|_2 \tag{1.4}$$

is solved. The assumption (1.3) ensures that this is a full rank least-squares problem thereby having a unique solution which we designate by $x(\mu)$. It is well known that $\lim_{\mu\to\infty} x(\mu) = x_{LSE}$. Perhaps the easiest way to see this is to observe that if

$$\begin{bmatrix} 0 & 0 & B \\ 0 & I_m & A \\ B^T & A^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ r \\ x \end{bmatrix} = \begin{bmatrix} d \\ b \\ 0 \end{bmatrix} \tag{1.5}$$

then $x = x_{LSE}$ while

$$\begin{bmatrix} \mu^{-2} I_p & 0 & B \\ 0 & I_m & A \\ B^T & A^T & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ z \end{bmatrix} = \begin{bmatrix} d \\ b \\ 0 \end{bmatrix} \tag{1.6}$$

implies $z = x(\mu)$. Equation (1.5) arises by applying Lagrange multipliers to (1.1) while (1.6) can be derived by considering the normal equations associated with (1.4). Clearly, as μ gets large these two systems approach one another. Thus, $x(\mu) = z \rightarrow x = x_{LSE}$ as the weight μ tends to infinity.

The method of weights is attractive for its simplicity. We merely compute the QR factorization

$$(1.7) \qquad \begin{pmatrix} \mu B \\ A \end{pmatrix} = Q_\mu \begin{pmatrix} R_\mu \\ 0 \end{pmatrix}$$

and solve the nonsingular nxn upper triangular system

$$R_\mu x(\mu) = Q_1(\mu)^T \begin{pmatrix} \mu d \\ b \end{pmatrix}$$

where

$$Q_\mu = [\ \underset{n}{Q_1(\mu)}\ ,\ \underset{m+p-n}{Q_2(\mu)}\ ]$$

is orthogonal. Standard LINPACK [7] routines can be used for this purpose. If A and B are both sparse, then the George-Heath sparse least-squares algorithm can be invoked; see [10]. In the systolic array setting, the standard QR arrays proposed in [9] and [16] are applicable.

The accuracy of $x(\mu)$ is of obvious concern with the weighting approach. An exact expression for the error using the generalized singular value decomposition is given in §2. Unfortunately, a large weight may be necessary to render an acceptable $x(\mu)$ and this can cause severe numerical problems as we illustrate in §3. In §4 we present an iterative procedure that can be used to improve a "small weight" $x(\mu)$. Some numerical results and implementation details are discussed in §5.

2. Theoretical Analysis of x_{LSE} and $x(\mu)$

The best way to analyze the method of weighting is through the generalized singular value decomposition. This decomposition is discussed in [19] and [24]. See [23] and [26] for computational issues. We establish a specially normalized version of the decomposition that simplifies our analysis of the LSE problem and the method of weighting for solving it.

Theorem 2.1

If $A \in R^{m\times n}$ $(m \geq n)$ and $B \in R^{p\times n}$ $(\mathrm{rank}(B) = p)$ satisfy (1.3), then there exist

$$U = [u_1,\ldots,u_m] \in R^{m\times m} \qquad \text{(orthogonal)}$$
$$V = [v_1,\ldots,v_p] \in R^{p\times p} \qquad \text{(orthogonal)}$$
$$X = [x_1,\ldots,x_n] \in R^{n\times n} \qquad \text{(nonsingular)}$$

such that

$$U^T A X = D_A = \mathrm{diag}(\alpha_1,\ldots,\alpha_n) \qquad (2.1)$$

$$V^T B X = D_B = \mathrm{diag}(\beta_1,\ldots,\beta_p) \quad , \qquad (2.2)$$

If $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the singular values of the matrix $\begin{pmatrix} A \\ B \end{pmatrix}$ then without loss of generality we may assume that

$$\| X \|_2 = 1 \qquad (2.3)$$

$$\| X^{-1} \|_2 = \sigma_1 / \sigma_n \qquad (2.4)$$

$$(2.5)\qquad 0 = \alpha_1 = \dots = \alpha_q < \alpha_{q+1} \leqq \dots \leqq \alpha_p \leqq \alpha_{p+1} = \dots = \alpha_n = \sigma_n$$

$$(2.6)\qquad \beta_1 \geqq \dots \geqq \beta_p \geqq 0$$

$$(2.7)\qquad \alpha_i^2 + \beta_i^2 = \sigma_n^2 \qquad (i = 1,\dots,p)$$

Proof

Let

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \operatorname{diag}(\sigma_1,\dots,\sigma_n) Z^T$$

be the singular value decomposition (SVD) of $\begin{pmatrix} A \\ B \end{pmatrix}$ with $Q_1^T Q_1 + Q_2^T Q_2 = I_n$, $\sigma_1 \geqq \cdots \geqq \sigma_n \geqq 0$ and $Z^T Z = I_n$. Let

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} C \\ S \end{pmatrix} W^T$$

be the CS decomposition of Q_1 and Q_2 where $U \in R^{m\times m}$, $V \in R^{p\times p}$, and $W \in R^{n\times n}$ are each orthogonal and where

$$C = \operatorname{diag}(c_1,\dots,c_n) \in R^{m\times n} \qquad c_i \geqq 0$$

$$S = \operatorname{diag}(s_1,\dots,s_p) \in R^{p\times n} \qquad s_1 \geqq s_2 \geqq \cdots \geqq s_p \geqq 0$$

satisfy $C^T C + S^T S = I_n$. (The SVD and the CS decomposition are discussed in [11].) Our theorem follows by setting $D_A = \sigma_n C$, $D_B = \sigma_n S$, and

$$X = \sigma_n Z \operatorname{diag}\left(\frac{1}{\sigma_1},\dots,\frac{1}{\sigma_n}\right) W .$$

Note that (1.3) guarantees that σ_n is positive. □

Note that (1.2) implies $\beta_p > 0$. We define the generalized singular values to be the quotients

(2.8) $$\mu_i = \alpha_i / \beta_i \qquad (i = 1,\ldots,p) \quad .$$

From (2.5) and (2.6) it follows that

(2.9) $$0 = \mu_1 = \ldots = \mu_q < \mu_{q+1} \leqq \ldots\ldots \leqq \mu_p \quad .$$

Moreover, we have

(2.10) $$Ax_i = \alpha_i u_i \qquad (i = 1,\ldots,n)$$

(2.11) $$Bx_i = \beta_i v_i \qquad (i = 1,\ldots,p) \quad .$$

Thus, $N(A) = \text{span}\{x_1,\ldots,x_q\}$.

Theorem 2.1 can be used to effectively diagonalize the LSE problem. In particular, by setting

$$\begin{aligned} \tilde{b} &= U^T b = (u_1^T b \ ,\ldots,\ u_m^T b\ )^T \\ \tilde{d} &= V^T d = (v_1^T d \ ,\ldots,\ v_p^T d\ )^T \\ x &= Xy = X(y_1,\ldots,y_n)^T \end{aligned}$$

we find that (1.1) transforms to

(LSE') $$\min_{D_B y = \tilde{d}} \| D_A y - \tilde{b} \|_2$$

It is not hard to show that

$$y_{LSE} = \left(\frac{v_1^T d}{\beta_1}, \ldots, \frac{v_p^T d}{\beta_p}, \frac{u_{p+1}^T b}{\alpha_{p+1}}, \ldots, \frac{u_n^T b}{\alpha_n} \right)^T$$

solves (LSE') . In light of the normalization (2.5) we have

$$(2.12) \qquad x_{LSE} = Xy_{LSE} = \sum_{i=1}^{p} \frac{v_i^T d}{\beta_i} x_i + \frac{1}{\sigma_n} \sum_{i=p+1}^{n} (u_i^T b) x_i .$$

Note that the minimum 2-norm solution to the unconstrained LS problem $\min \| Ax - b \|_2$ is given by

$$(2.13) \qquad x_{LS} = \sum_{i=q+1}^{p} \frac{u_i^T b}{\alpha_i} x_i + \frac{1}{\sigma_n} \sum_{i=p+1}^{n} (u_i^T b) x_i$$

Using (2.12), (2.13) and recalling (2.10) we have

$$(2.14) \qquad r_{LS} = b - Ax_{LS} = \sum_{i=1}^{q} (u_i^T b) u_i + \sum_{i=n+1}^{m} (u_i^T b) u_i$$

$$(2.15) \qquad r_{LSE} = b - Ax_{LSE} = r_{LS} + \sum_{i=q+1}^{p} \rho_i u_i$$

where

$$(2.16) \qquad \rho_i = u_i^T b - \mu_i v_i^T d \qquad (i = 1, \ldots, p) .$$

Note that

$$(2.17) \qquad \Delta^2 = \sum_{i=1}^{p} \rho_i^2$$

measures how much the minimum residual increases as a result of the constraint $Bx = d$.

The generalized singular value decomposition can also be used to obtain a useful expression for $x(\mu)$. Note that $x(\mu)$ satisfies the normal equation

$$(2.18)\qquad (A^TA + \mu^2B^TB)x(\mu) = A^Tb + \mu^2B^Td .$$

Analogous to how we obtained (LSE'), this equation transforms to

$$(D_A^TD_A + \mu^2D_B^TD_B)y(\mu) = D_A^T\tilde{b} + \mu^2D_B^T\tilde{d}$$

where $x(\mu) = Xy(\mu)$. is easy to deduce from this diagonal system that

$$(2.19)\qquad x(\mu) = \sum_{i=1}^{p} \frac{\alpha_iu_i^Tb + \mu^2\beta_iv_i^Td}{\alpha_i^2 + \mu^2\beta_i^2}\, x_i + \frac{1}{\sigma_n}\sum_{i=p+1}^{n}(u_i^Tb)x_i$$

Subtracting (2.12) from this expression and doing a little algebra we obtain the following expansion for the error:

$$(2.20)\qquad e(\mu) \equiv x(\mu) - x_{LSE} = \sum_{i=q+1}^{p} \frac{\mu_i^2}{\mu_i^2 + \mu^2}\cdot\frac{\rho_i}{\alpha_i}\, x_i$$

Note that the error is confined to the subspace span $\{x_{q+1},\ldots,x_p\}$ and that it obviously tends to zero as the weight tends to infinity.

As part of a general analysis that we shall perform in §4, we show that

$$\| x(\mu) - x_{LSE} \|_2 \le \frac{\Delta}{2\mu}\, \frac{1}{\beta_p}$$

This suggests that if β_p is small (or equivalently μ_p is large) then a large weight may be necessary. As we discuss in the next section, this can cause numerical difficulties. However, it should be noted from (2.12) that x_{LSE} will be sensitive to perturbation in this case and so we can expect difficulties no matter what method we use to compute x_{LSE} .

3. Difficulties Associated with a Large Weight

It is well known that care must be exercised when Householder matrices are used to compute the QR factorization of a matrix whose rows vary greatly in norm, e.g., the matrix in (1.4). Powell and Reid [21] examined this problem in conjunction with the Businger-Golub algorithm described in [6] and advise incorporation of row interchanges, much as in Gaussian elimination. Specifically, they recommend that the k-th column be searched and its largest entry pivoted to the (k,k) position before the k-th Householder matrix is applied.

Note that near-domination of the pivot elements will result if we apply the Businger-Golub algorithm to $\begin{pmatrix} \mu B \\ A \end{pmatrix}$ but <u>not</u> if we apply it to the matrix in

$$\min \left\| \begin{pmatrix} A \\ \mu B \end{pmatrix} x - \begin{pmatrix} b \\ \mu d \end{pmatrix} \right\|_2 \tag{3.1}$$

which is mathematically equivalent to (1.4). To appreciate the difference between the B-over-A and the A-over-B approaches to the LSE problem, consider the example

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} , \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} , \quad B = (1 \ \ -1) , \quad d = (2) .$$

This problem is well-conditioned and has exact solution $\frac{1}{29}(39 , -19)^T$. In Table 1 we record the magnitude of $\| x(\mu) - x_{LSE} \|_2$ for both approaches as a function of μ.

μ	10^{1}	10^{3}	10^{5}	10^{7}	10^{9}	10^{11}	10^{13}	10^{15}	10^{17}
B-over-A Error	10^{-3}	10^{-7}	10^{-11}	10^{-15}	10^{-15}	10^{-17}	10^{-17}	10^{-17}	10^{-17}
A-over-B Error	10^{-3}	10^{-7}	10^{-11}	10^{-11}	10^{-10}	10^{-7}	10^{-6}	10^{-4}	10^{-2}

Table 1.

Computations were performed using VAX double-precision arithmetic in the MATLAB environment. Let machep be the machine precision, in our case 10^{-17}. The divergence of performance between the two approaches in the vicinity of $\mu = (\text{machep})^{-\frac{1}{2}}$ is fairly typical. We observed this to be the case even in ill-conditioned examples. Although the B-over-A approach is always preferable from the numerical standpoint, it is sometimes difficult to set up (1.4) with the constraint equations on top. For example, the minimization of fill-in may force us to choose some alternative row-ordering. However, some interesting thoughts about how to preserve both sparsity and stability are given in [4].

Another inconvenience thrust upon us if we must employ a large weight is the need for column interchanges when computing the QR factorization of $\begin{pmatrix} \mu B \\ A \end{pmatrix}$. An example suggested by our colleague Per-Åke Wedin makes this clear. Suppose

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \qquad d = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$$

This example is well-conditioned and $x_{LSE} = \frac{1}{8}(46,-2,12)^T$.

In Table 2 we tabulate the error in $x(\mu)$ for various values of μ . The cases when column pivoting is used and when it is not are recorded.

μ	10^1	10^3	10^5	10^7	10^9	10^{11}	10^{13}	10^{15}
With Column Pivoting	10^{-2}	10^{-7}	10^{-10}	10^{-14}	10^{-16}	10^{-16}	10^{-16}	10^{-16}
No Column Pivoting	10^{-2}	10^{-7}	10^{-11}	10^{-11}	10^{-9}	10^{-7}	10^{-5}	10^{-3}

Table 2.

Trouble arises without column pivoting because the first two columns of the matrix $\begin{pmatrix} \mu B \\ A \end{pmatrix}$ are nearly dependent for large μ . Consequently, the (2,2) element of the upper triangular matrix R_μ in (1.7) approaches zero. These difficulties are circumvented when column interchanges are performed.

From the examples in this section we conclude that both row and column ordering can be critical when the weighting method is used to solve the LSE problem. This limits its usefulness for sparse problems and complicates its implementation on systolic architectures because the QR arrays that have thus far been proposed do not have column pivoting capability. See Gentleman and Kung [8] and Heller and Ipsen [13] for example.

4. An Improvement Scheme for $x(\mu)$

We now show how the solution $x(\mu)$ can be improved using only the QR factorization of $\begin{pmatrix}\mu B\\ A\end{pmatrix}$. This will give us the opportunity to "get by" with reasonably small weights thereby circumventing the problems alluded to in the previous section. The key idea behind our procedure is to correct solutions to (1.6) by exploiting (1.5).

Algorithm 4.1

Choose μ and compute the solution $x(\mu)$ to (1.4).

Set

$$x^{(1)} = x(\mu)$$
$$r^{(1)} = b - Ax(\mu)$$
$$\lambda^{(1)} = \mu^2(d - Bx(\mu))$$

For $k = 1,2,...$

Compute the residual associated with (1.5):

$$\begin{bmatrix}\delta_1^{(k)}\\ \delta_2^{(k)}\\ \delta_3^{(k)}\end{bmatrix} = \begin{bmatrix}d\\ b\\ 0\end{bmatrix} - \begin{bmatrix}0 & 0 & B\\ 0 & I_m & A\\ B^T & A^T & 0\end{bmatrix}\begin{bmatrix}\lambda^{(k)}\\ r^{(k)}\\ x^{(k)}\end{bmatrix}$$

Solve the (1.6) system

$$\begin{bmatrix}\mu^{-2}I_p & 0 & B\\ 0 & I_m & A\\ B^T & A^T & 0\end{bmatrix}\begin{bmatrix}\Delta\lambda^{(k)}\\ \Delta r^{(k)}\\ \Delta x^{(k)}\end{bmatrix} = \begin{bmatrix}\delta_1^{(k)}\\ \delta_2^{(k)}\\ \delta_3^{(k)}\end{bmatrix}$$

Set

$$\lambda^{(k+1)} = \lambda^{(k)} + \Delta\lambda^{(k)}$$
$$r^{(k+1)} = r^{(k)} + \Delta r^{(k)}$$
$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$$

It is not at all obvious that the $x^{(k)}$ converge to x_{LSE}. Nor is it obvious how one would actually solve for the corrections $\Delta\lambda^{(k)}$, $\Delta r^{(k)}$, and $\Delta x^{(k)}$. These issues will be addressed once we make some simplifications based on the following theorem:

Theorem 4.1

For all k in Algorithm 4.1, $\delta_2^{(k)} = 0$ and $\delta_3^{(k)} = 0$.

Proof

Since

$$\begin{bmatrix} d \\ b \\ 0 \end{bmatrix} = \begin{bmatrix} \mu^{-2} I_p & 0 & B \\ 0 & I_m & A \\ B^T & A^T & 0 \end{bmatrix} \begin{bmatrix} \lambda^{(1)} \\ r^{(1)} \\ x^{(1)} \end{bmatrix}$$

we have $\delta_2^{(1)} = 0$ and $\delta_3^{(1)} = 0$. Now suppose for some $k \geq 1$ that $\delta_2^{(k)} = 0$ and $\delta_3^{(k)} = 0$. Since

$$\begin{bmatrix} \delta_1^{(k+1)} \\ \delta_2^{(k+1)} \\ \delta_3^{(k+1)} \end{bmatrix} = \begin{bmatrix} \delta_1^{(k)} \\ \delta_2^{(k)} \\ \delta_3^{(k)} \end{bmatrix} - \begin{bmatrix} B\Delta x^{(k)} \\ \Delta r^{(k)} + A\Delta x^{(k)} \\ B^T\Delta\lambda^{(k)} + A^T\Delta r^{(k)} \end{bmatrix}$$

the theorem will follow if we can show that

$$\Delta r^{(k)} + A\,\Delta x^{(k)} = 0$$

and

$$B^T\,\Delta\lambda^{(k)} + A^T\,\Delta r^{(k)} = 0 \quad .$$

But these two results hold because by the induction hypothesis we have

$$\begin{bmatrix} \mu^{-2} I_p & 0 & B \\ 0 & I_m & A \\ B^T & A^T & 0 \end{bmatrix} \begin{bmatrix} \Delta\lambda^{(k)} \\ \Delta r^{(k)} \\ \Delta x^{(k)} \end{bmatrix} = \begin{bmatrix} \delta_1^{(k)} \\ 0 \\ 0 \end{bmatrix} \qquad \square$$

Observing from (1.6) that $\Delta x^{(k)}$ solves the problem

$$\text{(4,1)} \qquad \min_z \left\| \begin{pmatrix} \mu B \\ A \end{pmatrix} z - \begin{pmatrix} \mu\delta_1^{(k)} \\ 0 \end{pmatrix} \right\|_2$$

we see that Algorithm 4.1 transforms to

<u>Algorithm</u> <u>4.2</u>

Choose μ and compute the solution $\dot{x}(\mu)$ to (1.4).

Set $x^{(1)} = x(\mu)$

For $k = 1,2,\ldots$

$$\delta_1^{(k)} = d - Bx^{(k)}$$

$$\text{Solve} \left\| \begin{pmatrix} \mu B \\ A \end{pmatrix} \Delta x^{(k)} - \begin{pmatrix} \mu\delta_1^{(k)} \\ 0 \end{pmatrix} \right\|_2 = \min$$

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$$

It is important to stress that only a single QR factorization is necessary to carry out the iteration. In particular, once we have the QR factorization

$$\begin{pmatrix} \mu B \\ A \end{pmatrix} = [\ Q_1(\mu)\ ,\ Q_2(\mu)\ ] \begin{pmatrix} R_\mu \\ 0 \end{pmatrix}$$

then

$$R_\mu\ x(\mu) = Q_1(\mu)^T \begin{pmatrix} \mu d \\ b \end{pmatrix}$$

and

$$R_\mu \Delta x^{(k)} = Q_1(\mu)^T \begin{pmatrix} \mu \delta_1^{(k)} \\ 0 \end{pmatrix} .$$

We're now set to analyze the convergence properties of Algorithm 4.2 . We first establish an explicit expression for the error in $x^{(k)}$.

Theorem 4.1

The vectors $x^{(k)}$ in Algorithm 4.2 satisfy

$$x^{(k)} = x_{LSE} + e(\mu,k)$$

where

$$e(\mu,k) = \sum_{i=q+1}^{p} \frac{\rho_i}{\alpha_i} \left(\frac{\mu_i^2}{\mu^2 + \mu_i^2} \right)^k x_i \tag{4.2}$$

Proof

From (2.20) it is clear that the theorem holds for $k = 1$. We use induction to establish the result for general k. Since

$$e(\mu,k+1) = x^{(k+1)} - x_{LSE} = e(\mu,k) + \Delta x^{(k)}$$

and from the normal equations for (4.1) we have

$$\begin{aligned}(A^TA + \mu^2B^TB)\Delta x^{(k)} &= \mu^2B^T\,\delta_1^{(k)} = \mu^2B^T(d - Bx^{(k)}) = \\ &= \mu^2\,B^T\,[\,(d - Bx_{LSE}) + B(x_{LSE} - x^{(k)})\,] \\ &= -\,\mu^2\,B^TB\,e(\mu,k)\end{aligned}$$

it follows that

$$e(\mu,k+1) = [\,I - \mu^2(A^TA + \mu^2\,B^TB)^{-1}\,B^TB\,]\,e(\mu,k) \quad .$$

Using the generalized singular value decomposition of A and B (see §2) we have

$$[\,I - \mu^2(A^TA + \mu^2\,B^TB)^{-1}\,B^TB\,]\;x_i = \frac{\mu_i^2}{\mu^2 + \mu_i^2}\,x_i$$

for $i = q+1,\ldots,p$. Assuming that the theorem holds for k it follows that it holds for $k+1$ since

$$e(\mu,k+1) = [\,I - \mu^2(A^TA + \mu^2\,B^TB)^{-1}\,B^TB\,]\sum_{i=q+1}^{p}\frac{\rho_i}{\alpha_i}\left(\frac{\mu_i^2}{\mu^2 + \mu_i^2}\right)^k x_i \quad . \;\Box$$

With this result we can establish bounds on the error in $x^{(k)}$ as well as on the associated residuals $b - Ax^{(k)}$ and $d - Bx^{(k)}$.

Corollary 4.3

Let $m_\mu = \mu_p^2 / (\mu_p^2 + \mu^2)$ where μ_p is the largest generalized singular value of the pair (A,B) and let Δ be defined by (2.17). The vectors $x^{(k)}$ generated by Algorithm 4.2 satisfy

$$(4.3) \qquad \| x^{(k)} - x_{LSE} \|_2 \leq \frac{\Delta}{2\mu} \frac{1}{\beta_p} m_\mu^{k-1}$$

$$(4.4) \qquad \| d - Bx^{(k)} \|_2 \leq \frac{\Delta}{2\mu} m_\mu^{k-1}$$

$$(4.5) \qquad 0 \leq \| b - Ax_{LSE} \|_2 - \| b - Ax^{(k)} \|_2 \leq \frac{\Delta}{2\mu} \mu_p m_\mu^{k-1}$$

Proof

From (4.2) it follows that

$$(4.6) \qquad e(\mu,k) = \sum_{i=q+1}^{p} \frac{\rho_i}{\beta_i} \left(\frac{\mu_i}{\mu_i^2 + \mu^2}\right) \left(\frac{\mu_i^2}{\mu_i^2 + \mu^2}\right)^{k-1} x_i$$

Inequality (4.3) now follows by taking norms in (4.6), invoking the definition of m_μ, and using the facts

$$\| [x_{q+1}, \ldots, x_p] \| \leq 1$$

$$\frac{1}{\beta_i} \leq \frac{1}{\beta_p} \qquad i = 1,\ldots,p$$

and

$$\frac{\mu_i}{\mu_i^2 + \mu^2} \leq \frac{1}{2\mu} .$$

Inequality (4.5) likewise follows by taking norms in the expression

$$d - Bx^{(k)} = - B\,e(\mu,k) = - \sum_{i=q+1}^{p} \rho_i \frac{\mu_i}{\mu_i^2 + \mu^2} \left(\frac{\mu_i^2}{\mu_i^2 + \mu^2} \right)^{k-1} v_i$$

and remembering that the v_i are mutually orthogonal. Finally, note that

$$b - Ax_{LSE} = b - Ax^{(k)} + Ae(\mu,k) = b - Ax^{(k)} + \sum_{i=q+1}^{p} \rho_i \left(\frac{\mu_i^2}{\mu_i^2 + \mu^2} \right)^{k} u_i .$$

The upper bound in (4.5) is readily obtained by taking norms. The lower bound follows upon comparison of

$$b - Ax_{LSE} = \sum_{i=1}^{q} (u_i^T b) u_i + \sum_{i=n+1}^{m} (u_i^T b) u_i + \sum_{i=q+1}^{p} \rho_i u_i$$

which can be derived from (2.14) and (2.15), and the expression

$$b - Ax^{(k)} = b - Ax_{LSE} - \sum_{i=q+1}^{p} \rho_i \left(\frac{\mu_i^2}{\mu_i^2 + \mu^2} \right)^{k} u_i \; . \; \Box$$

Before passing on to the next section we address a concern of the referee about the behavior of Algorithm 4.2 when the constraint equation $Bx = d$ is incompatable. Suppose $\text{rank}(B) = t \leqq p$ but that we still have $N(A) \cap N(B) = \{0\}$. This implies that in the GSVD of A and B everything is the same as before except that

$$(2.2') \qquad V^T B X = \text{diag}(\beta_1, \ldots, \beta_t, \underbrace{0, \ldots, 0}_{p-t}) \in R^{p\times n} .$$

The vector

$$(2.12') \qquad x_{LSE} = \sum_{i=1}^{t} \frac{v_i^T d}{\beta_i} x_i + \frac{1}{\sigma_n} \sum_{i=t+1}^{n} (u_i^T b) x_i$$

is the unique minimum of $\|Ax - b\|_2$ subject to the constraint that $\|Bx - d\|_2$ is minimum. By repeating the above analysis, which amounts to just replacing p with t, it is easy to confirm that $x^{(k)}$ converges to x_{LSE} defined by (2.12').

5. Numerical Results and Implementation Details

The preceding analysis shows that *in principle* Algorithm 4.2 converges for any nonzero μ. However, the size of μ is of great practical importance. To illustrate this point, we applied Algorithm 4.2 to a problem in which μ_p = 5000. The relative error in $x^{(k)}$ for various values of μ is tabulated in Table 3.

μ	k = 1	k = 2	k = 3	k = 4	k = 5
10^3	10^0	10^0	10^0	10^0	10^0
10^4	10^{-1}	10^{-2}	10^{-3}	10^{-3}	10^{-4}
10^5	10^{-3}	10^{-6}	10^{-8}	10^{-11}	10^{-14}
10^8	10^{-9}	10^{-14}	10^{-14}	10^{-14}	10^{-14}

Table 3

Note that the iteration cannot substantially improve the accuracy of $x(\mu)$ unless $m_\mu = \mu_p^2/(\mu_p^2 + \mu^2)$ is somewhat less than unity.

As a rule of thumb, we suggest implementing Algorithm 4.2 with the weight μ set to $(\text{machep})^{-\frac{1}{2}}$. Larger values may be successful but will depend upon the row and column ordering imposed by the QR factorization scheme that is invoked as we discussed in §3,

Finally, we mention that we cannot expect greater accuracy in the computed LSE solution than the condition of the underlying problem warrents. A detailed sensitivity analysis of x_{LSE} using the theory of weighted pseudo-inverses is given in [8]. The results in this reference confirm what (2.12) suggests: x_{LSE} is sensitive to perturbation if β_p and/or σ_n is small. Note that the error bounds for $x^{(k)}$ that we developed in §4 get worse as β_p gets small. This prompts us to conjecture that the improvement iteration converges slowly only on ill-conditioned LSE problems.

The conditioning issue is important from the standpoint of developing an intelligent termination criteria for Algorithm 4.2. Let δ be a predetermined tolerance. We have found that the stopping criteria

$$\| d - Bx^{(k)} \|_2 \leq \delta \| B \|_\infty \| x^{(k)} \|_2 \tag{5.1}$$

works quite well in practice. Proceeding heuristically, (5.1) coupled with (4.4) implies that

$$\frac{\Delta}{2\mu} m_\mu^{k-1} \approx \delta \| B \|_\infty \| x^{(k)} \|_2$$

and so

$$\| x^{(k)} - x_{LSE} \|_2 \leq \delta \frac{1}{\beta_p} \| B \|_\infty \| x^{(k)} \|_2$$

Thus, ill-conditioning evidenced by a small β_p is taken into account by our method of termination.

We hasten to add that ill-conditioning in the LSE problem due to a small σ_n surfaces when the corrections $\Delta x^{(k)}$ are calculated. If σ_n is small then then the matrix R_μ in (1.7) will tend to be ill-conditioned thereby contaminating $\Delta x^{(k)}$ with errors of order machep/ σ_n^2 . (Recall that in nonzero residual problems, the square of the condition is involved in the error bound; see Golub and Wilkinson [13] .) Hence, the accuracy of the computed $x^{(k)}$ depends upon both of the factors $1/\beta_p$ and $1/\sigma_n^2$.

Note that in a problem with $0 \cong \beta_p << \beta_{p-1}$ the correction

$$x^{(k)} = - \sum_{i=q+1}^{p} \frac{\rho_i}{\alpha_i} \frac{\mu^2}{\mu_i + \mu^2} \left(\frac{\mu_i^2}{\mu_i^2 + \mu^2} \right)^{k-1} x_i$$

is increasingly in the direction of x_p. Thus we obtain the heuristic

$$c_k^2 = \frac{\| \Delta x^{(k)} \|_2}{\| \Delta x^{(k-1)} \|_2} \cong \frac{\mu_p^2}{\mu_p^2 + \mu^2}$$

whereupon we obtain the estimate

$$\mu_p \cong \frac{|c_k| \mu}{\sqrt{1 - c_k^2}} .$$

As an illustration of the effect of terminating on the relative size of $\|d - Bx\|_2$ we applied the method to the problem

$$A = \begin{bmatrix} 0.2498 & 0.8873 & 0.7710 & 0.9195 \\ 0.8233 & 0.6996 & 0.2996 & 0.6763 \\ 0.0545 & 0.8812 & 0.6295 & 0.3206 \\ 0.3511 & 0.0937 & 0.2540 & 0.9563 \\ 0.6485 & 0.6165 & 0.1797 & 0.2535 \\ 0.6564 & 0.6907 & 0.2486 & 0.3397 \end{bmatrix} \qquad b = \begin{bmatrix} 0.4052 \\ 0.9185 \\ 0.0437 \\ 0.4819 \\ 0.2640 \\ 0.4148 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0044 & 0.0112 & 0.0086 & 0.0096 \\ 0.2308 & 0.5847 & 0.4503 & 0.5022 \end{bmatrix} \qquad d = \begin{bmatrix} 0.2693 \\ 0.6326 \end{bmatrix}$$

In this example, $\beta_p = 10^{-5}$ and $\sigma_n = 10^{-2}$. The results are tabulated in Table 4 for the case $\mu = 10^6$.

k	$\dfrac{\| d - Bx^{(k)} \|_2}{\| B \|_\infty \| x^{(k)} \|_2}$	$\dfrac{\| x^{(k)} - x_{LSE} \|_2}{\| x^{(k)} \|_2}$	μ_p estimate
2	10^{-11}	10^{-5}	10^4
3	10^{-13}	10^{-6}	10^4
4	10^{-15}	10^{-7}	10^4
5	10^{-17}	10^{-9}	10^4
6	10^{-19}	10^{-11}	10^4

Table 4.

We close with several remarks concerned with the implementation of the algorithm. The referee has suggested that it may be advisable to initially reduce A to upper triangular form via the QR factorization. In particular, if we compute an orthogonal Q such that

$$Q^T(A\ \ b) = \begin{array}{cc} \begin{bmatrix} R & s \\ 0 & t \end{bmatrix} & \begin{array}{c} n \\ m-n \end{array} \\ \begin{array}{cc} n & \ \ 1 \end{array} & \end{array}$$

then we can apply Algorithm 4.2 with $(A\ b)$ replaced by $(R\ s)$. The matrix Q need not be saved. The point of this reduction is that the matrix Q_μ is then of order $n+p$ rather than $m+p$ -- a dimension reduction that could be critical in large problems. Another fringe benefit of this maneuver is that A is "concentrated" into R before the (possibly contaminating) effects of μ are felt.

Focussing further on the large sparse case, we mention that Algorithm 4.2 can be implemented <u>without</u> storing the orthogonal matrix Q_μ. This follows because the correction vectors $\Delta x^{(k)}$ satisfy the normal equation

$$(A^TA + \mu^2B^TB)\ \Delta x^{(k)} = \mu^2\ B^T\ \delta_1^{(k)}$$

and thus

$$R_\mu^T R_\mu \Delta x^{(k)} = \mu^2\ B^T\ \delta_1^{(k)}\ . \tag{5.1}$$

Consequently, $\Delta x^{(k)}$ can be found by solving a pair of triangular systems. (Of course, Q_μ is needed to compute $x^{(1)} = x(\mu)$, but it can be multiplied into the right hand side $\begin{pmatrix}\mu d\\ b\end{pmatrix}$ as it is generated.)

We refer to (5.1) as a "semi-normal equation" because the factor R_μ is stably determined via orthogonalization. (In "ordinary" normal equations, one would find R_μ by performing the Cholesky factorization on $A^TA + \mu^2B^TB$.) Unfortunately, the usual pitfalls of normal equations plague semi-normal equations unless a follow-up step of iterative improvement is executed. This is detailed in Bjorck [1]. Applying his recommendations to our situation, we should compute $\Delta x^{(k)}$ as follows:

$$
\begin{aligned}
R_\mu^T R_\mu \Delta z^{(k)} &= \mu^2 B^T \delta_1^{(k)} \\
r^{(k)} &= \begin{pmatrix} \mu d \\ b \end{pmatrix} - \begin{pmatrix} \mu B \\ A \end{pmatrix} \Delta z^{(k)} \\
R_\mu^T R_\mu \Delta w^{(k)} &= \begin{pmatrix} \mu B \\ A \end{pmatrix}^T r^{(k)} \\
\Delta x^{(k)} &= \Delta z^{(k)} + \Delta w^{(k)}
\end{aligned}
$$

It can be shown that even in single precision, the $\Delta x^{(k)}$ produced in this fashion is as good as can be expected.

Acknowledgements

The author is indebted to Ake Bjorck and the referee for their numerous valuable suggestions.

REFERENCES

[1] Å.Björck (1978), "Comment on the Iterative Refinement of Least-Squares Solutions", J. American Statistical Association, 73, 161-166.

[2] Å.Björck (1981), "A general updating algorithm for constrained linear least squares problems", Report LITH-MAT-R-81-18, Departmentof Mathematics, University of Linkoping, Sweden.

[3] Å.Björck and G.H.Golub (1967), "Iterative refinement of linear least squares solutions by Householder transformations", BIT 7,327-337.

[4] Å. Björck and I.S.Duff (1980), "A direct method for the solution of sparse linear least squares problems", Lin.Alg.&Applic., 34, 43-67.

[5] R.P. Brent, F.Luk, and C. Van Loan (1982), "Computation of the generalized singular value decomposition using mesh-connected processors", Cornell Computer Science Technical Report TR 83-563, Ithaca, New York 14853.

[6] P. Businger and G.H. Golub (1965), "Linear least squares solutions by Householder matrices", Numer. Math., 7, 269-76.

[7] J.J. Dongarra, J.R. Bunch, C.B. Moler, G.W. Stewart (1978), <u>LINPACK Users Guide</u>, SIAM Publications, Philadelphia.

[8] L. Elden (1980), "Perturbation theory for the least squares problem with equality constraints", SIAM J.Numer. Anal., 17, 338-350.

[9] M. Gentleman and H.T. Kung (1981), "Matrix triangularization by systolic arrays"< in <u>Real Time Signal Processing IV</u>: <u>SPIE Proceedings Vol. 298</u>, Society of Photo-Optical Instrumentation Engineers, Bellingham, Washington, pp. 19-26.

[10] J.A. George and M.T. Heath (1980), "Solution of sparse least squares problems using Givens transformations", Lin.Alg.&Applic., 34, 69-84.

[11] G.H.Golub and R.J. Plemmons (1980), "Large-scale geodetic least squares adjustment by dissections and orthogonal decomposition", Lin.Alg.&Applic., 34, 3-27.

[12] G.H. Golub and C. Van Loan (1983), <u>Matrix Computations</u>, Johns-Hopkins University Press, Baltimore, Md.

[13] G.H. Golub and J.H. Wilkinson (1966), "Note on the Iterative Refinement of Least Squares Solution", Numer.Math., 9, 139-148.

[14] R.J. Hanson (1982), "Linear Least Squares with Bounds and Linear Constraints", Sandia Report SAND82-1517, Albuquerque, New Mexico.

[15] M.T. Heath, R.J. Plemmons, and R.C. Ward (1983), "Sparse orthogonal schemes for structural optimization using the force method", Research Report ORNL/CSD-119, Union Carbide, Nuclear Division, Oak Ridge Tennessee 37830.

[16] D.Heller and I. Ibsen (1983),"Systolic networks for orthogonal decompositions", SIAM J. Sci.&Stat. Comp.,4, 261-269.

[17] C.L. Lawson and R.J.Hanson (1974), Solving least squares problems, Prentice Hall, Englewood Cliffs NJ.

[18] C.B. Moler (1980), "MATLAB User's Guide", Technical Report CS81-1, Department of Computer Science, University of New Mexico.

[19] C.C. Paige and M.A. Saunders (1981) , "Towards a generalized singular value decomposition, SIAM J.Numer.Anal. 18, 398-405.

[20] A. Pothen (1984), "Sparse null bases and marraige theorems", Ph.D. Dissertation, Center for Applied Mathematics, Cornell University, Ithaca, New York 14853.

[21] M.J.D. Powell and J.K. Reid (1969), "On applying Householder's method to linear least squares problems", Proc. IFIP Congress, 1968.

[22] G.W. Stewart (1977) , "On the perturbation of pseudo-inverses, projections and linear least squares problems", SIAM Review, 19, 634-662.

[23] G.W. Stewart (1982), "Computing the CS decomposition of a partitioned orthogonal matrix", Numer. Math., 40, 297-306.

[24] C. Van Loan (1976), "Generalizing the singular value decomposition", SIAM J.Numer.Anal., 13, 76-83.

[25] C. Van Loan (1983), "A generalized SVD analysis of some weighting methods for equality constrained least squares", in Matrix Pencil Proceedings, Pite Havsbad, B. Kågström and A. Ruhe (eds), Lecture Notes in Mathematics #973 , Springer-Verlag New York.

[26] C. Van Loan (1984), "Computing the CS and the generalized singular value decompositions", Cornell Computer Science Technical Report 84-604, Ithaca, New York 14853.

[27] P.Å. Wedin (1979), " Notes on the constrained least squares problem---a new approach based on generalized inverses", Report UMINF 75.79, Institute for Information Processing, University of Umea, Sweden.