Contents

Chapter 8

Linear Least Squares Problems

8.1 Preliminaries

8.1.1 The Least Squares Principle

A fundamental task in scientific computing is to estimate parameters in a mathematical model from collected data which are subject to errors. The influence of the errors can be reduced by using a greater number of data than the number of unknowns. If the model is linear, the resulting problem is then to "solve" an in general inconsistent linear system $Ax = b$, where $A \in \mathbf{R}^{m\times n}$ and $m \geq n$. In other words, we want to find a vector $x \in \mathbf{R}^n$ such that Ax is in some sense the "best" approximation to the known vector $b \in \mathbf{R}^m$.

There are many possible ways of defining the "best" solution to an inconsistent linear system. A choice which can often be motivated for statistical reasons (see Theorem 8.1.4) and also leads to a simple computational problem is the following: Let x be a vector which minimizes the Euclidian length of the **residual vector** $r = b - Ax$; i.e., a solution to the minimization problem

$$\min_x \|Ax - b\|_2, \tag{8.1.1}$$

where $\|\cdot\|_2$ denotes the Euclidian vector norm. Note that this problem is equivalent to minimizing the sum of squares of the residuals $\sum_{i=1}^m r_i^2$. Hence, we call (8.1.1) a **linear least squares problem** and any minimizer x a **least squares solution** of the system $Ax = b$. [1]

Example 8.1.1. Consider a model described by a scalar function $y(t) = f(x, t)$, where $x \in \mathbf{R}^n$ is a parameter vector to be determined from measurements (y_i, t_i), $i = 1, \ldots, m$, $m > n$. In particular let $f(x, t)$ be *linear* in x,

$$f(x, t) = \sum_{j=1}^n x_j \phi_j(t).$$

[1] This draft last revised 2003 10 31.

Then the equations $y_i = \sum_{j=1}^{n} x_j \phi_j(t_i)$, $i = 1, \ldots, m$ form an overdetermined system, which can be written in matrix form $Ax = b$, where $a_{ij} = \phi_j(t_i)$, and $b_i = y_i$.

We shall see that a least squares solution x is characterized by $r \perp \mathcal{R}(A)$, where $\mathcal{R}(A)$ the range space of A. The residual vector r is always uniquely determined and the solution x is unique if and only if $\text{rank}\,(A) = n$, i.e., when A has linearly independent columns. If $\text{rank}\,(A) < n$, we seek the unique least squares solution of minimum Euclidean norm.

When there are more variables than needed to match the observed data, then we have an **underdetermined problem**. In this case we can seek the **minimum norm solution** $y \in \mathbb{R}^m$ of a linear system, i.e. solve

$$\min \|y\|_2, \qquad A^T y = c, \tag{8.1.2}$$

where $c \in \mathbb{R}^n$ and $A^T y = c$ is assumed to be consistent.

8.1.2 Linear Models and the Gauss–Markoff Theorem

We first need to introduce some concepts from statistics. Let the probability that random variable $y \leq x$ be equal to $F(x)$, where $F(x)$ is nondecreasing, right continuous, and satisfies

$$0 \leq F(x) \leq 1, \quad F(-\infty) = 0, \quad F(\infty) = 1.$$

Then $F(x)$ is called the **distribution function** for y.

The **expected value** and the **variance** of y are defined as the Stieltjes integrals

$$\mathcal{E}(y) = \mu = \int_{-\infty}^{\infty} y dF(y), \qquad \mathcal{E}(y-\mu)^2 = \sigma^2 = \int_{-\infty}^{\infty} (y-\mu)^2 dF(y),$$

If $y = (y_1, \ldots, y_n)^T$ is a vector of random variables and $\mu = (\mu_1, \ldots, \mu_n)^T$, $\mu_i = \mathcal{E}(y_i)$, then we write $\mu = \mathcal{E}(y)$. If y_i and y_j have the joint distribution $F(y_i, y_j)$ the **covariance** between y_i and y_j is

$$\begin{aligned} \sigma_{ij} &= \mathcal{E}[(y_i - \mu_i)(y_j - \mu_j)] = \int_{-\infty}^{\infty} (y_i - \mu_i)(y_j - \mu_j) dF(y_i, y_j) \\ &= \mathcal{E}(y_i y_j) - \mu_i \mu_j. \end{aligned}$$

The covariance matrix $V \in \mathbf{R}^{n \times n}$ of y is defined by

$$V = \mathcal{V}(y) = \mathcal{E}[(y-\mu)(y-\mu)^T] = \mathcal{E}(yy^T) - \mu\mu^T.$$

where the diagonal element σii is the variance of y_i.

We now prove some properties which will be useful in the following.

Lemma 8.1.1.
Let $B \in \mathbf{R}^{r\times n}$ be a matrix and y a random vector with $\mathcal{E}(y) = \mu$ and covariance matrix V. Then

$$\mathcal{E}(By) = B\mu, \qquad \mathcal{V}(By) = BVB^T.$$

In the special case that $B = b^T$ is a row vector, $r = 1$, then $\mathcal{V}(b^T y) = \mu\|b\|_2^2$.

Proof. The first property follows directly from the definition of expected value. The second follows from the relation

$$\begin{aligned}\mathcal{V}(By) &= \mathcal{E}[(B(y-\mu)(y-\mu)^T B^T] \\ &= B\mathcal{E}[(y-\mu)(y-\mu)^T]B^T = BVB^T.\end{aligned}$$

□

In linear statistical models one assumes that the vector $b \in \mathbf{R}^m$ of observations is related to the unknown parameter vector $x \in \mathbf{R}^n$ by a linear relationship

$$Ax = b + \epsilon, \tag{8.1.3}$$

where $A \in \mathbf{R}^{m\times n}$ is a known matrix, and, ϵ is a vector of random errors. In the **standard case** ϵ has zero mean and covariance matrix $\sigma^2 I$, i.e.,

$$\mathcal{E}(\epsilon) = 0, \qquad \mathcal{V}(\epsilon) = \sigma^2 I.$$

We also assume that rank $(A) = n$, and make the following definitions:

Definition 8.1.2.
A function g of the random vector y is called unbiased estimate of a parameter θ if $\mathcal{E}(g(y)) = 0$. When such a function exists, then θ is called an estimable parameter.

Definition 8.1.3.
The linear function $g = c^T y$, where c is a constant vector, is a minimum variance (best) unbiased estimate of the parameter θ if $\mathcal{E}(g) = \theta$, and $\mathcal{V}(g)$ is minimized over all linear estimators.

Gauss gave the method of least squares a sound theoretical basis in [23, 1821], without any assumptions that the random variables follow a normal distribution. This contribution of Gauss was somewhat neglected until rediscovered by Markoff 1912. We state the relevant theorem without proof.

Theorem 8.1.4. The Gauss–Markoff theorem.
Consider the linear model (8.1.3)*, where $A \in \mathbf{R}^{m\times n}$ is a known matrix, and ϵ is a random vector with zero mean and covariance matrix $\mathcal{V}(\epsilon) = \sigma^2 I$. Let $\hat{x}$ be the least square estimator, obtained by minimizing over x the sum of squares*

$\|Ax-b\|_2^2$. *Then the best linear unbiased estimator of any linear function* $g = c^T x$ *is* $c^T\hat{x}$. *Furthermore, the covariance matrix of the estimate* $\hat{x}$ *equals*

$$\mathcal{V}(\hat{x}) = V = \sigma^2(A^TA)^{-1} \tag{8.1.4}$$

and $\mathcal{E}(s^2) = \sigma^2$, *where* s^2 *is the quadratic form*

$$s^2 = \frac{1}{m-n}\|b - A\hat{x}\|_2^2.$$

Proof. See Zelen [67]. □

In the next subsection we show that the residual vector $\hat{r} = \hat{b} - Ax$ satisfies $A^T\hat{r} = 0$. Hence there are n linear relations among the m components of $\hat{r}$. It can be shown that the residuals $\hat{r}$ and therefore also s^2 are uncorrelated with $\hat{x}$, i.e.,

$$\mathcal{V}(\hat{r}, \hat{x}) = 0, \qquad \mathcal{V}(s^2, \hat{x}) = 0.$$

In the **general univariate linear model** the covariance matrix equals $\mathcal{V}(\epsilon) = \sigma^2 W$, where $W \in \mathbf{R}^{m\times m}$ is a positive semidefinite symmetric matrix. For full column rank A and positive definite W the best unbiased linear estimate is the solution of

$$\min_x (Ax-b)^T W^{-1}(Ax-b). \tag{8.1.5}$$

In particular, if the errors are uncorrelated with variances $w_{ii} > 0$, $i = 1, \ldots, m$, then W is diagonal and the best estimate is obtained form the problem the **weighted least squares** problem

$$\min_x \|D^{-1}(Ax-b)\|_2, \quad D = \operatorname{diag}(\sqrt{w_{11}}, \ldots, \sqrt{w_{mm}}). \tag{8.1.6}$$

Hence if the ith equation is scaled by $1/\sqrt{w_{ii}}$ we get the standard case. This is consistent with the obvious observation that the larger the variance the smaller weight should be given to a particular equation. It is important to note that different scalings will give different solutions, unless the system is *consistent*, i.e., $b \in \mathcal{R}(A)$.

8.1.3 Generalized Inverses

IN

The SVD is a powerful tool both for analyzing and solving linear least squares problems. The reason for this is that the orthogonal matrices that transform A to diagonal form do not change the l_2-norm. We have the following fundamental result.

Theorem 8.1.5.
Let $A \in \mathbf{R}^{m\times n}$, $\operatorname{rank}(A) = r$, *and consider the general linear least squares problem*

$$\min_{x\in S}\|x\|_2, \qquad S = \{x \in \mathbf{R}^n \mid \|b - Ax\|_2 = \min\}. \tag{8.1.7}$$

This problem always has a unique solution, which in terms of the SVD of A can be written as

$$x = V \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T b, \tag{8.1.8}$$

Proof. Let

$$c = U^T b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix},$$

where $z_1, c_1 \in \mathbf{R}^r$. Using the orthogonal invariance of the l_2 norm we have

$$\begin{aligned} \|b - Ax\|_2 &= \|U^T(b - AVV^T x)\|_2 \\ &= \left\| \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} - \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} c_1 - \Sigma_1 z_1 \\ c_2 \end{pmatrix} \right\|_2. \end{aligned}$$

The residual norm will attain its minimum value equal to $\|c_2\|_2$ for $z_1 = \Sigma_1^{-1} c_1$, z_2 arbitrary. Obviously the choice $z_2 = 0$ minimizes $\|x\|_2 = \|Vz\|_2 = \|z\|_2$. □

Note that problem (8.1.7) includes as special cases the solution of both overdetermined and underdetermined linear systems. We can write $x = A^\dagger b$, where

$$A^\dagger = V \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T \in \mathbf{R}^{n \times m} \tag{8.1.9}$$

is the unique **pseudo-inverse** of A and x is called the pseudo-inverse solution of $Ax = b$.

Methods for computing the SVD are described in Sec. 10.8. Note that for solving least squares problems we only need to compute the singular values, the matrix V_1 and vector $c = U_1^T b$, where we have partitioned $U = (U_1\ U_2)$ and $V = (V_1\ V_2)$ so that U_1 and V_1 have $r = \operatorname{rank}(A)$ columns. The pseudo-inverse solution (8.1.9) can then be written

$$x = V_1 \Sigma_1^{-1} U_1^T b = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} \cdot v_i, \quad r = \operatorname{rank}(A). \tag{8.1.10}$$

The matrix $A^\dagger$ is often called the **Moore–Penrose inverse**. Moore 1920 developed the concept of the general reciprocal in 1920. Penrose [1955], gave an elegant algebraic characterization and showed that $X = A^\dagger$ is uniquely determined by the four **Penrose conditions** :

$$(1) \quad AXA = A, \qquad (2) \quad XAX = X, \tag{8.1.11}$$

$$(3) \quad (AX)^T = AX, \qquad (4) \quad (XA)^T = XA. \tag{8.1.12}$$

It can be directly verified that $X = A^\dagger$ given by (8.1.9) satisfies these four conditions. In particular this shows that $A^\dagger$ does not depend on the particular choices of U and V in the SVD. (See also Problem 2.)

The orthogonal projections onto the four fundamental subspaces of A have the following simple expressions in terms of the pseudo-inverse :

$$P_{\mathcal{R}(A)} = AA^\dagger, \qquad P_{\mathcal{N}(A^T)} = I - AA^\dagger, \tag{8.1.13}$$
$$P_{\mathcal{R}(A^T)} = A^\dagger A, \qquad P_{\mathcal{N}(A)} = I - A^\dagger A.$$

These expressions are easily verified using the definition of an orthogonal projection and the Penrose conditions.

Another very useful characterization of the pseudo-inverse solution is the following:

Theorem 8.1.6. *The pseudo-inverse solution $x = A^\dagger b$ is uniquely characterized by the two geometrical conditions*

$$x \perp \mathcal{N}(A), \qquad Ax = P_{\mathcal{R}(A)}\, b. \tag{8.1.14}$$

Proof. These conditions are easily verified from (8.1.10). □

In the special case that $A \in \mathbf{R}^{m\times n}$ and $\operatorname{rank}(A) = n$ it holds that

$$A^\dagger = (A^TA)^{-1}A^T, \qquad (A^T)^\dagger = A(A^TA)^{-1} \tag{8.1.15}$$

These expressions follow from the normal equations (8.2.3) and (8.2.4). Some properties of the usual inverse can be extended to the pseudo-inverse, e.g., the relations

$$(A^\dagger)^\dagger = A, \qquad (A^T)^\dagger = (A^\dagger)^T,$$

easily follow form (8.1.9). In general $(AB)^\dagger \neq B^\dagger A^\dagger$. The following theorem gives a useful *sufficient* conditions for the relation $(AB)^\dagger = B^\dagger A^\dagger$ to hold.

Theorem 8.1.7.
If $A \in \mathbf{R}^{m\times r}$, $B \in \mathbf{R}^{r\times n}$, and $\operatorname{rank}(A) = \operatorname{rank}(B) = r$, then

$$(AB)^\dagger = B^\dagger A^\dagger = B^T(BB^T)^{-1}(A^TA)^{-1}A^T. \tag{8.1.16}$$

Proof. The last equality follows from (8.1.15). The first equality is verified by showing that the four Penrose conditions are satisfied. □

A matrix X which only satisfy some of the Penrose conditions is called a **generalized inverse**. A matrix X is called an **inner inverse** or $\{1\}$-inverse if it satisfies condition (1). Any matrix X which satisfies condition (2) is called an **outer inverse** or a $\{2\}$-inverse. A matrix which satisfies conditions (1) and (3), is called a $\{1,3\}$-inverse, etc.

Let X be a $\{1\}$-inverse of $A \in \mathbf{C}^{m\times n}$. Then for all b such that $Ax = b$ is consistent $x = Xb$ is a solution. The general solution can be written

$$x = Xb + (I - XA)y, \quad y \in \mathbf{C}^n.$$

Let $A \in \mathbf{R}^{m\times n}$ of rank r and X an $\{1\}$-inverse. Then $AXA = A$ and we have

$$(AX)^2 = AXAX = AX, \qquad (XA)^2 = XAXA = XA.$$

This shows that AX and XA are idempotent and therefore (in general oblique) projectors

$$AX = P_{\mathcal{R}(A),S}, \qquad XA = P_{T,\mathcal{N}(A)},$$

where S and T are some subspaces complementary to $\mathcal{R}(A)$ and $\mathcal{N}(A)$, respectively.

If A is a $\{1,3\}$-inverse, then AX is symmetric and therefore is the orthogonal projector onto $\mathcal{R}(A)$. Similarly, if A is a $\{1,4\}$-inverse, then XA is symmetric and therefore the orthogonal projector orthogonal to $\mathcal{N}(A)$.

Theorem 8.1.8.

Let $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$. Then $\|Ax - b\|_2$ is the smallest when $x = Xb$, where X is a $\{1,3\}$-inverse.

Conversely, if $X \in \mathbf{R}^{n\times m}$ has the property that for all b, $\|Ax-b\|_2$ is smallest when $x = Xb$, then X is a $\{1,3\}$-inverse.

Theorem 8.1.9.

Let $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$. If $Ax = b$ has a solution, the unique solution for which $\|x\|_2$ is smallest is given by $x = Xb$, where X is a $\{1,4\}$-inverse.

Conversely, if $X \in \mathbf{R}^{n\times m}$ is such that, whenever $Ax = b$ has a solution, $x = Xb$ is the solution of smallest norm, then X is a $\{1,4\}$-inverse.

8.1.4 Matrix Approximation and the SVD

A useful relationship between the SVD and a symmetric eigenvalue problem is given in the following theorem.

Theorem 8.1.10. *Let the SVD of $A \in \mathbf{R}^{m\times n}$ be $A = U\Sigma V^T$, where $U = \mathbf{R}^{m\times m}$ and $V \in \mathbf{R}^{n\times n}$ are orthogonal. Let $r = \operatorname{rank}(A) \le \min(m,n)$ and $\Sigma_1 = \operatorname{diag}(\sigma_1,\dots,\sigma_r) > 0$. Then it holds that*

$$C = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} = Q \begin{pmatrix} \Sigma & 0 & 0 \\ 0 & -\Sigma & 0 \\ 0 & 0 & 0 \end{pmatrix} Q^T, \tag{8.1.17}$$

where

$$Q = \frac{1}{\sqrt{2}} \begin{pmatrix} U_1 & U_1 & \sqrt{2}\,U_2 & 0 \\ V_1 & -V_1 & 0 & \sqrt{2}\,V_2 \end{pmatrix}. \tag{8.1.18}$$

and U and V have been partitioned conformally. Hence the eigenvalues of C are $\pm\sigma_1, \pm\sigma_2, \dots, \pm\sigma_r$, and zero repeated $(m+n-2r)$ times.

Proof. Form the product on the right hand side of (8.1.17) and note that $A = U_1\Sigma_1 V_1^T$ and $A^T = V_1\Sigma_1 U_1^T$. □

The singular values have the following important extremal property, the **minimax characterization**.

Theorem 8.1.11.
Let $A \in \mathbf{R}^{m\times n}$ have singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$, $p = \min(m,n)$, and S be a linear subspace of $\mathbf{R}^n$ of dimension $\dim(S)$. Then

$$\sigma_i = \min_{dim(S)=n-i+1} \max_{\substack{x\in S \\ x\neq 0}} \frac{\|Ax\|_2}{\|x\|_2}. \tag{8.1.19}$$

Proof. The result is established in almost the same way as for the corresponding eigenvalue theorem, Theorem 10.3.9 (Fischer's theorem). □

The minimax characterization of the singular values may be used to establish the following relations between the singular values of two matrices A and B.

Theorem 8.1.12.
Let $A, B \in \mathbf{R}^{m\times n}$ have singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p$ and $\tau_1 \geq \tau_2 \geq \ldots \geq \tau_p$ respectively, where $p = \min(m,n)$. Then

$$\max_i |\sigma_i - \tau_i| \leq \|A - B\|_2, \tag{8.1.20}$$

$$\sum_{i=1}^{p} |\sigma_i - \tau_i|^2 \leq \|A - B\|_F^2. \tag{8.1.21}$$

Proof. See Stewart [1973, pp. 321-322]. □

Hence perturbations of the elements of a matrix A result in perturbations of the same, or smaller, magnitude in the singular values. This result is important for the use of the SVD to determine the "numerical rank" of a matrix; see below.

The eigenvalues of the leading principal minor of order $n-1$ of a Hermitian matrix C can be shown to interlace the eigenvalues of C, see Theorem 10.3.8. From the relation (8.1.17) corresponding results can be derived for the singular values of a matrix A.

Theorem 8.1.13.
Let

$$\hat{A} = (A, u) \in \mathbf{R}^{m\times n}, \quad m \geq n, \quad u \in \mathbf{R}^m.$$

Then the ordered singular values σ_i of A interlace the ordered singular values $\hat{\sigma}_i$ of $\hat{A}$ as follows

$$\hat{\sigma}_1 \geq \sigma_1 \geq \hat{\sigma}_2 \geq \sigma_2 \ldots \geq \hat{\sigma}_{n-1} \geq \sigma_{n-1} \geq \hat{\sigma}_n.$$

Similarly, if A is bordered by a row,

$$\hat{A} = \begin{pmatrix} A \\ v^* \end{pmatrix} \in \mathbf{R}^{m\times n}, \quad m > n, \quad v \in \mathbf{R}^n,$$

then

$$\hat{\sigma}_1 \geq \sigma_1 \geq \hat{\sigma}_2 \geq \sigma_2 \ldots \geq \hat{\sigma}_{n-1} \geq \sigma_{n-1} \geq \hat{\sigma}_n \geq \sigma_n.$$

The SVD plays an important role in a number of matrix approximation problems. In the theorem below we consider the approximation of one matrix by another of lower rank.

Theorem 8.1.14. *Let $\mathcal{M}_k^{m\times n}$ denote the set of matrices in $\mathbf{R}^{m\times n}$ of rank k. Assume that $A \in \mathcal{M}_r^{m\times n}$ and consider the problem*

$$\min_{X\in\mathcal{M}_k^{m\times n}} \|A - X\|, \quad k < r.$$

Then the SVD expansion of A truncated to k terms $X = B = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, solves this problem both for the l_2 norm and the Frobenius norm. Further, the minimum distance is given by

$$\|A - B\|_2 = \sigma_{k+1}, \qquad \|A - B\|_F = (\sigma_{k+1}^2 + \ldots + \sigma_r^2)^{1/2}.$$

The solution is unique for the Frobenius norm but not always for the l_2 norm.

Proof. See Mirsky [42] for the l_2, norm and Eckhard and Young[20] for the Frobenius norm. □

According to this theorem σ_i equals the distance in l_2 norm to the nearest matrix of rank $i-1$, $i \leq \min(m,n)$. In particular $\sigma_1 = \|A\|_2$.

Inaccuracy of data and rounding errors made during the computation usually perturb the ideal matrix A. In this situation the *mathematical* notion of rank may not be appropriate. For example, let A be a matrix of rank $r < n$, whose elements are perturbed by a matrix E of small random errors. Then it is most likely that the perturbed matrix $A + E$ has full rank n. However, $A + E$ is close to a rank deficient matrix, and should be considered as *numerically rank deficient*.

Clearly the **numerical rank** assigned to a matrix should depend on some tolerance δ, which reflects the error level in the data and/or the precision of the floating point arithmetic used. A useful definition is the following:

Definition 8.1.15.
A matrix $A \in \mathbf{R}^{m\times n}$ has numerical δ-rank equal to k ($k \leq \min\{m,n\}$) if

$$\sigma_1 \geq \ldots \geq \sigma_k > \delta \geq \sigma_{k+1} \geq \ldots \geq \sigma_n,$$

where σ_i, $i = 1, 2, \ldots, n$ are the singular values of A. If we write

$$A = U\Sigma V^T = U_1\Sigma_1 V_1^T + U_2\Sigma_2 V_2^T,$$

where $\Sigma_2 = \operatorname{diag}(\sigma_{k+1}, \ldots, \sigma_n)$ then $\mathcal{R}(V_2) = \operatorname{span}\{v_{k+1}, \ldots, v_n\}$ is called the **numerical nullspace** *of A.*

It follows from Theorem 8.1.12, that if the numerical δ-rank of A equals k, then $\operatorname{rank}(A+E) \geq k$ for all perturbations such that $\|E\|_2 \leq \delta$, i.e., such perturbations cannot *lower* the rank. Definition 8.1.15 is only useful when there is a well defined gap between σ_{k+1} and σ_k. This should be the case if the exact matrix A is rank deficient but well-conditioned. However, it may occur that there does not exist a gap for any k, e.g., if $\sigma_k = 1/k$. In such a case the numerical rank of A is not well defined!

If $r < n$ then the system is *numerically underdetermined*. Note that this can be the case even when $m > n$.

Let $A \in \mathbf{R}^{m\times n}$, be a matrix of rank n with the "thin" SVD $A = U_1 \Sigma V^T$. Since $A = U_1\Sigma V^T = U_1 \Sigma U_1^T U_1 V^T$ we have

$$A = PH, \qquad P = U_1 V^T, \qquad H = V\Sigma V^T, \tag{8.1.22}$$

where $P \in \mathbf{R}^{m\times n}$ has orthogonal columns, and $H \in \mathbf{R}^{n\times n}$ is symmetric, positive semidefinite. The decomposition (8.1.22) is called the **polar decomposition** of A, since it can be regarded as a generalization to matrices of the complex number representation $z = re^{i\theta}$, $r \geq 0$.

The significance of the factor P in the polar decomposition is that it is the closest matrix with orthogonal columns to A.

Theorem 8.1.16.

Let $\mathcal{M}_{m\times n}$ denote the set of all matrices in $\mathbf{R}^{m\times n}$ with orthogonal columns. Let $A \in \mathbf{R}^{m\times n}$ be a given matrix and $A = PH$ its polar decomposition, where $P \in \mathcal{M}_{m\times n}$ and H is symmetric positive semidefinite. Then for any matrix $Q \in \mathcal{M}_{m\times n}$,

$$\|A - Q\|_F \geq \|A - P\|_F.$$

Proof. This theorem was proved for $m = n$ and general unitarily invariant norms by Fan and Hoffman [21]. The generalization to $m > n$ follows from the additive property of the Frobenius norm. □

An generalization of Theorem 8.1.16 has important application in factor analysis in statistics.

Theorem 8.1.17.

Let $\mathcal{M}_{m\times n}$ denote the set of all matrices in $\mathbf{R}^{m\times n}$ with orthogonal columns. Let A and B be given matrices in $\mathbf{R}^{m\times n}$. If $B^T A = PH$ is the polar decomposition then for any matrix $Q \in \mathcal{M}_{m\times n}$ it holds that

$$\|A - BQ\|_F \geq \|A - BP\|_F.$$

Proof. See P. Schönemann [54]. □

8.1.5 Perturbation Analysis

We now consider the effect of perturbations of A and b on the least squares solution x. In this analysis the condition number of the matrix $A \in \mathbf{R}^{m\times n}$ will play a significant role. The following definition generalizes the condition number (6.6.3) of a square nonsingular matrix.

Definition 8.1.18.
Let $A \in \mathbf{R}^{m\times n}$ have rank $r > 0$ and singular values equal to $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$. Then the condition number of A is

$$\kappa(A) = \|A\|_2\|A^\dagger\|_2 = \sigma_1/\sigma_r,$$

where the last equality follows from the relations $\|A\|_2 = \sigma_1$, $\|A^\dagger\|_2 = \sigma_r^{-1}$.

Using the singular value decomposition $A = U\Sigma V^T$ we obtain

$$A^TA = V\Sigma^T(U^TU)\Sigma V^T = V\begin{pmatrix}\Sigma_r^2 & 0\\ 0 & 0\end{pmatrix}V^T. \tag{8.1.23}$$

Hence, $\sigma_i(A^TA) = \sigma_i^2(A)$, and it follows that

$$\kappa(A^TA) = \kappa^2(A).$$

This shows that *the matrix of the normal equations has a condition number which is the square of the condition number of A.*

We now give a first order perturbation analysis for the least squares problem when $\operatorname{rank}(A) = n$. Denote the perturbed data $A+\delta A$ and $b+\delta b$ and assume that δA sufficiently small so that we have $\operatorname{rank}(A+\delta A) = n$. Let the perturbed solution be $x + \delta x$ and $r + \delta r$, where $r = b - Ax$ is the residual vector. Then, neglecting second order perturbations, we have

$$\delta r = \delta b - (A + \delta A)(x + \delta x) = (\delta b - \delta A\, x) - A\delta x.$$

The perturbed solution satisfies

$$(A + \delta A)^T\big((A + \delta A)(x + \delta x) - (b + \delta b)\big) = 0.$$

Subtracting $A^T(Ax - b) = 0$ and neglecting second order perturbations, we get

$$\delta x = (A^TA)^{-1}A^T(\delta b - \delta A\, x) + (A^TA)^{-1}\delta A^T r, \tag{8.1.24}$$
$$\delta r = (I - A(A^TA)^{-1}A^T)(\delta b - \delta Ax) - A(A^TA)^{-1}\delta A^T\, r, \tag{8.1.25}$$

Here we can identify

$$(A^TA)^{-1}A^T = A^\dagger, \qquad A(A^TA)^{-1} = (A^\dagger)^T,$$
$$I - A(A^TA)^{-1}A^T = I - AA^\dagger = P_{\mathcal{N}(A^T)}.$$

Using (8.1.9) and (8.1.23) it follows that

$$\|A^\dagger\|_2 = \|(A^\dagger)^T\|_2 = 1/\sigma_n, \qquad \|(A^TA)^{-1}\|_2 = 1/\sigma_n^2, \qquad \|P_{\mathcal{N}(A^T)}\|_2 = 1.$$

Hence, taking norms in (8.1.25) and (8.1.25) we obtain

$$\|\delta x\|_2 \lessapprox \frac{1}{\sigma_n}\|\delta b\|_2 + \frac{1}{\sigma_n}\|\delta A\|_2\left(\|x\|_2 + \frac{1}{\sigma_n}\|r\|_2\right), \tag{8.1.26}$$

$$\|\delta r\|_2 \lessapprox \|\delta b\|_2 + \|\delta A\|_2\left(\|x\|_2 + \frac{1}{\sigma_n}\|r\|_2\right), \tag{8.1.27}$$

A more refined perturbation analysis (see Wedin [65]) shows that if

$$\eta = \|A^\dagger\|_2\|\delta A\|_2 \ll 1.$$

then $\operatorname{rank}(A+\delta A) = n$, and there are perturbations δA and δb such that these upper bounds are almost attained.

Assuming that $x \neq 0$ and setting $\delta b = 0$, we get a bound for the normwise relative perturbation

$$\frac{\|\delta x\|_2}{\|x\|_2} \leq \kappa(A)\frac{\|\delta A\|_2}{\|A\|_2}\Big(1 + \frac{\|r\|_2}{\sigma_n^2\|x\|_2}\Big) \tag{8.1.28}$$

Note that if the system $Ax = b$ is consistent, then $r = 0$ and the bound is identical to that obtained for a square nonsingular linear system. Otherwise, there is a second term present in the perturbation bound.

An upper bound for the condition number for x in the least squares problems with respect to A is

$$\kappa_{LS} = \kappa(A)\Big(1 + \frac{\|r\|_2}{\sigma_n\|x\|_2}\Big) \tag{8.1.29}$$

The two following facts should be noted:

- κ_{LS} depends not only on A but also on r and therefore on b;
- If $\|r\|_2 \ll \sigma_n\|x\|_2$ then $\kappa_{LS} \approx \kappa(A)$, but if $\|r\|_2 > \sigma_n\|x\|_2$ the second term in (8.1.29) will dominate,

Example 8.1.2. The following simple example illustrates the perturbation analysis above. Consider a least squares problem with

$$A = \begin{pmatrix} 1 & 0 \\ 0 & \delta \\ 0 & 0 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 0 \\ \alpha \end{pmatrix}, \qquad \delta A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \delta/2 \end{pmatrix}.$$

and $\kappa(A) = 1/\delta \gg 1$. If $\alpha = 1$ then

$$x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \delta x = \frac{2}{5\delta}\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad r = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \delta r = -\frac{1}{5}\begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}.$$

For this right hand side $\|x\|_2 = \|r\|_2$ and $\kappa_{LS} = 1/\delta + 1/\delta^2 \approx \kappa^2(A)$. This is reflected in the size of δx.

If instead we take $\alpha = \delta$, then a short calculation shows that $\|r\|_2/\|x\|_2 = \delta$ and $\kappa_{LS} = 2/\delta$. The same perturbation δA now gives

$$\delta x = \frac{2}{5}\begin{pmatrix}0\\1\end{pmatrix}, \qquad \delta r = -\frac{\delta}{5}\begin{pmatrix}0\\2\\1\end{pmatrix}.$$

It should be stressed that in order for the perturbation analysis above to be useful, the matrix A and vector b should be scaled so that perturbations are "well defined" by bounds on $\|\delta A\|_2$ and $\|b\|_2$. If the columns in $A = (a_1, a_2, \ldots, a_n)$ have widely differing norms, then a much better estimate may often be obtained by applying (8.1.28) to the scaled problem $\min_{\tilde{x}} \|\tilde{A}\tilde{x} - b\|_2$, chosen so that $\tilde{A}$ has columns of unit length, i.e.,

$$\tilde{A} = AD^{-1}, \qquad \tilde{x} = Dx, \qquad D = \operatorname{diag}(\|a_1\|_2, \ldots, \|a_n\|_2).$$

By Theorem 8.2.5 this column scaling approximately minimizes $\kappa(AD^{-1})$ over $D > 0$. Note however that scaling the columns also changes the norm in which the error in x is measured.

If the *rows* in A differ widely in norm, then (8.1.28) may also considerably overestimate the perturbation in x. As remarked above, we cannot scale the rows in A without changing the least squares solution.

Perturbation bounds with better scaling properties can be obtained by considering component-wise perturbations.

$$|\delta A| \le \omega E, \qquad |\delta b| \le \omega f. \tag{8.1.30}$$

Substituting in (8.1.25)–(8.1.25) yields the bounds

$$|\delta x| \lessapprox \omega \left(|A^\dagger|(f + E|x|) + |(A^TA)^{-1}|E^T|r| \right), \tag{8.1.31}$$

$$|\delta r| \lessapprox \omega \left(|I - AA^\dagger|(f + E|x|) + |(A^\dagger)^T|E^T|r| \right). \tag{8.1.32}$$

where terms of order $O(\omega^2)$ have been neglected. In particular, if $E = |A|$, $f = |b|$, we obtain taking norms

$$\|\delta x\| \lessapprox \omega \left(\| \, |A^\dagger|(|b| + |A||x|)\| + \| \, |(A^TA)^{-1}||A|^T|r| \, \| \right), \tag{8.1.33}$$

$$\|\delta r\| \lessapprox \omega \left(\| \, |I - AA^\dagger|(|A||x| + |b|)\| + \| \, |(A^\dagger)^T||A|^T|r| \, \| \right). \tag{8.1.34}$$

8.1.6 Backward Error and Stability

An algorithm for solving the linear least squares problem is said to numerically stable if for any data A and b, there exist small perturbation matrices and vectors δA and δb, such that the computed solution $\bar{x}$ is the *exact* solution to

$$\min_x \|(A + \delta A)x - (b + \delta b)\|_2, \tag{8.1.35}$$

where $\|\delta A\| \leq \tau$, $\|\delta b\| \leq \tau$, with τ being a small multiple of the unit roundoff u. Methods which explicitly form the normal equations are not backward stable. However, many methods based on orthogonal factorizations have been proved to be numerically stable with $\delta b = 0$.

Any computed solution $\bar{x}$ is called a stable solution if it satisfies (8.1.35). This does not mean that $\bar{x}$ is close to the exact solution x. If the least squares problem is ill-conditioned then a stable solution can be very different from x. For a stable solution the error $\|x - \bar{x}\|$ can be estimated using the perturbation results given in Section 8.1.5.

Many special fast methods exist for solving structured least squares problems, e.g., where A is a Toeplitz matrix. These methods cannot be proved to be backward stable, which is one reason why a solution to the following problem is of interest:

Given an alleged solution $\tilde{x}$, find the smallest backward error, i.e. a perturbation δA of smallest norm such that $\tilde{x}$ is the exact solution to the perturbed problem

$$\min_x \|(b + \delta b) - (A + \delta A)x\|_2. \tag{8.1.36}$$

If we could find the backward error of smallest norm, this could be used to verify numerically the stability properties of an algorithm. There is not much loss in assuming that $\delta b = 0$ in (8.1.37). Then the optimal backward error in the Frobenius norm is

$$\eta_F(\tilde{x}) = \min\{\|\delta A\|_F \mid \tilde{x} \text{ solves } \min_x \|b - (A + \delta A)x\|_2\}. \tag{8.1.37}$$

This the optimal backward error can be found by characterizing the set of all backward perturbations and then finding an optimal bound, which minimizes the Frobenius norm.

Theorem 8.1.19. *Let $\tilde{x}$ be an alleged solution and $\tilde{r} = b - A\tilde{x} \neq 0$. The optimal backward error in the Frobenius norm is*

$$\eta_F(\tilde{x}) = \begin{cases} \|A^T\tilde{r}\|_2/\|\tilde{r}\|_2, & \text{if } \tilde{x} = 0, \\ \min\{\eta, \sigma_{min}([A \quad C])\} & \text{otherwise.} \end{cases} \tag{8.1.38}$$

where

$$\eta = \|\tilde{r}\|_2/\|\tilde{x}\|_2, \qquad C = I - (\tilde{r}\tilde{r}^T)/\|\tilde{r}\|_2^2$$

and $\sigma_{min}([A \quad C])$ denotes the smallest (nonzero) singular value of the matrix $[A \quad C] \in \mathbb{R}^{m\times(n+m)}$.

The task of computing $\eta_F(\tilde{x})$ is thus reduced to that of computing $\sigma_{min}(\mathcal{A})$. Since this is expensive, approximations that are accurate and less costly have been derived. If a QR factorization of A is available lower and upper bounds for $\eta_F(\tilde{x})$ can be computed in only $\mathcal{O}(mn)$ operations. Let $r_1 = P_{\mathcal{R}(A)}\tilde{r}$ be the orthogonal projection of $\tilde{r}$ onto the range of A. If $\|r_1\|_2 \leq \alpha\|r\|_2$ it holds that

$$\frac{\sqrt{5}-1}{2}\,\tilde{\sigma}_1 \leq \eta_F(\tilde{x}) \leq \sqrt{1+\alpha^2}\,\tilde{\sigma}_1, \tag{8.1.39}$$

where

$$\tilde{\sigma}_1 = \left\|(A^TA + \eta I)^{-1/2}A^T\tilde{r}\right\|_2 / \|\tilde{x}\|_2. \tag{8.1.40}$$

Since $\alpha \to 0$ for small perturbations $\tilde{\sigma}_1$ is an asymptotic upper bound.

Review Questions

1. State the Gauss–Markov theorem.

2. Assume that A has full column rank. Show that the matrix $P = A(A^TA)^{-1}A^T$ is symmetric and satisfies the condition $P^2 = P$.

3. (a) Give conditions for a matrix P to be the orthogonal projector onto a subspace $S \in \mathbf{R}^n$.

(b) Define the orthogonal complement of S in $\mathbf{R}^n$.

4. (a) Which are the four fundamental subspaces of a matrix? Which relations hold between them? Express the orthogonal projections onto the fundamental subspaces in terms of the SVD.

(b) Give two geometric conditions which are necessary and sufficient conditions for x to be the pseudo-inverse solution of $Ax = b$.

5. Which of the following relations are universally correct?

(a) $\mathcal{N}(B) \subseteq \mathcal{N}(AB)$. (b) $\mathcal{N}(A) \subseteq \mathcal{N}(AB)$. (c) $\mathcal{N}(AB) \subseteq \mathcal{N}(A)$.

(d) $\mathcal{R}(AB) \subseteq \mathcal{R}(B)$. (e) $\mathcal{R}(AB) \subseteq \mathcal{R}(A)$. (f) $\mathcal{R}(B) \subseteq \mathcal{R}(AB)$.

6. (a) What are the four Penrose conditions for X to be the pseudo-inverse of A?

(b)A matrix X is said to be a **left-inverse** if $XA = I$. Show that a left-inverse is an $\{1,2,3\}$-inverse, i.e. satisfies the Penrose conditions (1), (2), and (3). Similarly show that a **right-inverse** is an $\{1,2,4\}$-inverse.

7. Let the singular values of $A \in \mathbf{R}^{m\times n}$ be $\sigma_1 \geq \cdots \geq \sigma_n$. What relations are satisfied between these and the singular values of

$$\tilde{A} = (A, u), \qquad \hat{A} = \begin{pmatrix} A \\ v^T \end{pmatrix}?$$

8. (a) Show that $A^\dagger = A^{-1}$ when A is a nonsingular matrix.

(b) Construct an example where $G \neq A^\dagger$ despite the fact that $GA = I$.

Problems

1. (a) Compute the pseudo-inverse $x^\dagger$ of a column vector x.

(b) Take $A = (\,1 \quad 0\,)$, $B = (\,1 \quad 1\,)^T$, and show that $1 = (AB)^\dagger \neq B^\dagger A^\dagger = 1/2$.

2. (a) Verify that the Penrose conditions uniquely defines the matrix X. Do it first for $A = \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$, and then transform the result to a general matrix A.

3 (a) Show that if $w \in \mathbf{R}^n$ and $w^T w = 1$, then the matrix $P(w) = I - 2ww^T$ is both symmetric and orthogonal.

(b) Given two vectors $x, y \in \mathbf{R}^n$, $x \neq y$, $\|x\|_2 = \|y\|_2$, then

$$P(w)x = y, \qquad w = (y - x)/\|y - x\|_2.$$

4. Let $S \subseteq \mathbf{R}^n$ be a subspace, P_1 and P_2 be orthogonal projections onto $S = \mathcal{R}(P_1) = \mathcal{R}(P_2)$. Show that $P_1 = P_2$, i.e., the orthogonal projection onto S is unique.

Hint: Show that for any $z \in \mathbf{R}^n$

$$\|(P_1 - P_2)z\|_2^2 = (P_1 z)^T (I - P_2)z + (P_2 z)^T (I - P_1)z = 0.$$

5. (R. E. Cline) Let A and B be any matrices for which the product AB is defined, and set

$$B_1 = A^\dagger AB, \qquad A_1 = AB_1 B_1^\dagger.$$

Show that $AB = AB_1 = A_1 B_1$ and that $(AB)^\dagger = B_1^\dagger A_1^\dagger$.

Hint: Use the Penrose conditions.

6. (a) Show that the matrix $A \in \mathbf{R}^{m\times n}$ has a **left inverse** $A^L \in \mathbf{R}^{n\times m}$, i.e., $A^L A = I$, if and only if $\operatorname{rank}(A) = n$. Although in this case $Ax = b \in \mathcal{R}(A)$ has a unique solution, the left inverse is not unique. Find the general form of Σ^L and generalize the result to A^L.

(b) Discuss the **right inverse** A^R in a similar way.

7. Show that $A^\dagger$ minimizes $\|AX - I\|_F$.

8. Prove *Bjerhammar's characterization* : Let A have full column rank and let B be any matrix such that $A^T B = 0$ and $(\,A \quad B\,)$ is nonsingular. Then $A^\dagger = X^T$ where

$$\begin{pmatrix} X^T \\ Y^T \end{pmatrix} = (\,A \quad B\,)^{-1}.$$

8.2 The Method of Normal Equations

8.2.1 Characterization of Least Squares Solutions

We now show a necessary condition for a vector x to minimize $\|b - Ax\|_2$.

Theorem 8.2.1.

Given the matrix $A \in \mathbf{R}^{m\times n}$ and a vector $b \in \mathbf{R}^m$. The vector x minimizes $\|b - Ax\|_2$ if and only if the residual vector $r = b - Ax$ is orthogonal to $\mathcal{R}(A)$, or equivalently

$$A^T(b - Ax) = 0. \tag{8.2.1}$$

Proof. Let x be a vector for which $A^T(b - Ax) = 0$. Then for any $y \in \mathbf{R}^n$ $b - Ay = (b - Ax) + A(x - y)$. Squaring this and using (8.2.1) we obtain

$$\|b - Ay\|_2^2 = \|b - Ax\|_2^2 + \|A(x - y)\|_2^2 \geq \|b - Ax\|_2^2.$$

On the other hand assume that $A^T(b - Ax) = z \neq 0$. Then if $x - y = -\epsilon z$ we have for sufficiently small $\epsilon \neq 0$,

$$\|b - Ay\|_2^2 = \|b - Ax\|_2^2 - 2\epsilon\|z\|_2^2 + \epsilon^2\|Az\|_2^2 < \|b - Ax\|_2^2$$

so x does not minimize $\|b - Ax\|_2$. □

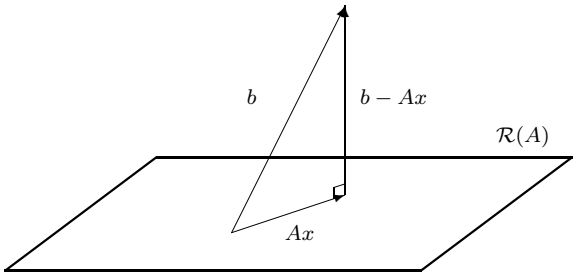

Figure 8.2.1. *Geometric characterization of the least squares solution.*

Here $A^TA \in \mathbf{R}^{n\times n}$ is a symmetric matrix and since

$$x^TA^TAx = \|Ax\|_2^2 \geq 0,$$

also positive semidefinite. The normal equations $A^TAx = A^Tb$ are *consistent* since

$$A^Tb \in \mathcal{R}(A^T) = \mathcal{R}(A^TA),$$

and therefore a least squares solution always exists.

By Theorem 8.2.1 any least squares solution x will decompose the right hand side b into two orthogonal components

$$b = Ax + r, \quad r \perp Ax. \tag{8.2.2}$$

Here $Ax = P_{\mathcal{R}(A)}b$ is the orthogonal projection (see Sec. 8.3.1) onto $\mathcal{R}(A)$ and $r \in \mathcal{N}(A^T)$ (cf. Fig. 8.2.1). Any solution to the (always consistent) normal equations (8.2.1) is a least squares solution. Note that although the least squares solution x may not be unique the decomposition in (8.2.2) always is unique.

Theorem 8.2.2.

The matrix A^TA is positive definite if and only if the columns of A are linearly independent, i.e., when $\operatorname{rank}(A) = n$. *In this case the least squares solution x is unique and given by*

$$x = (A^TA)^{-1}A^Tb. \tag{8.2.3}$$

Proof. If the columns of A are linearly independent, then $x \neq 0 \Rightarrow Ax \neq 0$. Therefore $x \neq 0 \Rightarrow x^TA^TAx = \|Ax\|_2^2 > 0$, and hence A^TA is positive definite. On

the other hand, if the columns are linearly dependent, then for some $x_0 \neq 0$ we have $Ax_0 = 0$. Then $x_0^T A^T A x_0 = 0$, and therefore $A^T A$ is not positive definite. When $A^T A$ is positive definite it is also nonsingular and (8.2.3) follows. □

For the minimum norm problem (8.1.2) let y be any solution of $A^T y = c$, and write $y = y_1 + y_2$, where $y_1 \in \mathcal{R}(A)$. $y_2 \in \mathcal{N}(A^T)$. Then $A^T y_2 = 0$ and hence y_1 is also a solution. Since $y_1 \perp y_2$ we have

$$\|y_1\|_2^2 = \|y\|_2^2 - \|y_2\|_2^2 \leq \|y\|_2^2,$$

with equality only if $y_2 = 0$. Hence the minimum norm solution lies in $\mathcal{R}(A)$ and we can write $y = Az$, for some z. Then we have $A^T y = A^T A z = c$. If A^T has linearly independent rows the inverse of $A^T A$ exists and the minimum norm solution $y \in \mathbb{R}^m$ satisfies the normal equations of second kind

$$y = A(A^T A)^{-1} c. \tag{8.2.4}$$

8.2.2 Forming and Solving the Normal Equations

From the time of Gauss until the computer age the basic computational tool for solving (8.1.1) was to form $A^T A$ and $A^T b$ and solve the normal equations by symmetric Gaussian elimination (which Gauss did), or later by the Cholesky factorization [7]. We now discuss the numerical implementation of this method. We defer treatment of rank deficient problems to later and assume throughout this section that rank $(A) = n$.

The first step is to compute the elements of the symmetric matrix $C = A^T A$ and the vector $d = A^T b$. If $A = (a_1, a_2, \ldots, a_n)$ has been partitioned by columns, we can use the inner product formulation

$$c_{jk} = (A^T A)_{jk} = a_j^T a_k, \qquad d_j = (A^T b)_j = a_j^T b, \quad 1 \leq j \leq k \leq n. \tag{8.2.5}$$

Since C is symmetric it is only necessary to compute and store its lower (or upper) triangular which requires $\frac{1}{2}mn(n+1)$ multiplications. Note that if $m \gg n$, then the number of elements $\frac{1}{2}n(n+1)$ in the upper triangular part of $A^T A$ is much smaller than the number mn of elements in A. Hence in this case the formation of $A^T A$ and $A^T b$ can be viewed as a *data compression*!

The formulas in (8.2.5) may not be suitable for large problems, where the matrix A is held in secondary storage, since each column needs to be accessed many times. An alternative row oriented outer product algorithm only needs *one pass* through the data (A, b). Denoting by $\tilde{a}_i^T$, the ith row of A, $i = 1, \ldots, m$, we have

$$C = A^T A = \sum_{i=1}^{m} \tilde{a}_i \tilde{a}_i^T, \qquad d = A^T b = \sum_{i=1}^{m} b_i \tilde{a}_i. \tag{8.2.6}$$

This is an form, where $A^T A$ is expressed as the sum of m matrices of rank one and $A^T b$ as a linear combination of the transposed rows of A. Using this alternative no

more storage is needed than that for A^TA and A^Tb. This outer product form is also preferable if the matrix A is sparse; see the hint to Problem 7.6.1. Note that both formulas can be combined if we adjoin b to A and form

$$(A,b)^T(A,b) = \begin{pmatrix} A^TA & A^Tb \\ b^TA & b^Tb \end{pmatrix}.$$

The matrix $C = A^TA$ is symmetric, and if $\operatorname{rank}(A) = n$ also positive definite. Gauss solved the normal equations by symmetric Gaussian elimination, but computing the Cholesky factorization

$$C = A^TA = R^TR, \quad R \in \mathbf{R}^{n\times n}, \tag{8.2.7}$$

is now the standard approach. The Cholesky factor R is upper triangular and nonsingular and can be computed by one of the algorithms given in Sec. 7.4.2. The least squares solution is then obtained by solving the two triangular systems

$$R^Tz = d, \qquad Rx = z. \tag{8.2.8}$$

Forming and solving the normal equations requires (neglecting lower order terms) about $\frac{1}{2}mn^2 + \frac{1}{6}n^3$ flops. If we have several right hand sides b_i, $i = 1:p$, then the Cholesky factorization need only be computed once. To solve for each new right hand side then only needs $mn + n^2$ additional flops.

Example 8.2.1.

Linear regression is the problem of fitting a linear model $y = \alpha + \beta x$ to a set of given points (x_i, y_i), $i = 1:m$. This leads to a overdetermined linear system

$$\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

Forming the normal equations we get

$$\begin{pmatrix} m & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m y_i \\ \sum_{i=1}^m y_ix_i \end{pmatrix}. \tag{8.2.9}$$

Eliminating α we obtain the "classical" formulas

$$\beta = \Big(\sum_{i=1}^m y_ix_i - m\bar{y}\bar{x}\Big) \Big/ \Big(\sum_{i=1}^m x_i^2 - m\bar{x}^2\Big),$$

where

$$\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i, \qquad \bar{x} = \frac{1}{m}\sum_{i=1}^m x_i. \tag{8.2.10}$$

are the mean values. The first equation in (8.2.9) gives

$$\bar{y} = \alpha + \beta\bar{x}. \tag{8.2.11}$$

which shows that $(\bar{y}, \bar{x})$ lies on the fitted line. This determines $\alpha = \bar{y} - \beta\bar{x}$.

A more accurate formula for β is obtained by first subtracting out the mean values from the data. We have

$$(y - \bar{y}) = \beta(x - \bar{x})$$

In the new variables the matrix of normal equation is *diagonal.* and we find

$$\beta = \textstyle\sum_{i=1}^{m}(y_i - \bar{y})(x_i - \bar{x})_i \Big/ \sum_{i=1}^{m}(x_i - \bar{x})^2. \tag{8.2.12}$$

A drawback of this formula is that it requires two passes through the data.

In many least squares problems the matrix A has the property that in each row all nonzero elements in A are contained in a narrow band. For banded rectangular matrix A we define:

Definition 8.2.3.

For $A \in \mathbf{R}^{m\times n}$ let f_i and l_i be the column subscripts of the first and last nonzero in the ith row of A, i.e.,

$$f_i = \min\{j \mid a_{ij} \neq 0\}, \qquad l_i = \max\{j \mid a_{ij} \neq 0\}. \tag{8.2.13}$$

Then the matrix A is said to have row bandwidth w, where

$$w = \max_{1\le i\le m} w_i, \qquad w_i = (l_i - f_i + 1). \tag{8.2.14}$$

Alternatively w is the smallest number for which it holds that

$$a_{ij}a_{ik} = 0, \quad \textit{if} \quad |j - k| \geq w. \tag{8.2.15}$$

For this structure to have practical significance we need to have $w \ll n$. Matrices of small row bandwidth often occur naturally, since they correspond to a situation where only variables "close" to each other are coupled by observations. We now prove a relation between the row bandwidth of the matrix A and the bandwidth of the corresponding matrix of normal equations A^TA.

Theorem 8.2.4.

Assume that the matrix $A \in \mathbf{R}^{m\times n}$ has row bandwidth w. Then the symmetric matrix A^TA has bandwidth $r \leq w - 1$.

Proof. From the Definition 8.2.3 it follows that $a_{ij}a_{ik} \neq 0 \Rightarrow |j - k| < w$. Hence,

$$|j - k| \geq w \Rightarrow (A^TA)_{jk} = \sum_{i=1}^{m} a_{ij}a_{ik} = 0.$$

□

If the matrix A also has full column rank it follows that we can use the band Cholesky Algorithm 6.4.6 to solve the normal equations.

The covariance matrix estimate in (8.1.4) can be expressed in terms of the Cholesky factor as

$$V = \sigma^2(A^TA)^{-1} = \sigma^2(R^TR)^{-1} = \sigma^2R^{-1}R^{-T}.$$

In order to assess the accuracy of the computed least squares estimate of x it is often required to compute the matrix V, or part of it. The matrix $S = R^{-1}$, which is also upper triangular, can be computed from the triangular system $RS = I$ by back-substitution. Often just the diagonal elements v_{ii} of $V = \sigma^2SS^T$ are required, which are the variances of the components of the least squares solution x. These elements are the 2-norms squared of the rows of S,

$$v_{ii} = \sigma^2 \sum_{j=i}^{n} s_{ij}^2, \quad i = 1, 2, \ldots, n.$$

In many situations the matrix V only occurs as an intermediate quantity in a formula. For example the variance of a linear functional $\varphi = f^T\hat{x}$ is equal to $\sigma^2 v$, where

$$v = f^TVf = f^TR^{-1}R^{-T}f = z^Tz, \quad z = R^{-T}f.$$

Thus to compute v we only need to solve the triangular system $R^Tz = f$ and form z^Tz. This is a more stable and efficient approach than using the expression f^TVf.

We have $r - \hat{r} = -A(A^TA)^{-1}A^T\epsilon$, where $\hat{r} = b - A\hat{x}$ is the least squares residual and ϵ the random error in the model. Hence $r - \hat{r}$ has covariance matrix

$$V_r = \sigma^2(A(A^TA)^{-1}A^T)^2 = \sigma^2A(A^TA)^{-1}A^T = \sigma^2P_{\mathcal{R}(A)}.$$

Note that the orthogonal projector $P_{\mathcal{R}(A)}$ can be computed from

$$P_{\mathcal{R}(A)} = A(R^TR)^{-1}A^T = QQ^T, \qquad Q = AR^{-1}.$$

The **normalized residuals** are defined by

$$\tilde{r} = (\operatorname{diag}(V_r))^{-1/2}\hat{r}.$$

Large components in $\tilde{r}$ can be assumed to correspond to "bad" data.

8.2.3 Stability and Accuracy with Normal Equations

We now turn to a discussion of the accuracy of the method of normal equations for least squares problems. First we consider rounding errors in the formation of the system of normal equations. Using the standard model for floating point computation we get for the elements $\bar{c}_{ij}$ in the computed matrix $\bar{C} = fl(A^TA)$

$$\bar{c}_{ij} = fl\Big(\sum_{k=1}^{m} a_{ik}a_{jk}\Big) = \sum_{k=1}^{m} a_{ik}a_{jk}(1 + \delta_k),$$

where (see (2.4.4)) $|\delta_k| < 1.06(m+2-k)u$ (u is the machine unit). It follows that the computed matrix satisfies

$$\bar{C} = A^T A + E, \qquad |e_{ij}| < 1.06um \sum_{k=1}^{m} |a_{ik}||a_{jk}|. \tag{8.2.16}$$

A similar estimate holds for the rounding errors in the computed vector $A^T b$. Note that it is *not* possible to show that $\bar{C} = (A+E)^T(A+E)$ for some small error matrix E, i.e., the rounding errors in forming the matrix $A^T A$ are not in general equivalent to small perturbations of the initial data matrix A. From this we can deduce that *the method of normal equations is not backwards stable.* The following example illustrates that when $A^T A$ is ill-conditioned, *it might be necessary to use double precision in forming and solving the normal equations in order to avoid loss of significant information.*

Example 8.2.2. (LÄUCHLI) Consider the system $Ax = b$, where

$$A = \begin{pmatrix} 1 & 1 & 1 \\ \epsilon & & \\ & \epsilon & \\ & & \epsilon \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad |\epsilon| \ll 1.$$

We have, exactly

$$A^T A = \begin{pmatrix} 1+\epsilon^2 & 1 & 1 \\ 1 & 1+\epsilon^2 & 1 \\ 1 & 1 & 1+\epsilon^2 \end{pmatrix}, \qquad A^T b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

$$x = \frac{1}{3+\epsilon^2}\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T, \qquad r = \frac{1}{3+\epsilon^2}\begin{pmatrix} \epsilon^2 & -1 & -1 & -1 \end{pmatrix}^T.$$

Now assume that $\epsilon = 10^{-4}$, and that we use eight-digit decimal floating point arithmetic. Then $1+\epsilon^2 = 1.00000001$ rounds to 1, and the computed matrix $A^T A$ will be singular. We have lost all information contained in the last three rows of A! Note that the residual in the first equation is $O(\epsilon^2)$ but $O(1)$ in the others.

Least squares problems of this form occur when the error in some equations (here $x_1 + x_2 + x_3 = 1$) have a much smaller variance than in the others; see Sec. 8.6.2.

To assess the error in the least squares solution $\bar{x}$ computed by the method of normal equations, we must also account for rounding errors in the Cholesky factorization and in solving the triangular systems. Using Theorem 6.6.6 and the perturbation bound in Theorem 6.6.2 it can be shown that provided that $2n^{3/2}u\kappa(A^T A) < 0.1$, the error in the computed solution $\bar{x}$ satisfies

$$\|\bar{x} - x\|_2 \leq 2.5n^{3/2}u\kappa(A^T A)\|x\|_2. \tag{8.2.17}$$

As seen in Sec. 8.1.5, for "small" residual least squares problem the true condition number is approximately $\kappa(A) = \kappa^{1/2}(A^T A)$. In this case *the system of normal*

equations can be much worse conditioned than the least squares problem from which it originated.

Sometimes ill-conditioning is caused by an unsuitable formulation of the problem. Then a different choice of parameterization can significantly reduce the condition number. For example, in approximation problems one should try to use orthogonal, or nearly orthogonal, base functions. In case the elements in A and b are the original data the ill-conditioning cannot be avoided in this way.

In statistics the linear least squares problem $\min_x \|b - Ax\|_2$ derives from a **multiple linear regression** problem, where the vector b is a response variable and the columns of A contain the values of the explanatory variables.

In Secs. 8.3 and 8.4 we consider methods for solving least squares problems based on orthogonalization. These methods work directly with A and b and are backwards stable.

8.2.4 Scaling Least Squares Problems

In Sec. 7.7.7 we discussed how the scaling of rows and columns of a linear system $Ax = b$ influenced the solution computed by Gaussian elimination. For a least squares problem $\min_x \|Ax - b\|_2$ a row scaling of (A, b) is not allowed since such a scaling would change the exact solution. However, we can scale the columns of A. If we take $x = Dx'$, the normal equations will change into

$$(AD)^T(AD)x' = D(A^TA)Dx' = DA^Tb.$$

Hence this corresponds to a *symmetric scaling* of rows and columns in A^TA. It is important to note that if the Cholesky algorithm is carried out without pivoting the computed solution is *not* affected by such a scaling, cf. Theorem 7.5.6. This means that even if no explicit scaling is carried out, the rounding error estimate (8.2.17) for the computed solution $\bar{x}$ holds for *all* D,

$$\|D(\bar{x} - x)\|_2 \leq 2.5n^{3/2}u\kappa(DA^TAD)\|Dx\|_2.$$

(Note, however, that scaling the columns changes the norm in which the error in x is measured.)

Denote the *minimum* condition number under a symmetric scaling with a positive diagonal matrix by

$$\kappa'(A^TA) = \min_{D>0} \kappa(DA^TAD). \tag{8.2.18}$$

The following result by van der Sluis [1969] shows the scaling where D is chosen so that in $D(A^TA)D$ all column norms are equal, i.e. $D = \text{diag}(\|a_1\|_2, \ldots, \|a_n\|_2)^{-1}$, comes within a factor of n of the minimum value.

Theorem 8.2.5. *Let $C \in \mathbf{R}^{n\times n}$ be a symmetric and positive definite matrix, and denote by $\mathcal{D}$ the set of $n\times n$ nonsingular diagonal matrices. Then if in C all diagonal elements are equal, and C has at most q nonzero elements in any row, it holds that*

$$\kappa(C) \leq q \min_{D\in\mathcal{D}} \kappa(DCD).$$

As the following example shows, this scaling can reduce the condition number considerably. In cases where the method of normal equations gives surprisingly accurate solution to a seemingly very ill-conditioned problem, the explanation often is that the condition number of the scaled problem is quite small!

Example 8.2.3. The matrix $A \in R^{21\times 6}$ with elements

$$a_{ij} = (i-1)^{j-1}, \qquad 1 \le i \le 21, \quad 1 \le j \le 6$$

arises when fitting a fifth degree polynomial $p(t) = x_0 + x_1 t + x_2 t^2 + \ldots + x_5 t^5$ to observations at points $x_i = 0, 1, \ldots, 20$. The condition numbers are

$$\kappa(A^T A) = 4.10 \cdot 10^{13}, \qquad \kappa(DA^T AD) = 4.93 \cdot 10^{6}.$$

where D is the column scaling in Theorem 8.2.5. Thus, the condition number of the matrix of normal equations is reduced by about seven orders of magnitude by this scaling!

A simple way to improve the accuracy of a solution $\bar{x}$ computed by the method of normal equations is by fixed precision iterative refinement, see Sec. 7.7.8. This requires that the data matrix A is saved and used to compute the residual vector $b - A\bar{x}$. In this way information lost when $A^T A$ was formed can be recovered. If also the corrections are computed from the normal equations we obtain the following algorithm:

Iterative Refinement with Normal Equations:

Set $x_1 = \bar{x}$, and for $s = 1, 2, \ldots$ until convergence do

$$\begin{aligned} r_s &:= b - Ax_s, \qquad R^T R \delta x_s = A^T r_s, \\ x_{s+1} &:= x_s + \delta x_s. \end{aligned}$$

Here R is computed by Cholesky factorization of the matrix of normal equation $A^T A$. This algorithm only requires one matrix-vector multiplication each with A and A^T and the solution of two triangular systems. Note that the first step, i.e., for $i = 0$, is identical to solving the normal equations. It can be shown that initially the errors will be reduced with rate of convergence equal to

$$\bar{\rho} = cu\kappa'(A^T A), \tag{8.2.19}$$

where c is a constant depending on the dimensions m, n. Several steps of the refinement may be needed to get good accuracy. (Note that $\bar{\rho}$ is proportional to $\kappa'(A^T A)$ even when no scaling of the normal equations has been performed!)

Example 8.2.4. If $\kappa'(A^T A) = \kappa(A^T A)$ and $c \approx 1$ the error will be reduced to a backward stable level in p steps if $\kappa^{1/2}(A^T A) \le u^{-p/(2p+1)}$. (As remarked before $\kappa^{1/2}(A^T A)$ is the condition number for a small residual problem.) For example, with $u = 10^{-16}$, the maximum value of $\kappa^{1/2}(A^T A)$ for different values of p are:

$$10^{5.3}, 10^{6.4}, 10^{8}, \qquad p = 1, 2, \infty.$$

For moderately ill-conditioned problems the normal equations combined with iterative refinement can give very good accuracy. For more ill-conditioned problems the methods based QR factorization described in Secs. 8.3 and 8.4 are usually to be preferred.

8.2.5 Methods Based on Gaussian Elimination

The pseudo-inverse of a matrix can also be computed using a LU factorization with complete pivoting. Usually it will be sufficient to use partial pivoting with a linear independence check. Let $\tilde{a}_{q,p+1}$ be the element of largest magnitude in column $p+1$. If $|\tilde{a}_{q,p+1}| < tol$, column $p+1$ is considered to be linearly dependent and is placed last. We then look for a pivot element in column $p+2$, etc.

Assume now that we have computed the LU factorization

$$\Pi_1 A \Pi_2 = \begin{pmatrix} L_{11} \\ L_{21} \end{pmatrix} (U_{11} \quad U_{12}), \tag{8.2.20}$$

where $L_{11}, U_{11} \in \mathbf{R}^{r\times r}$ are triangular and nonsingular. Then by Theorem 8.1.7 we have

$$\begin{aligned} A^\dagger &= \Pi_2 (U_{11} \quad U_{12})^\dagger \begin{pmatrix} L_{11} \\ L_{21} \end{pmatrix}^\dagger \Pi_1 \\ &= \Pi_2 (I_r \quad S)^\dagger U_{11}^{-1} L_{11}^{-1} \begin{pmatrix} I_r \\ T \end{pmatrix}^\dagger \Pi_1, \end{aligned}$$

where

$$T = L_{21} L_{11}^{-1}, \qquad S = U_{11}^{-1} U_{12},$$

Note the symmetry in the treatment of the L and U factors!

Standard algorithms for solving nonsymmetric linear systems $Ax = b$ are usually based on LU factorization with partial pivoting. Therefore it seems natural to consider such factorizations also for least squares problems which are only slightly overdetermined, i.e., where $m - n \ll n$.

A rectangular matrix $A \in \mathbf{R}^{m\times n}$, $m \geq n$, can be reduced by Gaussian elimination with partial pivoting to an upper triangular form U. In general, column interchanges are needed to ensure numerical stability. In the full rank case, $\operatorname{rank}(A) = n$, the resulting LDU factorization becomes

$$\Pi_1 A \Pi_2 = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = LDU = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix} DU, \tag{8.2.21}$$

where $L_1 \in \mathbf{R}^{n\times n}$ is unit lower triangular, D diagonal, and $U \in \mathbf{R}^{n\times n}$ is unit upper triangular and nonsingular. Thus the matrix L has the same dimensions as A and a lower trapezoidal structure. Computing this factorization requires $\frac{1}{2}n^2(m - \frac{1}{3}n)$ flops.

Using the LU factorization (8.2.21) and setting $\tilde{x} = \Pi_2^T x$, $\tilde{b} = \Pi_1 b$, the least squares problem $\min_x \|Ax - b\|_2$ is reduced to

$$\min_y \|Ly - \tilde{b}\|_2, \qquad DU\tilde{x} = y. \tag{8.2.22}$$

If partial pivoting by rows is used in the factorization (8.2.21), then L is usually a well-conditioned matrix. In this case the solution to the least squares problem (8.2.22) can be computed from the normal equations

$$L^T L y = L^T \tilde{b},$$

without substantial loss of accuracy. This is the approach taken by Peters and Wilkinson [49, 1970]. The following example shows that this is a more stable method than using the normal equation $A^T Ax = A^T b$.

Example 8.2.5. (Noble [43, 1976])
Consider the matrix A and its pseudo-inverse

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1+\epsilon^{-1} \\ 1 & 1-\epsilon^{-1} \end{pmatrix}, \quad A^\dagger = \frac{1}{6}\begin{pmatrix} 2 & 2-3\epsilon^{-1} & 2+3\epsilon^{-1} \\ 0 & 3\epsilon^{-1} & -3\epsilon^{-1} \end{pmatrix}.$$

The (exact) matrix of normal equations is

$$A^T A = \begin{pmatrix} 3 & 3 \\ 3 & 3+2\epsilon^2 \end{pmatrix}.$$

If $\epsilon \leq \sqrt{u}$, then in floating point computation $fl(3+2\epsilon^2) = 3$, and the computed matrix $fl(A^T A)$ has rank one. However, the LU factorization of A is

$$A = LDU = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},$$

where L and U are well-conditioned. The correct pseudo-inverse is now obtained from

$$A^\dagger = U^{-1}D^{-1}(L^T L)^{-1}L^T = \begin{pmatrix} 1 & -\epsilon \\ 0 & \epsilon \end{pmatrix} \begin{pmatrix} 1/3 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix}.$$

and now there is no cancellation. □

Forming the symmetric matrix $L^T L$ requires $\frac{1}{2}n^2(m - \frac{2}{3}n)$ flops, and computing its Cholesky factorization takes $n^3/6$ flops. Hence, neglecting terms of order n^2, the total number of flops to compute the least squares solution by the Peters–Wilkinson method is $n^2(m - \frac{1}{3}n)$. This is always more expensive than the method of normal equations applied to $A^T A$.

When $m - n < n$ an algebraic reformulation is advantageous. If we let $T = L_2 L_1^{-1}$ and $L_1 y = z$, problem (8.2.22) becomes

$$\min_z \left\| \begin{pmatrix} I_n \\ T \end{pmatrix} z - \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix} \right\|_2 .$$

The solution z can be computed from

$$\begin{aligned} z &= (I_n + T^T T)^{-1}(\tilde{b}_1 + T^T \tilde{b}_2) \\ &= \tilde{b}_1 + (I_n + T^T T)^{-1} T^T (\tilde{b}_2 - T\tilde{b}_1) \\ &= \tilde{b}_1 + T^T (I_{m-n} + TT^T)^{-1}(\tilde{b}_2 - T\tilde{b}_1). \end{aligned} \tag{8.2.23}$$

The last expression can be evaluated more efficiently if $m - n < n$ and leads to the most efficient method for solving slightly overdetermined least squares problems. (Note that for $m = n + 1$ the inversion in (8.2.23) is reduced to a scalar division.)

Methods based on the factorization (8.2.21) for solving the minimum norm problem $\min \|y\|_2$, subject to $A^T y = c$ can be similarly developed. Setting $\tilde{c} = \Pi_2^T c$ and $\tilde{y} = \Pi_1 y$, we have

$$\tilde{y} = (U^T L^T)^\dagger \tilde{c} = L(L^T L)^{-1} U^{-T} \tilde{c}.$$

For the case $m - n < n$ we note that from $U^T L^T \tilde{y} = \tilde{c}$ we have

$$\tilde{y}_1 = L_1^{-T} U^{-T} \tilde{c} - (L_2 L_1^{-1})^T \tilde{y}_2 = e - T^T \tilde{y}_2. \tag{8.2.24}$$

Hence $\tilde{y}_2$ can be obtained as the solution to the least squares problem

$$\min_{\tilde{y}_2} \left\| \begin{pmatrix} T^T \\ I_{m-n} \end{pmatrix} \tilde{y}_2 - \begin{pmatrix} e \\ 0 \end{pmatrix} \right\|_2 ,$$

or using the normal equations,

$$\tilde{y}_2 = (I_{m-n} + TT^T)^{-1} Te. \tag{8.2.25}$$

The reformulation used above for the almost square case follows from a useful identity, which holds for any matrix S of dimension $r \times (n - r)$ of rank r:

$$(I_r + S^T S)^{-1} S^T = S^T (I_{n-r} + SS^T)^{-1}. \tag{8.2.26}$$

This identity is easily proved using the Woodbury formula (3.1.6). It reduces the computation of the pseudo-inverse of a matrix of rank r to the computation of the pseudo-inverse of a matrix of rank $(n - r)$. If $n - r \ll r$, there is a great gain in efficiency.

Review Questions

1. Give a necessary and sufficient condition for x to be a solution to $\min_x \|Ax - b\|_2$, and interpret this geometrically. When is is the least squares solution x unique? When is $r = b - Ax$ unique?

2. What are the advantages and drawbacks with the method of normal equations for computing the least squares solution of $Ax = b$? Give a simple example, which shows that loss of information can occur in forming the normal equations.

3. Discuss how the accuracy of the method of normal equations can be improved by (a) scaling the columns of A, (b) iterative refinement.

4. Show that the more accurate formula in Example 8.2.1 can be interpreted as a special case of the method (8.5.5)–(8.5.6) for partitioned least squares problems.

5. (a) Let $A \in \mathbf{R}^{m\times n}$ with $m < n$. Show that A^TA is singular.
(b) Show, using the SVD, that $\operatorname{rank}(A^TA) = \operatorname{rank}(AA^T) = \operatorname{rank}(A)$.

6. Define the condition number $\kappa(A)$ of a rectangular matrix A. What terms in the perturbation of a least squares solution depend on κ and κ^2, respectively?

Problems

1. In order to estimate the height above sea level for three points, A,B, and C, the difference in altitude was measured between these points and points D,E, and F at sea level. The measurements obtained form a linear system in the heights x_A, x_B, and x_C of A,B, and C,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 1 \end{pmatrix}.$$

Show that the least squares solution and residual vector are

$$x = \frac{1}{4}(5, 7, 12)^T, \qquad r = \frac{1}{4}(-1, 1, 0, 2, 3, -3)^T.$$

and verify that the residual vector is orthogonal to all columns in A.

2. (a) Consider the linear regression problem of fitting $y(t) = \alpha + \beta(t - c)$ by the method of least squares to the data

t	1	3	4	6	7
$f(t)$	-2.1	-0.9	-0.6	0.6	0.9

With the (unsuitable) choice $c = 1,000$ the normal equations

$$\begin{pmatrix} 5 & 4979 \\ 4979 & 4958111 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} -2.1 \\ -2097.3 \end{pmatrix}$$

become very ill-conditioned. Show that if the element 4958111 is rounded to $4958 \cdot 10^3$ then β is perturbed from its correct value 0.5053 to -0.1306!

(b) As shown in Example 8.2.1, a much better choice of base functions is shifting with the mean value of t, i.e., taking $c = 4.2$. However, it is not necessary to shift with the *exact* mean; Show that shifting with 4, the midpoint of the interval $(1, 7)$, leads to a very well-conditioned system of normal equations.

3. Denote by x_w the solution to the weighted least squares problem (8.1.6) and let x be the solution to the corresponding unweighted problem ($W = I$). Using the normal equations show that

$$x_w - x = (A^T W^{-1} A)^{-1} A^T (W^{-1} - I)(b - Ax). \tag{8.2.27}$$

Conclude that weighting the rows affects the solution if $b \notin \mathcal{R}(A)$.

4. Assume that $\operatorname{rank}(A) = n$, and put $\bar{A} = (A, b) \in \mathbf{R}^{m\times(n+1)}$. Let the corresponding cross product matrix, and its Cholesky factor be

$$\bar{C} = \bar{A}^T \bar{A} = \begin{pmatrix} C & d \\ d^T & b^T b \end{pmatrix}, \qquad \bar{R} = \begin{pmatrix} R & z \\ 0 & \rho \end{pmatrix}.$$

Show that the solution x and the residual norm ρ to the linear least squares problem $\min_x \|b - Ax\|_2$ is given by

$$Rx = z, \qquad \|b - Ax\|_2 = \rho.$$

5. Let $A \in \mathbf{R}^{m\times n}$ and $\operatorname{rank}(A) = n$. Show that the minimum norm solution of the underdetermined system $A^T y = c$ can be computed as follows:

(i) Form the matrix $A^T A$, and compute its Cholesky factorization $A^T A = R^T R$.

(ii) Solve the two triangular systems $R^T z = c$, $Rx = z$, and compute $y = Ax$.

6. Compute the solution x using the LDU factorization in Example 8.6.2. Compare with the exact solution given in Example 8.2.2.

7. (B. Noble 1976) Consider the matrix A and its generalized inverse

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 + \epsilon^{-1} \\ 1 & 1 - \epsilon^{-1} \end{pmatrix}.$$

(a) Show that The (exact) matrix of normal equations is

$$A^T A = \begin{pmatrix} 3 & 3 \\ 3 & 3 + 2\epsilon^2 \end{pmatrix}.$$

Hence if $\epsilon \leq \sqrt{u}$, then in floating point computation $fl(3 + 2\epsilon^2) = 3$, and the computed matrix $fl(A^T A)$ has rank one.

(b) An LU factorization of A is

$$A = LU = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & \epsilon \end{pmatrix}.$$

Show that here L is well-conditioned. and that the pseudo-inverse can be stably computed from $A^\dagger = U^{-1}(L^T L)^{-1} L^T$.

8. (S. M. Stiegler [60].) In 1793 the French decided to base the new metric system upon a unit, the meter, equal to one 10,000,000th part of the distance from the the north pole to the equator along a meridian arc through Paris. The following famous data obtained in a 1795 survey consist of four measured subsections of an arc from Dunkirk to Barcelona. For each subsection the length of the arc S (in modules), the degrees d of latitude and the latitude L of the midpoint (determined by the astronomical observations) are given.

Segment	Arc length S	latitude d	Midpoint L
Dunkirk to Pantheon	62472.59	2.18910°	49° 56′ 30″
Pantheon to Evaux	76145.74	2.66868°	47° 30′ 46″
Evaux to Carcassone	84424.55	2.96336°	44° 41′ 48″
Carcassone to Barcelona	52749.48	1.85266°	42° 17′ 20″

If the earth is ellipsoidal, then to a good approximation it holds

$$z + y\sin^2(L) = S/d,$$

where z and y are unknown parameters. The meridian quadrant then equals $M = 90(z + y/2)$ and the eccentricity is e is found from $1/e = 3(z/y + 1/2)$. Use least squares to determine z and y and then M and $1/e$.

9. Consider the least squares problem $\min_x \|Ax - b\|_2^2$, where A has full column rank. Partition the problem as

$$\min_{x_1,x_2} \left\| (A_1 \ A_2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - b \right\|_2^2.$$

By a geometric argument show that the solution can be obtained as follows. First compute x_2 as solution to the problem

$$\min_{x_2} \|P_{A_1}^{\perp}(A_2x_2 - b)\|_2^2,$$

where $P_{A_1}^{\perp} = I - P_{A_1}$ is the orthogonal projector onto $\mathcal{N}(A_1^T)$. Then compute x_2 as solution to the problem

$$\min_{x_1} \|A_1x_1 - (b - A_2x_2)\|_2^2.$$

10. Show that if $A, B \in \mathbf{R}^{m\times n}$ and rank $(B) \neq$ rank (A) then it is not possible to bound the difference between $A^{\dagger}$ and $B^{\dagger}$ in terms of the difference $B - A$. *Hint*: Use the following example. Let $\epsilon \neq 0$, $\sigma \neq 0$, take

$$A = \begin{pmatrix} \sigma & 0 \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} \sigma & \epsilon \\ \epsilon & 0 \end{pmatrix},$$

and show that $\|B - A\|_2 = \epsilon$, $\|B^{\dagger} - A^{\dagger}\|_2 > 1/\epsilon$.

11. Show that for any matrix A it holds

$$A^\dagger = \lim_{\mu\to 0}(A^TA + \mu^2 I)^{-1}A^T = \lim_{\mu\to 0} A^T(AA^T + \mu^2 I)^{-1}. \tag{8.2.28}$$

12. (a) Let $A = (a_1, a_2)$, where $a_1^T a_2 = \cos\gamma$, $\|a_1\|_2 = \|a_2\|_2 = 1$. Hence γ is the angle between the vectors a_1 and a_2. Determine the singular values and right singular vectors v_1, v_2 of A by solving the eigenvalue problem for

$$A^TA = \begin{pmatrix} 1 & \cos\gamma \\ \cos\gamma & 1 \end{pmatrix}.$$

Then determine the left singular vectors u_1, u_2 from (7.1.33).
(b) Show that if $\gamma \ll 1$, then $\sigma_1 \approx \sqrt{2}$ and $\sigma_2 \approx \gamma/\sqrt{2}$ and

$$u_1 \approx (a_1 + a_2)/2, \qquad u_2 \approx (a_1 - a_2)/\gamma.$$

8.3 Methods using Orthogonal Factorizations

Orthogonality plays a key role in least squares problems. By Theorem 8.2.2, in the full rank case, $\operatorname{rank}(A) = n$, the residual $r = b - Ax$ can be written

$$r = P_{\mathcal{N}(A^T)}b, \qquad P_{\mathcal{N}(A^T)} = I - A(A^TA)^{-1}A^T, \tag{8.3.1}$$

which gives an expression for $P_{\mathcal{R}(A)}$, the orthogonal projector onto $\mathcal{R}(A)$, the range space of A. It follows that any solution to the consistent linear system

$$Ax = P_{\mathcal{R}(A)}b \tag{8.3.2}$$

is a least squares solution. In the next section we survey the theory of orthogonal and oblique projection.

8.3.1 Orthogonal and Oblique Projections

Recall that two vectors v and w in $\mathbf{R}^n$ are said to be **orthogonal** if $(v, w) = 0$. A set of vectors $v_1, \ldots, v_k$ in $\mathbf{R}^n$ is called orthogonal with respect to the Euclidian inner product if

$$v_i^T v_j = 0, \quad i \neq j,$$

and **orthonormal** if also $v_i^T v_i = 1, i = 1 : k$. An orthogonal set of vectors is linearly independent. More generally, a collection of subspaces $S_1, \ldots, S_k$ of $\mathbf{R}^n$ are mutually orthogonal if

$$x^Ty = 0, \quad \forall\, x \in S_i, \quad \forall\, y \in S_j, \quad i \neq j.$$

The **orthogonal complement** $S^\perp$ of a subspace $S \in \mathbf{R}^n$ is defined by

$$S^\perp = \{y \in \mathbf{R}^n |\ x^Ty = 0,\ x \in S\}.$$

Let $q_1, \ldots, q_k$ form an orthonormal basis for a subspace $S \subset \mathbf{R}^n$. Such a basis can always be extended to a full orthonormal basis $q_1, \ldots, q_n$ for $\mathbf{R}^n$, and then $S^\perp = \operatorname{span}\{q_{k+1}, \ldots, q_n\}$.

Let $q_1, \ldots, q_n \in \mathbf{R}^m$ be orthonormal Then the matrix $Q = (q_1, \ldots, q_n) \in \mathbf{R}^{m\times n}$, $m \geq n$, is called an **orthogonal matrix** and $Q^TQ = I_n$. If Q is square $(m = n)$ then $Q^{-1} = Q^T$, and hence also $QQ^T = I_n$. Further,

$$1 = \det(Q^TQ) = \det(Q^T)\det(Q) = (\det(Q))^2,$$

and it follows that $\det(Q) = \pm 1$.

In the complex case, $A = (a_{ij}) \in \mathbf{C}^{m\times n}$ the Hermitian inner product leads to modifications in the definition of symmetric and orthogonal matrices. Two vectors x and y in $\mathbf{C}^n$ are called orthogonal if $x^Hy = 0$. A square matrix U for which $U^HU = I$ is called **unitary**. Then

$$(Ux)^HUy = x^HU^HUy = x^Hy,$$

and hence unitary matrices have the property that they preserve the Hermitian inner product. In particular the Euclidian length of a vector is invariant under unitary transformations, i.e., $\|Ux\|_2^2 = \|x\|_2^2$. Note that when the vectors and matrices are real the definitions for the complex case are consistent with those made for the real case.

Any square matrix $P \in \mathbf{R}^{n\times n}$ such that

$$P^2 = P. \tag{8.3.3}$$

is called **idempotent** and a **projector**. An arbitrary vector $v \in \mathbf{R}^n$ can be decomposed in a unique way as

$$v = Pv + (I - P)v = v_1 + v_2. \tag{8.3.4}$$

Here $v_1 = Pv \in S$ is a projection of v onto $\mathcal{R}(P)$, the range space of P. Since $Pv_2 = (P - P^2)v = 0$ it follows that $(I - P)$ is a projection onto $\mathcal{N}(P)$, the null space of P.

If P is symmetric, $P^T = P$, then

$$v_1^Tv_2 = (Pv)^T(I - P)v = v^TP(I - P)v = v^T(P - P^2)v = 0.$$

It follows that $v_2 \perp S$, i.e., v_2 lies in the orthogonal complement $S^\perp$ of S; In this case P is the **orthogonal projector** onto S and $I - P$ the orthogonal projector onto $S^\perp$. It can be shown that the orthogonal projector P onto a given subspace S is unique, see Problem 1.

Example 8.3.1.

Let $Q = (q_1, \ldots, q_n) \in \mathbf{R}^{m\times n}$, $m \geq n$, where $q_1, \ldots, q_n \in \mathbf{R}^m$ are orthonormal vectors. Then the orthogonal projector onto the orthogonal complement of $\mathcal{R}(Q)$.

$$P = I_m - QQ^T, \tag{8.3.5}$$

If $n = 1$ then $P = I_m - q_1 q_1^T$ and the null space $\mathcal{N}(P) = \text{span}\,(q_1)$ has dimension one. P is then called an **elementary orthogonal projection**.

A projector P such that $P \neq P^T$ is called an **oblique projector**. We now briefly review oblique projections and their matrix representations. If λ is an eigenvalue of P then from $P^2 = P$ it follows that $\lambda^2 = \lambda$. Hence the eigenvalues of P are either 1 or 0 and we can write the eigendecomposition

$$P = (U_1\, U_2) \begin{pmatrix} I_k & 0 \\ 0 & 0_{n-k} \end{pmatrix} \begin{pmatrix} \hat{Y}_1^T \\ \hat{Y}_2^T \end{pmatrix}, \quad \begin{pmatrix} \hat{Y}_1^T \\ \hat{Y}_2^T \end{pmatrix} = (U_1\, U_2)^{-1}, \tag{8.3.6}$$

where $k = \text{trace}\,(P)$ is the rank of P and

$$\text{span}\,(U_1) = \mathcal{R}(P), \qquad \text{span}\,(U_2) = \mathcal{N}(P).$$

The matrices $U_1 \in \mathbf{R}^{n\times n_1}$ and $U_2 \in \mathbf{R}^{n\times n_2}$ $(n_1 + n_2 = n)$, can be chosen as orthogonal bases for the invariant subspaces corresponding to the eigenvalues 1 or 0, respectively. In terms of this eigendecomposition (8.3.4) can be written

$$v = (U_1\ U_2) \begin{pmatrix} \hat{Y}_1^T \\ \hat{Y}_2^T \end{pmatrix} v = (U_1 \hat{Y}_1^T)v + (U_2 \hat{Y}_2^T)v = v_1 + v_2, \tag{8.3.7}$$

that is

$$P = U_1 \hat{Y}_1^T, \qquad I - P = U_2 \hat{Y}_2^T. \tag{8.3.8}$$

If $P^T = P$ then P is an orthogonal projector and in (8.3.6) we can take $U = (U_1\, U_2)$ orthogonal and $\hat{Y}_1 = U_1$ and $\hat{Y}_2 = U_2$. The projectors (8.3.7) then take the form

$$P = U_1 U_1^T, \qquad I - P = U_2 U_2^T; \tag{8.3.9}$$

For an orthogonal projector we have

$$\|Pv\|_2 = \|U_1^T v\|_2 \leq \|v\|_2 \ \ \forall \ \ v \in \mathbf{R}^m, \tag{8.3.10}$$

where equality holds for all vectors in $\mathcal{R}(U_1)$. From this it follows that for an orthogonal projector $\|P\|_2 = 1$. The conversion is also true; P is an orthogonal projection only if (8.3.10) holds.

When P is not symmetric we call $v_1 = Pv$ the **oblique projection** of v onto $\mathcal{R}(U_1)$ along $\mathcal{R}(U_2)$, and the matrix $P = U_1 \hat{Y}_1^T$ is the corresponding oblique projector. Similarly $I - P = U_2 \hat{Y}_2^T$ is the oblique projector onto $\mathcal{R}(U_2)$ along $\mathcal{R}(U_1)$.

From (8.3.6) we have

$$\begin{pmatrix} \hat{Y}_1^T \\ \hat{Y}_2^T \end{pmatrix} (U_1\ U_2) = \begin{pmatrix} \hat{Y}_1^T U_1 & \hat{Y}_1^T U_2 \\ \hat{Y}_2^T U_1 & \hat{Y}_2^T U_2 \end{pmatrix} = \begin{pmatrix} I_k & 0 \\ 0 & I_{n-k} \end{pmatrix}. \tag{8.3.11}$$

In particular we have $\hat{Y}_1^T U_2 = 0$ and $\hat{Y}_2^T U_1 = 0$. Hence the columns of $\hat{Y}_1$ form a basis of the orthogonal complement of $\mathcal{R}(U_2)$ and, similarly, the columns of $\hat{Y}_2$ form a basis of the orthogonal complement of $\mathcal{R}(U_1)$.

Let Y_1 be an orthogonal matrix whose columns span $\mathcal{R}(\hat{Y}_1)$. Then there is a nonsingular matrix G_1 such that $\hat{Y}_1 = Y_1 G$. From (8.3.11) it follows that $G^T Y_1^T U_1 = I_k$, and hence $G^T = (Y_1^T U_1)^{-1}$. Similarly $Y_2 = (Y_2^T U_2)^{-1} Y_2$ is an orthogonal matrix whose columns span $\mathcal{R}(\hat{Y}_2)$. Hence using (8.3.8) the projectors can be written

$$P = U_1 (Y_1^T U_1)^{-1} Y_1^T, \qquad I - P = U_2 (Y_2^T U_2)^{-1} Y_2^T. \tag{8.3.12}$$

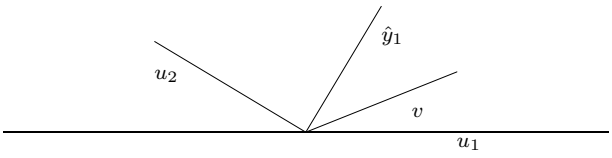

Figure 8.3.1. *The oblique projection of v on u_1 along u_2.*

Example 8.3.2.

We illustrate the case when $n = 2$ and $n_1 = 1$. Let the vectors u_1 and y_1 be normalized so that $\|u_1\|_2 = \|y_1\|_2 = 1$ and let $y_1^T u_1 = \cos\theta$, where θ is the angle between u_1 and y_1, see Fig. 8.4.1. Since

$$P = u_1 (y_1^T u_1)^{-1} y_1^T = \frac{1}{\cos\theta} u_1 y_1^T.$$

Hence $\|P\|_2 = 1/\cos\theta \geq 1$, and $\|P\|_2$ becomes very large when y_1 is almost orthogonal to u_1. When $y_1 = u_1$ we have $\theta = 0$ and P is an orthogonal projection.

8.3.2 Gram–Schmidt Orthogonalization

Gram–Schmidt orthogonalization is one of the fundamental algorithms in numerical linear algebra. Given a sequence of linearly independent vectors $a_1, a_2, \ldots, a_n$ Gram–Schmidt orthogonalization computes orthonormal vectors $q_1, q_2, \ldots, q_n$. such that

$$\text{span}\,[a_1, \ldots, a_k] = \text{span}\,[q_1, \ldots, q_k], \quad k = 1 : n. \tag{8.3.13}$$

Algorithm 8.3.1 Classical Gram–Schmidt (CGS).

for $k = 1 : n$

(i) If $k = 1$ then set $\hat{q}_1 = a_1$ else orthogonalize a_k against $q_1, \ldots, q_{k-1}$:

$$\hat{q}_k = a_k - \sum_{i=1}^{k-1} r_{ik} q_i, \qquad r_{ik} = q_i^T a_k, \quad i = 1 : k-1; \tag{8.3.14}$$

(ii) Normalize $\hat{q}_k$

$$r_{kk} = \|\hat{q}_k\|_2, \qquad q_k = \hat{q}_k / r_{kk}. \tag{8.3.15}$$

end;

Note that $\hat{q}_k \neq 0$, since otherwise a_k is a linear combination of the vectors $a_1, \ldots, a_{k-1}$, which contradicts the assumption. The CGS algorithm requires approximately mn^2 multiplications and can be interpreted in matrix terms as follows:

Theorem 8.3.1. *The QR Factorization*

Let the matrix $A = (a_1, a_2, \ldots, a_n) \in \mathbf{R}^{m\times n}$ have linearly independent columns. Then the Gram–Schmidt algorithm computes unique matrices $Q_1 \in \mathbf{R}^{m\times n}$ with orthonormal columns and an upper triangular $R \in \mathbf{R}^{n\times n}$ with positive diagonal elements, such that

$$A = (a_1, a_2, \ldots, a_n) = (q_1, q_2, \ldots, q_n) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{2n} \end{pmatrix} \equiv Q_1 R. \quad (8.3.16)$$

Proof. Combining (8.3.14) and (8.3.15) we obtain

$$a_k = r_{kk} q_k + \sum_{i=1}^{k-1} r_{ik} q_i = \sum_{i=1}^{k} r_{ik} q_i, \quad k = 1 : n,$$

which is equivalent with (8.3.16). Since the vectors q_k are mutually orthogonal by construction the theorem follows. □

Corollary 8.3.2. *The factor R in the factorization* (8.3.16) *equals the Cholesky factor of $A^T A$. Hence the GS algorithm computes the Cholesky factor directly from A.*

Proof. The Cholesky factor R of a nonsingular matrix $A^T A$ is uniquely determined provided R is normalized to have a positive diagonal. From (8.3.16) we have $A^T A = R^T Q_1^T Q_1 R = R^T R$, and the result follows. □

For the *numerical* GS factorization of a matrix A a small reordering of the above algorithm gives the **modified Gram–Schmidt** method (MGS). Although mathematically equivalent to the classical algorithm MGS has greatly superior numerical properties, and is therefore usually to be preferred.

The modified Gram–Schmidt (MGS) algorithm employs a sequence of elementary orthogonal projections. At the beginning of step k, we have computed

$$(q_1, \ldots, q_{k-1}, a_k^{(k)}, \ldots, a_n^{(k)}),$$

where we have put $a_j = a_j^{(1)}$, $j = 1 : n$. Here $a_k^{(k)}, \ldots, a_n^{(k)}$ have already been made orthogonal to $q_1, \ldots, q_{k-1}$, which are final columns in Q_1. In the kth step q_k is obtained by normalizing the vector $a_k^{(k)}$,

$$\tilde{q}_k = a_k^{(k)}, \qquad r_{kk} = \|\tilde{q}_k\|_2, \qquad q_k = \tilde{q}_k / r_{kk}, \quad (8.3.17)$$

and then $a_{k+1}^{(k)}, \ldots, a_n^{(k)}$ are orthogonalized against q_k

$$a_j^{(k+1)} = (I_m - q_k q_k^T) a_j^{(k)} = a_j^{(k)} - r_{kj} q_k, \quad r_{kj} = q_k^T a_j^{(k)}, \quad j = k+1 : n. \tag{8.3.18}$$

After n steps we have obtained the factorization (8.3.16). Note that for $n = 2$ MGS and CGS are identical.

Algorithm 8.3.2 Modified Gram–Schmidt.
Given $A \in R^{m \times n}$ with $\text{rank}(A) = n$ the following algorithm computes the factorization $A = Q_1 R$:

```
for k = 1 : n
    q̂_k = a_k^(k);   r_kk = ‖q̂_k‖_2;
    q_k = q̂_k / r_kk;
    for j = k + 1 : n
        r_kj = q_k^T a_j^(k);
        a_j^(k+1) = a_j^(k) − r_kj q_k;
    end
end
```

The operations in Algorithm 8.3.2 can be sequenced so that the elements in R are computed in a column-wise fashion. However, the row-wise version given above is more suitable if column pivoting is to be performed; see Sec. 8.4.2.

There is also a **square root free** version of the modified Gram–Schmidt orthogonalization method, which results if the normalization of the vectors $\tilde{q}_k$ is omitted. In this version one computes scaled factors $\tilde{Q}_1 = (\tilde{q}_1, \ldots, \tilde{q}_n)$ and $\tilde{R}$ so that

$$A = \tilde{Q}_1 \tilde{R},$$

where $\tilde{R}$ is **unit** upper triangular. We take $\tilde{r}_{kk} = 1$, $d_k = \tilde{q}_k^T \tilde{q}_k$, and change (8.3.18) to

$$a_j^{(k+1)} = a_j^{(k)} - \tilde{r}_{kj} \tilde{q}_k, \qquad \tilde{r}_{kj} = \tilde{q}_k^T a_j^{(k)} / d_k, \qquad j = k+1, \ldots, n. \tag{8.3.19}$$

The unnormalized vector $\tilde{q}_k$ is just the orthogonal projection of a_k onto the complement of $\text{span}[a_1, a_2, \ldots, a_{k-1}] = \text{span}[q_1, q_2, \ldots, q_{k-1}]$.

In CGS the orthogonalization of a_k in step (8.3.14) can be written

$$\hat{q}_k = (I - Q_{k-1} Q_{k-1}^T) a_k, \quad Q_{k-1} = (q_1, \ldots, q_{k-1}).$$

In MGS *the projections $r_{ik} q_i$ are subtracted from a_k as soon as they are computed*, which corresponds to computing

$$\hat{q}_k = (I - q_{k-1} q_{k-1}^T) \cdots (I - q_1 q_1^T) a_k.$$

For $k > 2$ these two expressions are identical only if the $q_1, \ldots, q_{k-1}$ are accurately orthogonal. However, due to round-off there will be a gradual (sometimes catastrophic) loss of orthogonality. In this respect CGS and MGS behave very differently.

In MGS the loss of orthogonality occurs in a predictable manner proportional to the $\kappa(A)$. This is not the case for CGS.

Loss of orthogonality will occur in orthogonalization whenever cancellation takes place in subtracting the orthogonal projection on q_i from $a_k^{(i)}$, that is when

$$a_j^{(k+1)} = (I - q_k q_k^T) a_j^{(k)}, \qquad \|a_k^{(i+1)}\|_2 \ll \alpha \|a_k^{(i)}\|_2. \tag{8.3.20}$$

Consider the case of orthogonalizing *two* vectors. Given a vector a_2, we want to orthogonalize it against a vector q_1, $\|q_1\|_2 = 1$, by computing

$$\hat{q}_2 = a_2 - r_{12} q_1, \quad r_{12} = q_1^T a_2. \tag{8.3.21}$$

We use the standard model for floating point computation, and the basic results in Sec. 2.3.2 to analyze the rounding errors. For the *computed* scalar product $\bar{r}_{12} = fl(q_1^T a_2)$ we get

$$|\bar{r}_{12} - r_{12}| < \gamma_m \|a_2\|_2, \quad \gamma_m = \frac{mu}{1 - mu/2},$$

where u is the unit roundoff. Using $|r_{12}| \leq \|a_2\|_2$ we obtain for $\bar{q}_2 = fl(a_2 - fl(\bar{r}_{12} q_1))$

$$\|\bar{q}_2 - \hat{q}_2\|_2 < \gamma_{m+2} \|a_2\|_2.$$

Since $q_1^T \hat{q}_2 = 0$, it follows that $|q_1^T \bar{q}_2| < \gamma_{m+2} \|a_2\|_2$ and the loss of orthogonality

$$\frac{|q_1^T \bar{q}_2|}{\|\bar{q}_2\|_2} \approx \frac{|q_1^T \bar{q}_2|}{\|\hat{q}_2\|_2} < \gamma_{m+2} \frac{\|a_2\|_2}{\|\tilde{q}_2\|_2} = \frac{\gamma_{m+2}}{\sin \phi(q_1, a_2)}, \tag{8.3.22}$$

is proportional to $\phi(q_1, a_2)$, the angle between q_1 and a_2.

Example 8.3.3. As an illustration consider the matrix

$$A = (a_1, a_2) = \begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix}.$$

Using the Gram–Schmidt algorithm and IEEE double precision we get

$$q_1 = \begin{pmatrix} 0.98640009002732 \\ 0.16436198585466 \end{pmatrix},$$

$r_{12} = q_1^T a_2 = 0.87672336001729$,

$$\hat{q}_2 = a_2 - r_{12} q_1 = \begin{pmatrix} -0.12501091273265 \\ 0.75023914025696 \end{pmatrix} 10^{-8},$$

$$q_2 = \begin{pmatrix} -0.16436196071471 \\ 0.98640009421635 \end{pmatrix},$$

and

$$R = \begin{pmatrix} 1.31478090189963 & 0.87672336001729 \\ 0 & 0.00000000760583 \end{pmatrix}.$$

Severe cancellation has taken place when computing $\hat{q}_2$, which leads to a serious loss of orthogonality between q_1 and q_2:

$$q_1^T q_2 = 2.5486557 \cdot 10^{-8},$$

which should be compared with the unit roundoff $1.11 \cdot 10^{-16}$. We note that the loss of orthogonality is roughly equal to a factor 10^{-8}.

Reorthogonalizing the computed vector $a_2^{(2)}$ against q_1 we obtain

$$q_1^T q_2 = 2.5486557 \cdot 10^{-8}, \quad \tilde{q}_2 = \begin{pmatrix} -0.16436198585466 \\ 0.98640009002732 \end{pmatrix}.$$

The vector $\tilde{q}_2$ is exactly orthogonal to q_1.

For MGS the loss of orthogonality can be bounded in terms of the condition number $\kappa(A)$ also for $n > 2$. (Note that for $n = 2$ MGS and CGS are the same.) In can be shown that if $c_2 \kappa u < 1$, then

$$\|I - \bar{Q}_1^T \bar{Q}_1\|_2 \leq \frac{c_1}{1 - c_2 \kappa u} \kappa u.$$

where c_1 and c_2 denote constants depending on m, n, and the details of the arithmetic. In contrast, the computed vectors q_k from CGS may depart from orthogonality to an almost arbitrary extent. The more gradual loss of orthogonality in the computed vectors q_i for MGS is illustrated in the example below; see also Problem 1.

Example 8.3.4. A matrix $A \in \mathbf{R}^{50 \times 10}$ was generated by computing

$$A = U \mathrm{diag}\,(1, 10^{-1}, \ldots, 10^{-9}) V^T$$

where U and V are orthonormal matrices. Hence A has singular values $\sigma_i = 10^{-i+1}$, $i = 1 : 10$, and $\kappa(A) = 10^9$. Fig. 8.5.1 shows the condition number of $A_k = (a_1, \ldots, a_k)$ and the loss of orthogonality in CGS and MGS after k steps as measured by $\|I_k - Q_k^T Q_k\|_2$.

For MGS the loss of orthogonality is more gradual and proportional to $\kappa(A_k)$, whereas for CGS the loss of orthogonality is roughly proportional to $\kappa^2(A_k)$,

In some applications it is important not only that the computed $\bar{Q}_1$ and $\bar{R}$ are such that $\bar{Q}_1 \bar{R}$ accurately represents A, but also that $\bar{Q}_1$ is accurately orthogonal. We call this the **orthogonal basis problem**. It can be show that for MGS it holds that

$$A + E = \bar{Q}_1 \bar{R}, \quad \|E\|_2 \leq c_0 u \|A\|_2.$$

However, to satisfy the second condition it is necessary to **reorthogonalize** the computed vectors in the Gram–Schmidt algorithm, whenever (8.3.20) is satisfied for some suitably chosen parameter $\alpha < 1$ typically chosen in the range $[0.1, 1/\sqrt{2}]$. In a sense to be made more precise below, *one reorthogonalization will always suffice*. Hence reorthogonalization will at most double the cost of the Gram–Schmidt factorization.

Table 8.3.1. *Loss of orthogonality and CGS and MGS.*

k	$\kappa(A_k)$	$\|I_k - Q_C^T Q_C\|_2$	$\|I_k - Q_M^T Q_M\|_2$
1	1.000e+00	1.110e-16	1.110e-16
2	1.335e+01	2.880e-16	2.880e-16
3	1.676e+02	7.295e-15	8.108e-15
4	1.126e+03	2.835e-13	4.411e-14
5	4.853e+05	1.973e-09	2.911e-11
6	5.070e+05	5.951e-08	3.087e-11
7	1.713e+06	2.002e-07	1.084e-10
8	1.158e+07	1.682e-04	6.367e-10
9	1.013e+08	3.330e-02	8.779e-09
10	1.000e+09	5.446e-01	4.563e-08

For the case $n = 2$ the following result is known:

Algorithm 8.3.3 Kahan–Parlett algorithm Parlett [48, Sec. 6.9].

Suppose that for any given z the expression $\bar{p} := \text{orthog}(a_1, z)$ computes an approximation to the (exact) orthogonal complement $p = z - a_1(a_1^T z)/\|a_1\|_2^2$ of z to a_1, such that the error satisfies $\|\bar{p} - p\|_2 \leq \epsilon\|z\|_2$ for some tiny positive ϵ. Then, given $A = (a_1, a_2)$, the following algorithm computes a vector $\bar{q}_2$, which satisfies

$$\|\bar{q}_2 - q_2\|_2 \leq (1+\alpha)\epsilon\|a_2\|_2, \qquad \|a_1^T \bar{q}_2\| \leq \epsilon\alpha^{-1}\|\bar{q}_2\|_2\|a_1\|_2, \tag{8.3.23}$$

where q_2 is the exact complement of a_2 orthogonal to a_1. The first inequality implies that $\bar{q}_2$ is close to a linear combination of a_1 and a_2. The second says that $\bar{q}_2$ is nearly orthogonal to a_1.

$\bar{q}_2 := \text{orthog}(a_1, a_2)$;
if $\|\bar{q}_2\|_2 < \alpha\|a_2\|_2$
  $\check{q}_2 := \text{orthog}(a_1, \bar{q}_2)$; (reorthogonalize$\bar{q}_2$)
  if $\|\check{q}_2\|_2 \geq \alpha\|\bar{q}_2\|_2$ $\check{q}_2$;
  else $\bar{q}_2 = \bar{q}_2 := 0$; (numerically singular case)
  end
end

Note that if $\|\check{q}_2\|_2 < \alpha\|\bar{q}_2\|_2$ we conclude that the given vectors (a_1, a_2) are linearly dependent and and signal this by setting $\bar{q}_2 := 0$.
When α is large, say $\alpha \geq 1/\sqrt{2}$, then the bounds in (8.3.23) are very good but reorthogonalization will occur more frequently. If α is small, reorthogonalization will be rarer, but the bound on orthogonality less good. For larger n there seems to be a good case for recommending the stringent value $\alpha = 1/\sqrt{2}$ or *always* perform one step of reorthogonalization ($\alpha = 1$).

Now consider the case $n > 2$. Assume we are given a matrix $Q_1 = (q_1, \ldots, q_{k-1})$

with $\|q_1\|_2 = \ldots = \|q_{k-1}\|_2 = 1$. Adding the new vector a_k, we want to compute a vector $\hat{q}_k$ such that

$$\hat{q}_k \in \text{span}(Q_1, a_k) \perp \text{span}(Q_1).$$

The solution equals $\hat{q}_k = a_k - Q_1 r_k$, where r_k solves the least squares problem

$$\min_{r_k} \|a_k - Q_1 r_k\|_2.$$

We first assume that Q_1 is accurately orthogonal. Then it can be rigorously proved that it suffices to run MGS twice on the matrix (Q_1, a_k). This generalizes the result by Kahan–Parlett to $n > 2$.

To solve the problem, when the columns of Q_1 are not accurately orthogonal, we can use **iterated** Gram–Schmidt methods. In the iterated CGS algorithm we put $\hat{q}_k^{(0)} := a_k$, $r_k^{(0)} := 0$, and for $p = 0, 1, \ldots$ compute

$$s_k^{(p)} := Q_1^T \hat{q}_k^{(p)}, \quad \hat{q}_k^{(p+1)} := \hat{q}_k^{(p)} - Q_1 s_k^{(p)}, \quad r_k^{(p+1)} := r_k^{(p)} + s_k^{(p)}.$$

The first step of this algorithm is the usual CGS algorithm, and each step is a reorthogonalization. The iterated MGS algorithm is similar, except that each projection is subtracted as soon as it computed: As in the Kahan–Parlett algorithm, the iterations can be stopped when $\|\hat{q}_k^{(p+1)}\|_2 > \alpha \|\hat{q}_k^{p}\|_2$.

The iterated Gram–Schmidt algorithm can be used recursively, adding one column a_k at a time, to compute the factorization $A = Q_1 R$. If A has full numerical column rank, then with $\alpha = 1/\sqrt{2}$ both iterated CGS and MGS computes a factor Q_1, which is orthogonal to almost full working precision, *using at most one reorthogonalization.* Hence in this case iterated CGS is *not* inferior to the iterated MGS.

8.3.3 Least Squares Problems by Gram–Schmidt

We now consider the use of the Modified Gram–Schmidt algorithm for solving linear least squares problem. It is important to note that because of the loss of orthogonality in Q_1 *computing* x *by forming* $c_1 = Q_1^T b$ *and then solving* $Rx = c_1$ *will not in general give an accurate solution.* Using the MGS factorization in this way seems to have contributed to an undeserved bad reputation of the method. Used correctly, as described below, the MGS factorization will give as accurate results as any competing method.

To solve a least squares problems the MGS algorithm is applied to the augmented matrix (A, b). If we skip the normalization of the $(n+1)$st column we obtain a factorization

$$(A,\, b) = (Q_1,\, r) \begin{pmatrix} R & z \\ 0 & 1 \end{pmatrix}, \tag{8.3.24}$$

where r is the residual vector. We have

$$\|Ax - b\|_2 = \left\| (A, b) \begin{pmatrix} x \\ -1 \end{pmatrix} \right\|_2 = \|Q_1(Rx - z) - r\|_2.$$

Let us assume that $q_{n+1} = r/\|r\|_2$ is orthogonal to Q_1. Then the minimum of the last expression occurs when $Rx - z = 0$ and the least squares residual equals r. Although this assumptions is not true to machine precision, we note that it is not necessary to assume that Q_1 is accurately orthogonal for the conclusion to hold. This heuristic argument leads to the following algorithm for solving linear least squares problems by MGS, which can be proved to be backward stable for computing the solution x:

Algorithm 8.3.4 Linear Least Squares Solution by MGS.
Carry out MGS on $A \in R^{m\times n}$, $\text{rank}(A) = n$, to give $Q_1 = (q_1, \ldots, q_n)$ and R, and put $b^{(1)} = b$. Compute the vector $z = (z_1, \ldots, z_n)^T$ by

$$\begin{aligned}
&\text{for } k = 1, 2, \ldots, n \\
&\quad z_k = q_k^T b^{(k)}; \quad b^{(k+1)} = b^{(k)} - z_k q_k; \\
&\text{end} \\
&r = b^{(n+1)}; \\
&\text{solve } Rx = z;
\end{aligned}$$

If implemented as above MGS gives very accurate results. Unfortunately, a common error can still found in some textbooks. This is to compute R by MGS, but in the final step solve $Rx = Q_1^T b$. *This destroys the accuracy and may be one reason the MGS method is not widely used.*

In some applications it is important to use an algorithm which is backwards stable for the computed residual $\bar{r}$, i.e. we want a relation

$$(A + E)^T \bar{r} = 0, \quad \|E\|_2 \le cu\|A\|_2, \tag{8.3.25}$$

to hold for some constant c. This implies that $A^T\bar{r} = -E^T\bar{r}$, and

$$\|A^T\bar{r}\|_2 \le cu\|\bar{r}\|_2\|A\|_2. \tag{8.3.26}$$

Note that this is much better than if we compute

$$\bar{r} = fl(b - fl(Ax)) = fl\left(\begin{pmatrix} b & A \end{pmatrix}\begin{pmatrix} 1 \\ -x \end{pmatrix}\right)$$

even when x is the *exact* least squares solution. We obtain using (2.3.13) and $A^T r = 0$

$$|A^T\bar{r}| < \gamma_{n+1}|A^T|(|b| + |A||x|).$$

From this we get the norm-wise bound

$$\|A^T\bar{r}\|_2 \le n^{1/2}\gamma_{n+1}\|A\|_2(\|b\|_2 + n^{1/2}\|A\|_2\|x\|_2),$$

which is a much weaker than (8.3.26) when, as is often the case, $\|\bar{r}\|_2 \ll \|b\|_2$!

An obvious remedy seems to be to reorthogonalize the computed residual r against $Q_1 = (q_1, q_2, \ldots, q_n)$. However, to obtain a backward stable algorithm for r this should be done in *reverse order*!

Algorithm 8.3.5 Orthogonal projection by MGS.
To make Algorithm 8.3.3 backward stable for r it suffices to add a loop where the vector $b^{(n+1)}$ is orthogonalized against $q_n, q_{n-1}, \ldots, q_1$ (*note the order*):

$$\begin{aligned}
&\text{for } k = n, n-1, \ldots, 1\\
&\quad z_k = q_k^T b^{(k+1)}; \quad b^{(k)} = b^{(k+1)} - z_k q_k;\\
&\text{end}\\
&r = b^{(1)};
\end{aligned}$$

It can be proved that this step "magically" compensates for the lack of orthogonality of Q_1 and the $\bar{r}$ computed by Algorithm 8.3.3 satisfies (8.3.25).

A similar idea is used to construct a backward stable algorithm for the minimum norm problem

$$\min \|y\|_2, \qquad A^T y = c.$$

Algorithm 8.3.6 Minimum Norm Solution by MGS.
Carry out MGS on $A^T \in R^{m\times n}$, with $\operatorname{rank}(A) = n$ to give $Q_1 = (q_1, \ldots, q_n)$ and R. Then the minimum norm solution $y = y^{(0)}$ is obtained from

$$\begin{aligned}
&R^T(\zeta_1, \ldots, \zeta_n)^T = c;\\
&y^{(n)} = 0;\\
&\text{for } k = n, \ldots, 2, 1\\
&\quad \omega_k = q_k^T y^{(k)}; \quad y^{(k-1)} = y^{(k)} - (\omega_k - \zeta_k) q_k;\\
&\text{end}
\end{aligned}$$

If the columns of Q_1 were orthogonal to working accuracy, then $\omega_k = 0$, $k = m, \ldots, 1$. Hence ω compensates for the lack of orthogonality to make this algorithm backwards stable!

8.3.4 Householder and Givens Transformations

Orthogonal matrices which are equal to the unit matrix modified by a matrix of rank one are called **elementary orthogonal matrices**. Such matrices are flexible and useful tools for constructing algorithms for solving a variety of problems in linear algebra. They are attractive since multiplication of a vector with an orthogonal matrix preserves the Euclidean length and hence there use leads to numerically stable algorithms.

Recall that a square matrix $Q \in \mathbf{R}^{m\times m}$, is called orthogonal if $Q^T Q = I$. Then $Q^{-1} = Q^T$, and hence $QQ^T = I$. Taking the determinant of both sides

$$\det(Q^T Q) = \det(Q^T)\det(Q) = \det(Q)^2 = 1.$$

and hence $\det(Q) = \pm 1$. A very important class of orthogonal transformations are matrices of the form

$$H = I - \beta uu^T, \quad \beta = 2/(u^T u). \tag{8.3.27}$$

By construction H is symmetric $H^T = H$, and using (8.3.27) we have

$$H^T H = H^2 = I - 2\beta uu^T + \beta^2 u(u^T u)u^T = I.$$

Hence H is orthogonal, and $H^2 = I$. The product Ha where a is a given vector can be computed without explicitly forming H itself using

$$Ha = (I - \beta uu^T)a = a - \beta u(u^T a).$$

Note that $Ha \in \text{span}[a, u]$. We have $Hu = -u$, i.e., H reverses u. Further $Ha = a$, for $a \perp u$. Hence H has $m-1$ eigenvalues equal to $+1$ and one equal to -1, and thus $\det(H) = -1$. The effect of the transformation Ha for a general vector a is to reflect a in the $(m-1)$ dimensional hyperplane characterized by the normal vector u, see Fig. 8.5.1. Therefore, H is called an **elementary reflector**. The use of elementary reflectors in numerical linear algebra was initiated by A. S. Householder. Matrices of the form (8.3.27) are therefore often called **Householder reflectors** and the vector u is called a **Householder vector**.

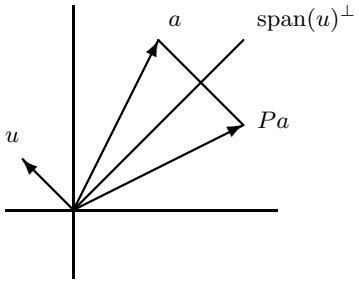

Figure 8.3.2.

Fig. 8.4.1 shows the vector a mapped into Ha by a reflection in the plane with normal vector u. Note that this is equivalent to subtracting twice the orthogonal projection onto u. Further the normal u is parallel to the difference $(a - Pa)$. Given $a \neq 0 \in \mathbf{R}^m$, we consider the problem of constructing a plane reflection $H \in \mathbf{R}^{m\times m}$ such that *multiplication by H zeros all components except the first in a*, i.e.,

$$Ha = \pm\sigma e_1, \quad \sigma = \|a\|_2. \tag{8.3.28}$$

Multiplying (8.3.28) from the left by H and using $H^2 = I$ it follows that $y = Ue_1$ satisfies $U^T y = e_1$ or

$$He_1 = \pm a/\sigma.$$

Hence (8.3.28) is equivalent to finding a square orthogonal matrix H with its first column proportional to $\pm a/\sigma$. It is easily seen that (8.3.28) is satisfied if we take

$$u = a \mp \sigma e_1 = \begin{pmatrix} \alpha_1 \mp \sigma \\ a_2 \end{pmatrix}, \quad a = \begin{pmatrix} \alpha_1 \\ a_2 \end{pmatrix}. \tag{8.3.29}$$

Note that u differs from a only in its first component. A short calculation shows that

$$1/\beta = \tfrac{1}{2}u^T u = \tfrac{1}{2}(a \mp \sigma e_1)^T(a \mp \sigma e_1) = \tfrac{1}{2}(\sigma^2 \mp 2\sigma\alpha_1 + \sigma^2) = \sigma(\sigma \mp \alpha_1).$$

If a is close to a multiple of e_1, then $\sigma \approx |\alpha_1|$ and cancellation may lead to a large relative error in β. To avoid this we take

$$u = a + \operatorname{sign}(\alpha_1)\sigma e_1, \qquad 1/\beta = \sigma(\sigma + |\alpha_1|), \tag{8.3.30}$$

which gives

$$Ha = -\operatorname{sign}(\alpha_1)\sigma e_1 = \hat{\sigma} e_1.$$

Note that with this choice of sign the vector $a = e_1$ will be mapped onto $-e_1$. (It is possible to rewrite the formula in (8.3.30) for β so that the other choice of sign does not give rise to numerical cancellation; for details see Parlett [48, pp. 91].)

The Householder transformation in (8.3.27) does not depend on the scaling of u. It is often more convenient to scale u so that its first component equals 1. If we write

$$H = I - \beta u u^T, \quad u = \begin{pmatrix} 1 \\ u_2 \end{pmatrix},$$

then

$$\beta = 1 + |\alpha_1|/\sigma. \qquad u_2 = \rho a_2, \quad \rho = \operatorname{sign}(\alpha_1)/(\sigma + |\alpha_1|), \tag{8.3.31}$$

This has the advantage that we can stably reconstruct β from u_2 using

$$\beta = 2/(u^T u) = 2/(1 + u_2^T u_2).$$

Algorithm 8.3.7 Let $a \in \mathbf{R}^m$ be a vector with $\|a\|_2 = \sigma$ and $a^T e_1 = \alpha_1$. The following algorithm constructs a Householder transformation $H = I - \beta u u^T$, where $u^T e_1 = 1$, such that $Ha = -sign(\alpha_1)\hat{\sigma} e_1$, where $\hat{\sigma} = -sign(\alpha_1)\sigma$.

$$\begin{aligned}
&[u, \beta, \hat{\sigma}] = \text{house}(a)\\
&\quad \alpha_1 = a(1);\\
&\quad \sigma = \|a\|_2;\\
&\quad \beta = 1 + |\alpha_1|/\sigma;\\
&\quad \hat{\sigma} = -\operatorname{sign}(\alpha_1)\sigma;\\
&\quad \rho = -1/(\hat{\sigma}\beta);\\
&\quad u = [1; \rho \cdot a(2:m)];
\end{aligned}$$

If a matrix $A = (a_1, \ldots, a_n) \in \mathbf{R}^{m\times n}$ is *premultiplied* by H the product can be computed in $2mn$ multiplications as

$$HA = (Ha_1, \ldots, Ha_n), \qquad Ha_j = a_j - \beta(u^T a_j)u. \tag{8.3.32}$$

An analogous formula, exists for *postmultiplying* A with H, where H now acts on the *rows* of A. Writing the products HA and AH as

$$HA = A - \beta u(u^T A), \qquad AH = A - \beta(Au)u^T,$$

shows that in both cases is A altered by a matrix of rank one.

Another useful class of orthogonal transformations are the matrices representing **plane rotations**, which are also called **Givens rotations** after Wallace Givens, who popularized their use for numerical computations. In $\mathbf{R}^2$ the matrix representing a rotation clockwise through an angle θ is

$$G(\theta) = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \qquad c = \cos\theta, \quad s = \sin\theta. \tag{8.3.33}$$

Note that $G^{-1}(\theta) = G(-\theta)$, and $\det G(\theta) = +1$.

In $\mathbf{R}^m$ the matrix representing a rotation in the plane spanned by the unit vectors e_i and e_j, $i < j$, is the following rank two modification of the unit matrix I_m

$$G_{ij}(\theta) = \begin{array}{c} \\ \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & i & & j & & \end{array} \\ \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix} \end{array}. \tag{8.3.34}$$

Premultiplying a vector $a = (\alpha_1, \ldots, \alpha_m)^T$ by $G_{ij}(\theta)$ we get

$$G_{ij}(\theta)a = (\tilde{\alpha}_1, \ldots, \tilde{\alpha}_m)^T, \qquad \tilde{\alpha}_k = \begin{cases} \alpha_k, & k \neq i, j; \\ c\alpha_i + s\alpha_j, & k = i; \\ -s\alpha_i + c\alpha_j, & k = j. \end{cases} \tag{8.3.35}$$

Thus a plane rotation may be multiplied into a vector at a cost of two additions and four multiplications. We can determine the rotation $G_{ij}(\theta)$ so that $\tilde{\alpha}_j$ becomes zero by taking

$$c = \alpha_i/\sigma, \qquad s = \alpha_j/\sigma, \qquad \sigma = (\alpha_i^2 + \alpha_j^2)^{1/2} \neq 0. \tag{8.3.36}$$

Note that $-G(\theta)$ also zeros $\tilde{\alpha}_j$ so c and s are only determined up to a factor ± 1.

To guard against possible overflow, the Givens rotation should be computed as in the following procedure:

Algorithm 8.3.8 Given $(\alpha, \beta)^T \neq 0$ the algorithm constructs c, s, σ such that $s^2 + c^2 = 1$ and

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \sigma \\ 0 \end{pmatrix}:$$

$$
\begin{aligned}
&[c, s, \sigma] = \text{givrot}(\alpha, \beta) \\
&\quad \text{if } |\alpha| > |\beta| \\
&\qquad t = \beta/\alpha; \quad c = 1/\sqrt{1+t^2}; \\
&\qquad s = tc; \quad \sigma = \alpha/c; \\
&\quad \text{else} \\
&\qquad t = \alpha/\beta; \quad s = 1/\sqrt{1+t^2}; \\
&\qquad c = ts; \quad \sigma = \beta/s; \\
&\quad \text{end}
\end{aligned}
$$

Premultiplication of a matrix $A \in R^{m\times n}$ with a Givens rotation G_{ij} will only affect the two *rows* i and j in A, which are transformed according to

$$a_{ik} := ca_{ik} + sa_{jk}, \tag{8.3.37}$$

$$a_{jk} := -sa_{ik} + ca_{jk}, \quad k = 1, 2, \ldots, n. \tag{8.3.38}$$

The product requires $4n$ multiplications. An analogous algorithm, which only affects *columns* i and j, exists for postmultiplying A with G_{ij}.

Givens rotations can be used in several different ways to construct an orthogonal matrix U, which satisfies (8.3.28). Let G_{1k}, $k = 2, \ldots, m$ be a sequence of Givens rotations, where G_{1k} is determined to zero the kth component in the vector a,

$$G_{1m} \ldots G_{13} G_{12} a = \sigma e_1.$$

Note that G_{1k} will not destroy previously introduced zeros. Another possible sequence is $G_{k-1,k}$, $k = m, m-1, \ldots, 2$, where $G_{k-1,k}$ is chosen to zero the kth component. This demonstrates the flexibility of Givens rotations compared to reflectors.

It is essential to note that the matrix G_{ij} is never explicitly formed, but represented by (i, j) and the two numbers c and s. When a large number of rotations need to be stored it is more economical to store just a single number, from which c and s can be retrieved in a numerically stable way. Since the formula $\sqrt{1-x^2}$ is poor if $|x|$ is close to unity a slightly more complicated method than storing just c or s is needed. In a scheme devised by G. W. Stewart one stores the number c or s of smallest magnitude. To distinguish between the two cases one stores the reciprocal of c. More precisely, if $c \neq 0$ we store

$$\rho = \begin{cases} s, & \text{if } |s| < |c|; \\ 1/c, & \text{if } |c| \leq |s| \end{cases}.$$

In case $c = 0$ we put $\rho = 1$, a value that cannot appear otherwise.

To reconstruct the Givens rotation, if $\rho = 1$, we take $s = 1$, $c = 0$, and

$$\rho = \begin{cases} s = \rho, & c = \sqrt{1-s^2}, & \text{if } |\rho| < 1; \\ c = 1/\rho, & s = \sqrt{1-c^2}, & \text{if } |\rho| > 1; \end{cases}$$

It is possible to rearrange the Givens rotations so that it uses only two instead of four multiplications per element and no square root. These modified transformations called "fast" Givens transformations, and are described in Golub and Van Loan [29, 1996, Sec. 5.1.13].

8.3.5 Householder QR Factorization

Methods for solving the linear least squares problem which, like the SVD, are based on orthogonal transformations avoid the squaring of the condition number that results from forming the normal equations. In this section we first develop algorithms using elementary orthogonal transformations to factor a matrix $A \in \mathbf{R}^{m\times n}$ $(m \geq n)$ into the product of a *square* orthogonal matrix $Q \in \mathbf{R}^{m\times m}$ and an upper triangular matrix $R \in \mathbf{R}^{m\times n}$. We then show how to use this **full QR factorization** for solving linear least squares problems.

Theorem 8.3.3. *The Full QR Factorization*

Let $A \in \mathbf{R}^{m\times n}$ with $\operatorname{rank}(A) = n$. Then there is an orthogonal matrix $Q \in \mathbf{R}^{m\times m}$ and an upper triangular matrix R with positive diagonal elements such that

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}. \tag{8.3.39}$$

Proof. A constructive proof will be given in Sec. 8.4.3. □

Since Q is orthogonal the singular values of R equal those of A and $\kappa(R) = \kappa(A)$. Indeed, to compute the SVD of A one can first compute the QR factorization and then the SVD of R.

The QR factorization can be written

$$A = (Q_1, Q_2) \begin{pmatrix} R \\ 0 \end{pmatrix} = Q_1 R. \tag{8.3.40}$$

where Q has been partitioned as $Q = (Q_1, Q_2)$, $Q_1 \in \mathbf{R}^{m\times n}$, $Q_2 \in \mathbf{R}^{m\times(m-n)}$. This is the the factorization computed by the Gram–Schmidt algorithm. From (8.3.40) it follows that the columns of Q_1 and Q_2 form orthonormal bases for the range space of A and its orthogonal complement,

$$\mathcal{R}(A) = \mathcal{R}(Q_1), \qquad \mathcal{N}(A^T) = \mathcal{R}(Q_2), \tag{8.3.41}$$

and the corresponding orthogonal projections are

$$P_{\mathcal{R}(A)} = Q_1 Q_1^T, \qquad P_{\mathcal{N}(A^T)} = Q_2 Q_2^T. \tag{8.3.42}$$

Note that although the matrix Q_1 in (8.3.40) is uniquely determined, Q_2 can be any orthogonal matrix with range $\mathcal{N}(A^T)$.

In contrast to the Gram–Schmidt algorithm for computing the QR factorization, the methods we now consider represents Q *implicitly* as a product of Householder or Givens matrices. This elegantly avoids the problem with loss of orthogonality in Q!

The QR factorization of a matrix $A \in \mathbf{R}^{m\times n}$ of rank n can be computed using a sequence of n Householder reflectors. Let $A = (a_1, a_2, \ldots, a_n)$, $\sigma_1 = \|a_1\|_2$, and choose $H_1 = I - \beta_1 u_1 u_1^T$, so that

$$H_1 a_1 = H_1 \begin{pmatrix} \alpha_1 \\ \hat{a}_1 \end{pmatrix} = \begin{pmatrix} r_{11} \\ 0 \end{pmatrix}, \qquad r_{11} = -\text{sign}\,(\alpha_1)\sigma_1.$$

By (8.3.30) we achieve this by choosing $\beta_1 = 1 + |\alpha_1|/\sigma_1$,

$$u_1 = \begin{pmatrix} 1 \\ \hat{u_1} \end{pmatrix}, \qquad \hat{u_1} = \text{sign}\,(\alpha_1)\hat{a}_1/\rho_1, \qquad \rho_1 = \sigma_1\beta_1.$$

H_1 is then applied to the remaining columns $a_2, \ldots, a_n$, giving

$$A^{(2)} = H_1 A = \begin{pmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ 0 & \tilde{a}_{22} & \ldots & \tilde{a}_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & \tilde{a}_{n2} & \ldots & \tilde{a}_{nn} \end{pmatrix}.$$

Here the first column has the desired form and, as indicated by the notation, the first row is the final first row in R. In the next step the $(m-1)\times(n-1)$ block in the lower right corner is transformed. All remaining steps, $k = 2, \ldots, n$ are similar to the first. Before the kth step we have computed a matrix of the form

$$A^{(k)} = \begin{matrix} \\ {\scriptstyle k-1} \end{matrix} \begin{matrix} \overset{k-1}{} \\ \end{matrix} \begin{pmatrix} R_{11}^{(k)} & R_{12}^{(k)} \\ 0 & \hat{A}^{(k)} \end{pmatrix}, \tag{8.3.43}$$

where the first $k-1$ rows of $A^{(k)}$ are rows in the final matrix R, and $R_{11}^{(k)}$ is upper triangular. In step k the matrix $a^{(k)}$ is transformed,

$$A^{(k+1)} = H_k A^{(k)}, \qquad H_k = \begin{pmatrix} I_k & 0 \\ 0 & \tilde{H}_k \end{pmatrix}. \tag{8.3.44}$$

Here $\tilde{H}_k = I - \beta_k u_k u_k^T$ is chosen to zero the elements below the main diagonal in the first column of the submatrix

$$\hat{A}^{(k)} = (a_k^{(k)}, \ldots, a_n^{(k)}) \in \mathbf{R}^{(m-k+1)\times(n-k+1)},$$

i.e. $\tilde{H}_k a_k^{(k)} = r_{kk} e_1$. With $\sigma_k = \|a_k^{(k)}\|_2$, using (8.3.29), we get $r_{kk} = -\text{sign}\,(a_{kk}^{(k)})\sigma_k$, and

$$\hat{u}_k = \text{sign}\,(\alpha_k^{(k)})\hat{a}_k^{(k)}/\rho_k, \qquad \beta_k = 1 + |a_{kk}|/\sigma_k. \tag{8.3.45}$$

where $\rho_k = \sigma_k\beta_k$. After n steps we have obtained the QR factorization of A, where

$$R = R_{11}^{(n+1)}, \qquad Q = H_1 H_2 \cdots H_n. \tag{8.3.46}$$

Note that the diagonal elements r_{kk} will be positive if $a_k^{(kk)}$ is negative and negative otherwise. Negative diagonal elements may be removed by multiplying the corresponding rows of R and columns of Q by -1. [2]

[2]The difference between the Householder and Gram–Schmidt QR algorithms has been aptly summarized by Trefethen, who calls Gram–Schmidt triangular orthogonalization as opposed to Householder which is orthogonal triangularization.

Algorithm 8.3.9 Householder QR Factorization.
Given a matrix $A^{(1)} = A \in \mathbf{R}^{m\times n}$ of rank n, the following algorithm computes R and Householder matrices:

$$H_k = \text{diag}\,(I_{k-1}, \tilde{H}_k), \qquad \tilde{H}_k = I - \beta_k u_k u_k^T, \quad k = 1, 2, \ldots, n, \tag{8.3.47}$$

so that $Q = H_1 H_2 \cdots H_n$.

```
for k = 1, 2, ..., n
    [u_k, β_k, r_kk] = house(a_k^(k));
    for j = k + 1, ..., n
        γ_jk = β_k u_k^T a_j^(k);
        r_kj = a_kj^(k) − γ_jk;
        a_j^(k+1) = â_j^(k) − γ_jk û_k;
    end
end
```

The vectors $\hat{u}_k$ can overwrite the elements in the strictly lower trapezoidal part of A. Thus, all information associated with the factors Q and R can be overwritten A. The vector $(\beta_1, \ldots, \beta_n)$ of length n can be recomputed from

$$\beta_k = \tfrac{1}{2}(1 + \|\hat{u}_k\|_2^2)^{1/2},$$

and therefore need not be saved. The algorithm requires $(mn^2 - n^3/3)$ multiplications, or $n^3/3$ less than for the MGS method. Note that in the special case that $m = n$ it would be possible to skip the last step which just computes $\tilde{H}_n = -1$ and $r_{nn} = -a_{nn}^{(n)}$.

Following Higham [Eq. (3.8)][33]) we will in the following frequently make use of the notation

$$\bar{\gamma}_k = \frac{cku}{1 - cku/2}, \tag{8.3.48}$$

where c denotes a small integer constant.

Theorem 8.3.4.

*Let $\bar{R}$ denote the upper triangular matrix R computed by the Householder QR algorithm. Then there exists an exactly orthogonal matrix $\hat{Q} \in \mathbf{R}^{m\times m}$ (*not *the matrix corresponding to exact computation throughout) such that*

$$A + E = \hat{Q} \begin{pmatrix} \bar{R} \\ 0 \end{pmatrix},$$

where

$$\|e_j\|_2 \le \bar{\gamma}_n \|a_j\|_2, \quad j = 1 : n,$$

As have been stressed before it is usually not advisable to compute the matrix Q in the QR factorization explicitly, even when it is to be used in later calculations. In the rare case that the $Q = H_1 H_2 \cdots H_n$ from the Householder algorithm is explicitly required it should be accumulated in backward order by taking $Q^{(n)} = I_m$, and computing $Q = Q^{(0)}$ in $2(m^2 n - mn^2 + n^3/3)$ flops by the recursion

$$Q^{(k-1)} = H_k Q^{(k)}, \quad k = n : -1 : 1.$$

Note that by setting

$$Q^{(n)} = \begin{pmatrix} I_n \\ 0 \end{pmatrix}, \quad \text{or} \quad Q^{(n)} = \begin{pmatrix} 0 \\ I_{m-n} \end{pmatrix},$$

we can similarly accumulate Q_1 or Q_2 separately. $mn^2 - n^3/3$ and $2m^2 n - 3mn^2 + n^3$ flops, respectively; see Problem 3.

It is often advantageous to use **column pivoting** in the QR factorization and compute

$$AP = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \tag{8.3.49}$$

where P is a permutation matrix. The following simple pivoting strategy, first suggested by Golub, has been shown to work well in practice. Assume that after k steps in the Householder Algorithm 7.3.3 we have computed the partial QR factorization

$$A^{(k+1)} = (H_k \cdots H_1) A (\Pi_1 \cdots \Pi_k) = \begin{pmatrix} R_{11}^{(k+1)} & R_{12}^{(k+1)} \\ 0 & \tilde{A}^{(k+1)} \end{pmatrix}, \tag{8.3.50}$$

Then the pivot column in the next step is chosen as a column of largest norm in the submatrix

$$\tilde{A}^{(k+1)} = (\tilde{a}_{k+1}^{(k+1)}, \ldots, \tilde{a}_n^{(k+1)}) \in \mathbf{R}^{(m-k)\times(n-k)},$$

i.e., Π_{k+1} is chosen to interchange columns p and $k+1$, where p is the smallest index such that

$$s_p^{(k+1)} \geq s_j^{(k+1)}, \qquad s_j^{(k+1)} = \|\tilde{a}_j^{(k+1)}\|_2^2, \qquad j = k+1, \ldots, n. \tag{8.3.51}$$

If $s_p^{(k+1)} = 0$ then the algorithm terminates with $\tilde{A}^{(k+1)} = 0$ in (8.3.50). This pivoting strategy can be viewed as choosing a remaining column of largest distance to the subspace spanned by the previously chosen columns. This is equivalent to maximizing the diagonal element $r_{k+1,k+1}$.

If the column norms in $\tilde{a}^{(k)}$ were recomputed at each stage, then column pivoting would increase the operation count by 50%. Instead the norms of the columns of A can be computed initially, and recursively updated as the factorization proceeds. This reduces the overhead of column pivoting to $0(mn)$ operations. This pivoting strategy can also be implemented in the Cholesky and modified Gram–Schmidt algorithms.

Since column norms are preserved by left orthogonal transformations it is easily shown that the elements in R, computed by QR factorization with pivoting, satisfy

$$r_{kk}^2 \geq \sum_{i=k}^{j} r_{ij}^2, \quad j = k+1, \ldots, n. \tag{8.3.52}$$

This implies in particular that $|r_{kk}| \geq |r_{kj}|$, $j > k$ and that the diagonal elements form a non-increasing sequence,

$$|r_{11}| \geq |r_{22}| \geq \cdots \geq |r_{nn}|. \tag{8.3.53}$$

To obtain near-peak performance for large dense matrix computations on current computing architectures requires code that is dominated by matrix-matrix operations since these involve less data movement per floating point computation. The QR factorization should therefore be organized in partitioned or blocked form, where the operations have been reordered and grouped into matrix operations.

For the QR factorization $A \in \mathbf{R}^{m\times n}$ $(m \geq n)$ is partitioned as

$$A = (A_1,\ A_2), \qquad A_1 \in \mathbf{R}^{m\times nb}, \tag{8.3.54}$$

where nb is a suitable block size and the QR factorization

$$Q_1^T A_1 = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}, \qquad Q_1 = H_1 H_2 \cdots H_{nb}, \tag{8.3.55}$$

is computed, where $H_i = I - u_i u_i^T$ are Householder reflections. Then the remaining columns A_2 are are updated

$$Q_1^T A_2 = Q_1^T \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} = \begin{pmatrix} R_{12} \\ \tilde{A}_{22} \end{pmatrix}. \tag{8.3.56}$$

In the next step we partition $\tilde{A}_{22} = (B_1, B_2)$, and compute the QR factorization of $B_1 \in \mathbf{R}^{(m-r)\times r}$. Then B_2 is updated as above, and we continue in this way until the columns in A are exhausted.

A major part of the computation in spent in the updating step (8.3.56). As written this step cannot use BLAS-3, which slows down the execution. To achieve better performance it is essential that this part is sped up. The solution is to aggregate the Householder transformations so that their application can be expressed as matrix operations. For use in the next subsection, we give a slightly more general result.

Lemma 8.3.5.

Let $H_1, H_2, \ldots, H_r$ be a sequence of Householder transformations. Set $r = r_1 + r_2$, and assume that

$$Q_1 = H_1 \cdots H_{r_1} = I - Y_1 T_1 Y_1^T, \qquad Q_2 = H_{r_1+1} \cdots H_r = I - Y_2 T_2 Y_2^T,$$

where $T_1, T_2 \in \mathbf{R}^{r\times r}$ are upper triangular matrices. Then for the product $Q_1 Q_2$ we have

$$Q = Q_1 Q_2 = (I - Y_1 T_1 Y_1^T)(I - Y_2 T_2 Y_2^T) = (I - YTY^T) \tag{8.3.57}$$

where

$$\hat{Y} = (Y_1,\ Y_2), \qquad \hat{T} = \begin{pmatrix} T_1 & -(T_1 Y_1^T)(Y_2 T_2) \\ 0 & T_2 \end{pmatrix}. \tag{8.3.58}$$

Note that Y is formed by concatenation, but computing the off-diagonal block in T requires extra operations.

For the partitioned algorithm we use the special case when $r_2 = 1$ to aggregate the Householder transformations for each processed block. Starting with $Q_1 = I - \tau_1 u_1 u_1^T$, we set $Y = u_1$, $T = \tau_1$ and update

$$Y := (Y,\ u_{k+1}), \qquad T := \begin{pmatrix} T & -\tau_k T Y^T u_k \\ 0 & \tau_k \end{pmatrix}, \quad k = 2 : nb. \tag{8.3.59}$$

Note that Y will have a trapezoidal form and thus the matrices Y and R can overwrite the matrix A. With the representation $Q = (I - YTY^T)$ the updating of A_2 becomes

$$B = Q_1^T A = (I - YT^T Y^T)A_2 = A_2 - YT^T Y^T A_2,$$

which now involves only matrix operations.

This partitioned algorithm requires more storage and operations than the point algorithm, namely those needed to produce and store the T matrices. However, for large matrices this is more than offset by the increased rate of execution.

As mentioned in Chapter 7 recursive algorithms can be developed into highly efficient algorithms for high performance computers and are an alternative to the currently more used partitioned algorithms by LAPACK. The reason for this is that recursion leads to automatic variable blocking that dynamically adjusts to an arbitrary number of levels of memory hierarchy.

Consider the partitioned QR factorization

$$A = (\, A_1 \quad A_2 \,) = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

where Let A_1 consist of the first $\lfloor n/2 \rfloor$ columns of A. To develop a recursive algorithm we start with a QR factorization of A_1 and update the remaining part A_2 of the matrix,

$$Q_1^T A_1 = \begin{pmatrix} R_{11} \\ 0 \end{pmatrix}, \qquad Q_1^T A_2 = Q_1^T \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} = \begin{pmatrix} R_{12} \\ \tilde{A}_{22} \end{pmatrix}.$$

Next $\tilde{A}_{22}$ is recursively QR decomposed giving Q_2, R_{22}, and $Q = Q_1 Q_2$.

As an illustration we give below a simple implementation in Matlab, which is convenient to use since it allows for the definition of recursive functions.

```
function [Y,T,R] = recqr(A)
%
% RECQR computes the QR factorization of the m by n matrix A,
% (m >= n). Output is the n by n triangular factor R, and
```

```
% Q = (I - YTY')  represented in aggregated form, where Y is
% m by n and unit lower trapezoidal, and T is n by n upper
% triangular
    [m,n] = size(A);
    if n == 1
    [Y,T,R] = house(A);
    else
      n1 = floor(n/2);
      n2 = n - n1; j = n1+1;
      [Y1,T1,R1]= recqr(A(1:m,1:n1));
      B = A(1:m,j:n) - (Y1*T1')*(Y1'*A(1:m,j:n));
      [Y2,T2,R2] = recqr(B(j:m,1:n2));
      R = [R1, B(1:n1,1:n2); zeros(n-n1,n1), R2];
      Y2 = [zeros(n1,n2); Y2];
      Y = [Y1, Y2];
      T = [T1, -T1*(Y1'*Y2)*T2; zeros(n2,n1), T2];
    end
%
```

The algorithm uses the function house(a) to compute a Householder transformation $P = I - +\tau\, uu^T$, such that $Pa = \sigma\, e1$, $\sigma = -\text{sign}\,(a_1)\|a\|_2$. A serious defect of this algorithm is the overhead in storage and operations caused by the T matrices. In the partitioned algorithm n/nb T-matrices of size we formed and stored, giving a storage overhead of $\frac{1}{2}n \cdot nb$. In the recursive QR algorithm in the end a T-matrix of size $n \times n$ is formed and stored, leading to a much too large storage and operation overhead. Therefore a better solution is to use a hybrid between the partitioned and the recursive algorithm, where the recursive QR algorithm is used to factorize the blocks in the partitioned algorithm.

8.3.6 Least Squares Problems by QR Factorization

We now show how to use the QR factorization to solve the linear least squares problem (8.1.1).

Theorem 8.3.6.

Let the QR factorization of $A \in \mathbf{R}^{m\times n}$ with $\text{rank}\,(A) = n \le m$ be given by (8.3.39). *Then the unique solution x to $\min_x ||Ax - b||_2$ and for the corresponding residual vector r are given by*

$$x = R^{-1}c_1, \quad c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = Q^T b, \quad r = Q \begin{pmatrix} 0 \\ c_2 \end{pmatrix}, \tag{8.3.60}$$

and hence $\|r\|_2 = \|c_2\|_2$.

Proof. Since Q is orthogonal we have

$$\|Ax-b\|_2^2 = \left\|Q^T(Ax-b)\right\|_2^2 = \left\|\begin{pmatrix} Rx \\ 0 \end{pmatrix} - \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}\right\|_2^2 = \|Rx-c_1\|_2^2 + \|c_2\|_2^2.$$

Obviously the minimum residual norm $\|c_2\|_2$ is obtained by taking $x = R^{-1}c_1$. With c defined by (8.3.60) and using the orthogonality of Q we have

$$b = QQ^Tb = Q_1c_1 + Q_2c_2 = Ax + r$$

which shows the formula for r. □

By Theorem 8.3.6, when R and $H_1, H_2, \ldots, H_n$ have been computed by Algorithm 8.3.5 the least squares solution x and residual r can be computed from

$$\begin{aligned} n\{\ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} &= H_n \cdots H_2H_1b, \qquad Rx = c_1, \\ r &= H_1 \cdots H_{n-1}H_n \begin{pmatrix} 0 \\ c_2 \end{pmatrix}, \end{aligned} \tag{8.3.61}$$

and $\|r\|_2 = \|c_2\|_2$. Note that the matrix Q should not be explicitly formed.

When $\operatorname{rank}(A) = m \le n$, i.e., the matrix A has full row rank, the QR factorization of A^T (which is equivalent to the LQ factorization of A) can be used to solve the minimum norm problem (8.1.2).

Theorem 8.3.7.
Let $A \in \mathbf{R}^{m\times n}$ with $\operatorname{rank}(A) = m$, have the LQ factorization

$$A = \begin{pmatrix} L & 0 \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix}, \quad Q_1 \in \mathbf{R}^{n\times m},$$

Then the general solution to the underdetermined system $Ax = b$ is

$$x = Q_1y_1 + Q_2y_2, \qquad y_1 = L^{-1}b \tag{8.3.62}$$

where y_2 is arbitrary. The minimum norm solution is obtained by taking $y_2 = 0$,

$$x = Q_1L^{-1}b. \tag{8.3.63}$$

Proof. Since $A = \begin{pmatrix} L & 0 \end{pmatrix} Q^T$ the system $Ax = b$ can be written

$$\begin{pmatrix} L & 0 \end{pmatrix} y = b, \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = Q^Tx.$$

L is nonsingular, and thus y_1 is determined by $Ly_1 = b$. The vector y_2 can be chosen arbitrarily. Further, since $\|x\|_2 = \|Qy\|_2 = \|y\|_2$ the minimum norm solution is obtained by taking $y_2 = 0$. □

The operation count $mn^2 - n^3/3$ for the QR method can be compared with that for the method of normal equations, which requires $\frac{1}{2}(mn^2 + n^3/3)$ multiplications. Hence, for $m = n$ both methods require the same work but for $m \gg n$ the QR method is twice as expensive. To compute c by (8.3.61) requires $(2mn - n^2)$ multiplications, and thus to compute the solution for each new right hand side takes only $(2mn - n^2/2)$ multiplications. The Householder QR algorithm, and the resulting method for solving the least squares problem are backwards stable, both for x and r, and the following result holds.

Theorem 8.3.8.

*Let $\bar{R}$ denote the computed R. Then there exists an exactly orthogonal matrix $\tilde{Q} \in \mathbf{R}^{m\times m}$ (*not *the matrix corresponding to exact computation throughout) such that*

$$A + E = \tilde{Q}\begin{pmatrix} \bar{R} \\ 0 \end{pmatrix}, \qquad \|E\|_F \le cu\|A\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $c = 6n(m - n/2 + 7)$, and u is the machine precision. Further, the computed solution $\bar{x}$ is the exact solution of a slightly perturbed least squares problem

$$\min_x \|(A + \delta A)x - (b + \delta b)\|_2,$$

where the perturbation can be bounded in norm by

$$\|\delta A\|_F \le cu\|A\|_F, \qquad \|\delta b\|_2 \le cu\|b\|_2, \tag{8.3.64}$$

Proof. See Higham [33, Theorem 19.5]. □

A method combining LU factorization and orthogonalization can be developed by solving the least squares problem in (8.2.22) by an orthogonal reduction of L to lower triangular form. The solution is then obtained by solving $\tilde{L}y = c_1$ by forward substitution, where

$$L = Q\begin{pmatrix} \tilde{L} \\ 0 \end{pmatrix}, \qquad Q^T\Pi_1 b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

In **Cline's method**, Householder transformations can be used to perform this reduction of L. The kth Householder transformation P_k is chosen to affect only rows $k, n+1, \ldots, m$ and zero elements in column k below row n. The total number of flops required for computing the least squares solution x by Cline's method is about $n^2(\frac{3}{2}m - \frac{7}{6}n)$ flops. Since the method of normal equations using the Cholesky factorization on A^TA requires $n^2(\frac{1}{2}m + \frac{1}{6}n)$ flops Cline's method uses fewer operations if $m \le \frac{4}{3}n$. Hence for slightly overdetermined least squares problems, the elimination method combined with Householder transformations is very efficient.

A version solving (8.2.22) with the MGS method has been analyzed by Plemmons [50, 1974]. If the lower triangular structure of L is taken advantage of then

this method requires $n^2(\frac{3}{2}m - \frac{5}{6}n)$ flops, which is slightly more than Cline's variant. Similar methods for the underdetermined case ($m < n$) based on the LU decomposition of A have been studied by Cline and Plemmons [17, 1976].

An algorithm similar to Algorithm 8.3.5, but using Givens rotations, can easily be developed. The greater flexibility of Givens rotations can be taken advantage of when the matrix A is structured or sparse; see, e.g., Problem 3, where the QR factorization of a Hessenberg matrix is considered.

Peters and Wilkinson commented in 1970: "*Evidence is accumulating that the modified Gram–Schmidt method gives better results than Householder. . . . The reasons for this phenomenon appear not to have been elucidated yet.*" A key observation for understanding the good numerical properties of the modified Gram–Schmidt algorithm is that it can be interpreted as Householder QR factorization applied to the matrix A *augmented with a square matrix of zero elements on top.* These two algorithms are not only mathematically but also *numerically* equivalent. In the MGS method the columns are transformed by

$$a_j^{(k+1)} = M_k a_j^{(k)}, \qquad M_k = I - q_k q_k^T,$$

where M_k is the orthogonal projection onto the complement of q_k. In the Householder method one computes the factorization

$$P^T \begin{pmatrix} 0 \\ A \end{pmatrix} = \begin{pmatrix} R \\ 0 \end{pmatrix}, \qquad P^T = P_n \cdots P_2 P_1,$$

$$P_k = I - v_k v_k^T, \qquad v_k = \begin{pmatrix} -e_k \\ q_k \end{pmatrix}.$$

Here $\|v_k\|_2^2 = 2$, and hence P_k is a Householder reflection. Because of the special structure of the augmented matrix the vectors v_k have a special form. Since the first n rows are initially zero, the scalar products of the vector v_k with later columns will only involve q_k, and it can be verified that the quantities r_{kj} and q_k are *numerically* equivalent to the quantities computed in the modified Gram–Schmidt method.

8.3.7 Condition and Error Estimation

Using the above pivoting strategy, a lower bound for $\kappa(A) = \kappa(R)$ can be obtained from the diagonal elements of R. We have $|r_{11}| \le \sigma_1 = \|R\|_2$, and since the diagonal elements of R^{-1} equal r_{ii}^{-1}, $i = 1, \ldots, n$, it follows that $r_{nn}^{-1}| \le \sigma_n^{-1} = \|R^{-1}\|_2$, provided $r_{nn} \neq 0$. Combining these estimates we obtain the *lower bound*

$$\kappa(A) = \sigma_1/\sigma_n \ge |r_{11}/r_{nn}| \tag{8.3.65}$$

Although this may considerably underestimate $\kappa(A)$, it has proved to give a fairly reliable estimate in practice. Extensive numerical testing has shown that (8.3.65) usually underestimates $\kappa(A)$ only by a factor of 2–3, and seldom by more than 10.

When column pivoting has not been performed, the above estimate of $\kappa(A)$ is not reliable. Then a condition estimator similar to that described in Sec. 7.6.5 can be used. Let u be a given vector and define v and w from

$$R^T v = u, \qquad Rw = v.$$

We have $w = R^{-1}(R^{-T}u) = (A^TA)^{-1}u$ so this is equivalent to one step of inverse iteration with A^TA, and requires about $0(n^2)$ multiplications. Provided that u is suitably chosen (cf. Sec. 7.6.5)

$$\sigma_n^{-1} \approx \|w\|_2/\|v\|_2$$

will usually be a good estimate of σ_n^{-1}. We can also take u as a random vector and perform and 2–3 steps of inverse iteration. This condition estimator will usually detect near rank deficiency even in the case when this is not revealed by a small diagonal element in R.

More reliable estimates can be based on the componentwise error bound (8.1.33) given in Sec. 8.1.5. This estimate has the form

$$\|\delta x\|_\infty \leq \omega(\| \, |B_1|g_1\|_\infty + \| \, |B_2|g_2\|_\infty), \tag{8.3.66}$$

where

$$B_1 = A^\dagger, \quad g_1 = |b| + |A||x|, \quad B_2 = (A^TA)^{-1}, \quad g_2 = |A^T||r|. \tag{8.3.67}$$

Consider now a general expression of the form $\| \, |B^{-1}|d\|_\infty$, where $d > 0$ is a known nonnegative vector. Writing $D = \operatorname{diag}(d)$ and $e = (1, 1, \ldots, 1)$, we have[3]

$$\| \, |B^{-1}|d\|_\infty = \| \, |B^{-1}|De\|_\infty = \| \, |B^{-1}D|e\|_\infty = \| \, |B^{-1}D|\|_\infty = \|B^{-1}D\|_\infty. \tag{8.3.68}$$

Hence the problem is equivalent to that of estimating $\|C\|_\infty$, where $C = B^{-1}D$. There are algorithms that produce reliable order-of-magnitude estimates of $\|C^T\|_1 = \|C\|_\infty$ using only a few matrix-vector products of the form Cx and C^Ty for some carefully selected vectors x and y. Since these are rather tricky we will not describe them in detail here. An excellent discussion is given in Higham [33, Chapter 15].

If A has full rank and $A = QR$ then $A^\dagger = R^{-1}Q^T$ and $(A^\dagger)^T = QR^{-T}$. Hence the required products can be computed inexpensively.

Review Questions

1 Let $w \in \mathbf{R}^n$, $\|w\|_2 = 1$. Show that $I - 2ww^T$ is orthogonal. What is the geometrical significance of the matrix $I - 2ww^T$? Give the eigenvalues and eigenvectors of these matrices.

2. Define a Givens transformations $G_{ij}(\phi) \in \mathbf{R}^{n\times n}$. Give a geometrical interpretations for the case $n = 2$.

3. Describe the difference between the classical and modified Gram–Schmidt methods for computing the factorization $A = Q_1R$. What can be said about the orthogonality of the computed matrix Q_1 for these two algorithms?

[3] This clever observation is due to Arioli, Demmel, and Duff [2].

4. Define the QR factorization of a matrix $A \in \mathbf{R}^{m\times n}$, in the case that $\operatorname{rank}(A) = n \le m$. What is its relation to the Cholesky factorization of A^TA?

Problems

1. Compute using Householder reflectors H_1, H_2, the factorization

$$Q^TA = H_2H_1A = \begin{pmatrix} R \\ 0 \end{pmatrix}, \qquad A = (a_1, a_2) = \begin{pmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{pmatrix},$$

to four decimal places

2. Solve the least squares problem $\min_x \|Ax - b\|_2$, where

$$\begin{pmatrix} \sqrt{2} & 0 \\ 1 & -1 \\ 1 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

using a QR factorization computed with Givens transformation;

3. Suppose the square root free version of modified Gram–Schmidt is used to compute the factorization $A = \tilde{Q}_1\tilde{R}$. Modify Algorithm 8.3.2 for computing the least squares solution and residual from this factorization.

4. Describe in detail how to compute the QR factorization of a Hessenberg matrix $H \in \mathbf{R}^{n\times n}$ using Givens transformations. For $n = 5$ such a matrix has the form

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} \\ & h_{32} & h_{33} & h_{34} & h_{35} \\ & & h_{43} & h_{44} & h_{45} \\ & & & h_{54} & h_{55} \end{pmatrix}.$$

Approximately how many multiplications are needed for general n?

5. (a) If the matrix Q in the QR factorization is explicitly required in the Householder algorithm it can be computed by setting $Q^{(n)} = I_m$, and computing $Q = Q^{(0)}$ by backward recursion

$$Q^{(k-1)} = H_kQ^{(k)}, \quad k = n : -1 : 1.$$

Show that if advantage is taken of the property that $H_k = \operatorname{diag}(I_{k-1}, \tilde{H}_k)$ this accumulation requires $2(m^2n - mn^2 + n^3/3)$ flops. What is the corresponding operation count if forward recursion is used?

(b) Show how we can compute

$$Q_1 = Q\begin{pmatrix} I_n \\ 0 \end{pmatrix}, \qquad Q_2 = Q\begin{pmatrix} 0 \\ I_{m-n} \end{pmatrix}$$

separately in $mn^2 - n^3/3$ and $2m^2n - 3mn^2 + n^3$ multiplications, respectively.

6. Let $Q = Q_1 = (q_1, q_2, \ldots, q_n) \in \mathbf{R}^{n\times n}$ be a real orthogonal matrix.
(a) Determine a reflector $H_1 = I - 2v_1v_1^T$, such that $H_1q_1 = e_1 = (1, 0, \ldots, 0)^T$, and show that $H_1Q_1 = Q_2$ has the form

$$Q_2 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \tilde{Q}_2 & \\ 0 & & & \end{pmatrix},$$

where $\tilde{Q}_2 = (\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_n) \in \mathbf{R}^{(n-1)\times(n-1)}$ is a real orthogonal matrix.
(b) Show, using the result in (a), that Q can be transformed to diagonal form with a sequence of orthogonal transformations

$$H_{n-1} \cdots H_2H_1Q = \text{diag}\,(1, \ldots, 1, \pm 1).$$

7. An orthogonal matrix Q such that $\det(Q) = 1$ is called a rotation matrix. Show that any rotation matrix $Q \in \mathbf{R}^{3\times3}$ can be written as a product of three Givens rotations

$$Q = G_{23}(\phi)G_{12}(\theta)G_{23}(\psi).$$

The three angles ϕ, θ, and ψ are called the **Euler angles**.
Hint: Consider the QR factorization of Q.

8. Test the recursive QR algorithm `recqr(A)` given in Sec. sec8.3.6 on some matrices. Check that you obtain the same result as from the built-in function qr(A).

8.4 Rank Deficient and Ill-Posed Problems

8.4.1 Regularized Least Squares Problems

In solving linear systems and linear least squares problems failure to detect ill-conditioning and possible rank deficiency in A can lead to a meaningless solution of very large norm, or even to breakdown of the numerical algorithm. In this section we discuss how to assign a **numerical rank** to a matrix and how algorithms should be modified to cope with rank deficiency and ill-conditioning.

Example 8.4.1. Consider an example based on the integral equation of the first kind

$$\int_1^1 k(s,t)f(s)ds = g(t), \quad k(s,t) = e^{-(s-t)^2},$$

on $-1 \le t \le 1$. To compute $g(t)$ given $f(s)$ is well conditioned problem. However, the inverse problem of reconstructing $f(s)$ given $g(t)$ is a very ill-conditioned problem.

The equation can be discretized using a uniform mesh on [-1,1] and the trapezoidal rule, giving a finite-dimensional linear system $Kf = g$, where $K \in \mathbf{R}^{n\times n}$, and $f, g \in \mathbf{R}^n$. For $n = 100$ the singular values σ_k of the matrix K are displayed in

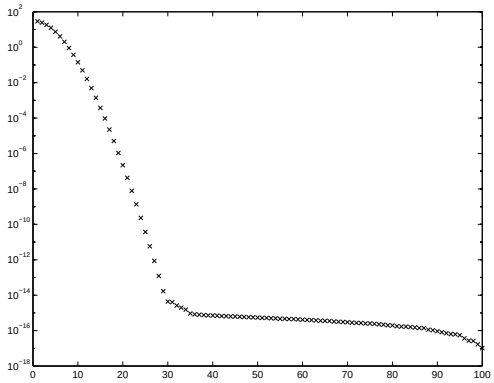

Figure 8.4.1. *Singular values of the matrix K.*

logarithmic scale in Figure 8.4.1. Note that for $k > 30$ all σ_k are close to roundoff level, so the numerical rank of K certainly is smaller than 30. This means that the linear system $Kf = g$ is numerically *under-determined* and has a meaningful solution only for special right hand sides g.

The choice of the parameter δ in Definition 8.1.15 is not always an easy matter. If the errors in a_{ij} satisfy $|e_{ij}| \leq \epsilon$, for all i, j, an appropriate choice is $\delta = (mn)^{1/2}\epsilon$. On the other hand, if the absolute size of the errors e_{ij} differs widely, then Definition 8.1.15 is not appropriate. One could then scale the rows and columns of A so that the magnitude of the errors become nearly equal. (Note that any such diagonal scaling $D_r A D_c$ will induce the same scaling $D_r E D_c$ of the error matrix.)

We now consider solving the linear least squares problem

$$\min_x \|Ax - b\|_2, \tag{8.4.1}$$

where the matrix A is ill-conditioned and possibly rank deficient. If A has numerical rank equal to $k < n$, we can get a more stable approximative solution by discarding terms in the expansion (8.1.10) corresponding to singular values smaller or equal to δ, and take the solution as the **truncated SVD (TSVD) solution**

$$x(\delta) = \sum_{\sigma_i > \delta} \frac{c_i}{\sigma_i} v_i. \tag{8.4.2}$$

If $\sigma_k > \delta \geq \sigma_{k+1}$ then the TSVD solution is $x(\delta) = A_k^\dagger b$ and solves the related least squares problem

$$\min_x \|A_k x - b\|_2, \qquad A_k = \sum_{\sigma_i > \delta} \sigma_i u_i v_i^T,$$

where A_k is the best rank k approximation of A. We have

$$\|A - A_k\|_2 = \|AV_2\|_2 \leq \delta, \quad V_2 = (v_{k+1}, \ldots, v_n).$$

In general the most reliable way to determine an approximate pseudo-inverse solution of a numerically rank deficient least squares problems is by first computing the SVD of A and then using an appropriate truncated SVD solution (8.4.2). However, this is also an expensive method. In practice the QR factorization often works as well, provided that some form of **column pivoting** is carried out.

An alternative to the truncated SVD (**TSVD**) solution is to consider the **regularized** problem

$$\min_x \ \|Ax - b\|_2^2 + \mu^2 \|Dx\|_2^2, \tag{8.4.3}$$

where $D = \operatorname{diag}(d_1, \ldots, d_n) > 0$ is a positive diagonal matrix. The problem (8.4.3), also called a **damped** least squares problem, is equivalent to the least squares problem

$$\min_x \left\| \begin{pmatrix} A \\ \mu D \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2, \tag{8.4.4}$$

where the matrix A has been modified by appending the matrix μD. When $\mu > 0$ this problem is always of full column rank and has a unique solution. (Often d_j is taken to be proportional to the 2-norm of the jth column in A.)

The solution to problem (8.4.3) satisfies the normal equations

$$(A^T A + \mu^2 D^2)x = A^T b.$$

However, from the formulation (8.4.4) it is seen that the solution can also be obtained from the QR factorization

$$\begin{pmatrix} A \\ \mu D \end{pmatrix} = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \tag{8.4.5}$$

which can be computed by some of the algorithms described before. The special structure can be taken advantage of. For example, in the Householder QR factorization the shape of the transformed matrix after $k = 2$ steps is as follows ($m = n = 4$):

$$\begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & + & + \\ & 0 & + & + \\ & & \times & \\ & & & \times \end{pmatrix}$$

Notice that the first two rows of D have filled in, but the remaining rows of D are still not touched. For each step $k = 1, \ldots, n$ there are m elements in the current column to be annihilated. Therefore the operation count for the Householder QR factorization will increase with $n^3/3$ to mn^2 flops. A similar increase in operations occurs in Givens or MGS QR factorizations. If $A = R$ already is in upper triangular form then the flop count for the reduction is reduced to approximately $n^3/3$ (cf. Problem 1b).

If $D = I$ the singular values of the modified matrix in (8.4.4) are equal to $\tilde{\sigma}_i = (\sigma_i^2 + \mu^2)^{1/2}$, $i = 1, \ldots, n$. In this case the solution can be expressed in terms of the SVD as

$$x(\mu) = \sum_{i=1}^{n} f_i \frac{c_i}{\sigma_i} v_i, \quad f_i = \frac{\sigma_i^2}{\sigma_i^2 + \mu^2}. \tag{8.4.6}$$

The quantities f_i are often called **filter factors**. Notice that as long as $\mu \ll \sigma_i$ we have $f_i \approx 1$, and if $\mu \gg \sigma_i$ then $f_i \ll 1$. This establishes a relation to the truncated SVD solution (8.4.2) which corresponds to a filter factor which is a step function $f_i = 1$ if $\sigma_i > \delta$ and $f_i = 0$ otherwise.

Note that the regularized problem (8.4.3) can be used also when $m < n$ (i.e., when A has fewer rows than columns). However, in this case it may be better to consider the regularized problem

$$\min \left\| \begin{pmatrix} x \\ z \end{pmatrix} \right\|_2^2, \quad \text{subject to} \quad (A \ \mu D) \begin{pmatrix} x \\ z \end{pmatrix} = b. \tag{8.4.7}$$

The solution of this problem can be written $x = A^T y$, $z = (\mu D)^{-1}(b - Ax)$, where y satisfies the system of normal equations

$$(AA^T + \mu^2 D^2)y = b.$$

Using Theorem 8.3.7, a method for solving problem (8.4.7) is obtained which uses the QR factorization of the matrix $(\mu D \ A)^T$, which can be computed in $m^2 n$ operations. Surprisingly, when $D = I$ the two problems (8.4.3) and (8.4.7) are equivalent. To see this set note that since $z = (\mu)^{-1}(b - Ax)$, both problems (8.4.4) and (8.4.7) are equivalent to

$$\min_x \left\{ \|r\|_2^2 + \mu^2 \|x\|_2^2 \right\}, \quad r = b - Ax.$$

Even with regularization we may not be able to compute the solution of an ill-conditioned problem with the accuracy that the data allows. In those cases it is possible to improve the solution by the following **iterated regularization** scheme. Take $x^{(0)} = 0$, and compute a sequence of approximate solutions by

$$x^{(q+1)} = x^{(q)} + \delta x^{(q)},$$

where $\delta x^{(q)}$ solves the least squares problem

$$\min_{\delta x} \left\| \begin{pmatrix} A \\ \mu I \end{pmatrix} \delta x - \begin{pmatrix} r^{(q)} \\ 0 \end{pmatrix} \right\|_2, \qquad r^{(q)} = b - Ax^{(q)}. \tag{8.4.8}$$

This iteration may be implemented very effectively since only the QR factorization (8.4.5) (with $D = I$) is needed. The convergence of iterated regularization can be expressed in terms of the SVD of A.

$$x^{(q)}(\mu) = \sum_{i=1}^{n} f_i^{(q)} \frac{c_i}{\sigma_i} v_i, \quad f_i^{(q)} = 1 - \Big(\frac{\mu^2}{\sigma_i^2 + \mu^2} \Big)^q. \tag{8.4.9}$$

Thus for $q = 1$ we have the standard regularized solution and as $q \to \infty$ $x^{(q)} \to A^\dagger b$.

8.4.2 QR Factorization and Rank Deficient Matrices

Although any matrix $A \in R^{m\times n}$ has a QR factorization, the following example shows that this may not always be useful when $\operatorname{rank}(A) < n$:

Example 8.4.2.
For any c and s such that $c^2 + s^2 = 1$ we have

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} 0 & s \\ 0 & c \end{pmatrix} = QR.$$

Here $\operatorname{rank}(A) = 1 < 2 = n$. Note that the columns of Q no longer provide any information about an orthogonal basis for $R(A)$ and its complement.

We now indicate how the QR factorization should be modified in the rank deficient case.

Theorem 8.4.1.
Given $A \in \mathbf{R}^{m\times n}$ with $\operatorname{rank}(A) = r \le \min(m, n)$ there is a permutation matrix Π and an orthogonal matrix $Q = \in \mathbf{R}^{m\times m}$ such that

$$A\Pi = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} \tag{8.4.10}$$

where $R_{11} \in \mathbf{R}^{r\times r}$ is upper triangular with positive diagonal elements.

Proof. Since $\operatorname{rank}(A) = r$, we can always choose a permutation matrix Π such that $A\Pi = (A_1, A_2)$, where $A_1 \in \mathbf{R}^{m\times r}$ has linearly independent columns. Then A_1 has a QR factorization and we can write

$$Q^T A\Pi = (\, Q^T A_1 \quad Q^T A_2 \,) = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

where R_{11} has positive diagonal elements. From $\operatorname{rank}(Q^T A\Pi) = \operatorname{rank}(A) = r$ it follows that $R_{22} = 0$, since otherwise $Q^T A\Pi$ would have more than r linearly independent rows. □

Note that it is not required that $m \ge n$ in Theorem 8.4.1. The factorization is not unique, since there may be several ways to choose the permutation Π. Pivoting strategies for determining a suitable Π will be discussed later in this section.

The factorization (8.4.10) can be used to solve rank deficient linear least squares problems. To simplify notations we assume in the following that $\Pi = I$. (This is no restriction since the column permutation of A can always be assumed to have been carried out in advance.) Using the invariance of the l_2-norm problem (8.1.1) becomes

$$\min_x \left\| \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \right\|_2,$$

where x and c have been partitioned conformally. Since R_{11} is nonsingular the first r equations can be satisfied for any x_2 by taking x_1 to be the solution to $R_{11}x_1 = c_1 - R_{12}x_2$. Hence the general least squares solutions can be written

$$x_1 = R_{11}^{-1}(c_1 - R_{12}x_2) = x_b - C_1x_2, \tag{8.4.11}$$

where x_2 is arbitrary and

$$d = R_{11}^{-1}c_1, \qquad C = R_{11}^{-1}R_{12}. \tag{8.4.12}$$

Here C can be computed by solving $n - r$ triangular systems $R_{11}C = R_{12}$, which requires $r^2(n-r)/2$ multiplications.

Taking $x_2 = 0$ we we obtain a particular solution $x_1 = d$ with at most $r = \text{rank}(A)$ nonzero components Any solution x such that Ax only involves at most r columns of A, is called a **basic least squares solution**. Such a solution is appropriate when we want to fit a vector b of observations using *as few columns of* A as possible. It is not unique and depends on the initial column permutation.

We now show how the pseudo-inverse solution can be computed using the factorization (8.4.10). Then we want to choose x_2 so that $\|x\|_2$ is minimized. From (8.4.11) it follows that this is achieved by solving the linear least squares problem for x_2

$$\min \left\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right\|_2 = \min_{x_2} \left\| \begin{pmatrix} d \\ 0 \end{pmatrix} - \begin{pmatrix} C \\ -I_{n-r} \end{pmatrix} x_2 \right\|_2. \tag{8.4.13}$$

Note that this problem always has a unique solution x_2 and that the pseudo-inverse solution $x = A^{\dagger}b$ equals *the residual* of the problem.

To compute x_2 we can form and solve the normal equations

$$(I + CC^T)x_2 = C^T d. \tag{8.4.14}$$

Alternatively we can use Householder QR factorization

$$Q_C^T \begin{pmatrix} C \\ I_{n-r} \end{pmatrix} = \begin{pmatrix} R_C \\ 0 \end{pmatrix}, \qquad Q_C^T \begin{pmatrix} d \\ 0 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix},$$

taking the special structure into account, to obtain x_2 from $R_C x_2 = d_1$.

We have

$$A \begin{pmatrix} C \\ -I_{n-r} \end{pmatrix} = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_{11}^{-1}R_{12} \\ -I_{n-r} \end{pmatrix} = 0,$$

from which it follows that the nullspace of A is given by

$$\mathcal{N}(A) = \mathcal{R}(W), \quad W = \begin{pmatrix} C \\ -I_{n-r} \end{pmatrix}. \tag{8.4.15}$$

By Theorem 8.1.7 the pseudo-inverse solution is the unique least squares solution which satisfies $x \perp \mathcal{N}(A)$. Hence it can be obtained by Gram–Schmidt orthogonalization applied to

$$\begin{pmatrix} C & d \\ I_{n-r} & 0 \end{pmatrix}. \tag{8.4.16}$$

It is possible to carry the factorization one step further to give the related **complete QR factorization** of A.

Theorem 8.4.2.

Given $A \in \mathbf{R}^{m\times n}$ with $\operatorname{rank}(A) = r \le \min(m,n)$. Then there are orthogonal matrices $Q = (Q_1, Q_2) \in \mathbf{R}^{m\times m}$, and $V = (V_1, V_2) \in \mathbf{R}^{n\times n}$ such that

$$A = Q \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} V^T \tag{8.4.17}$$

where $R \in \mathbf{R}^{r\times r}$ is upper triangular with positive diagonal elements. The pseudo-inverse of A is then given by

$$A^\dagger = V \begin{pmatrix} R^{-1} & 0 \\ 0 & 0 \end{pmatrix} Q^T = V_1 R^{-1} Q_1^T. \tag{8.4.18}$$

Proof. Starting from the factorization in (8.4.10) we can determine a sequence of Householder matrices such that

$$(\, R_{11} \quad R_{12}\,) P_r \cdots P_1 = (\, R \quad 0\,).$$

Here P_k, $k = r, r-1, \ldots, 1$, is constructed to zero elements in row k and only affect columns $k, r+1, \ldots, n$. These transformations require $r^2(n-r)$ multiplications. Then (8.4.17) holds with $V = \Pi P_1 \cdots P_r$. Using the orthogonal invariance of the l_2-norm it follows that $x = V_1 R^{-1} Q_1^T b$ is the minimum norm solution of the least squares problem (8.1.1). Since the pseudo-inverse is uniquely defined by this property, cf. Theorem 8.1.5, the last assertion follows. □

8.4.3 Rank Revealing QR Factorization

In Sec. 8.3.5 we studied the pivoted QR factorization. It was shown that if the pivot column in each step of the reduction was chosen as a column of largest norm in the remaining part, then we have the inequalities

$$r_{kk}^2 \ge \sum_{i=k}^{j} r_{ij}^2, \quad j = k+1, \ldots, n. \tag{8.4.19}$$

in particular it holds that $|r_{kk}| \ge |r_{kj}|$, $j > k$ and the diagonal elements form a non-increasing sequence, $|r_{11}| \ge |r_{22}| \ge \cdots \ge |r_{nn}|$.

Taking $x = e_1$ in $\sigma_1 = \max_{\|x\|=1} \|Ax\|_2$ we find that the lower bound $|r_{11}| \le \sigma_1(R)$ for the largest singular value σ_1. The matrix R^{-1} has diagonal elements $1/r_{kk}$ and singular values $1/\sigma_k(A)$. Hence we also have the inequality $\sigma_n \le |r_{nn}|$.

For a triangular matrix satisfying (8.4.19) we also have the upper bound

$$\sigma_1(R) = \|R\|_2 \le \Big(\sum_{i\le j} r_{ij}^2 \Big)^{1/2} \le \sqrt{n} r_{11},$$

and hence $\sigma_1(R) \le n^{1/2} r_{11}$. Using the interlacing property of singular values (Theorem 8.1.13), a similar argument gives the upper bounds

$$\sigma_k(R) \le (n-k+1)^{1/2} |r_{k,k}|, \quad 1 \le k \le n. \tag{8.4.20}$$

If after k steps in the pivoted QR factorization it holds that

$$|r_{k,k}| \leq (n-k+1)^{-1/2}\delta,$$

then $\sigma_k(A) = \sigma_k(R) \leq \delta$, and A has numerical rank at most equal to $k-1$, and we should terminate the algorithm. Unfortunately, the converse is not true, i.e., the rank is not always revealed by a small element $|r_{kk}|$, $k \leq n$. Let R be an upper triangular matrix whose elements satisfy (8.3.52). The best known *lower* bound for the smallest singular value is

$$\sigma_n \geq 3|r_{nn}|/\sqrt{4^n + 6n - 1} \geq 2^{1-n}|r_{nn}|. \tag{8.4.21}$$

(For a proof see Lawson and Hanson [38, Ch. 6].)

The lower bound in (8.4.21) can almost be attained as shown in the example below due to Kahan. Then the pivoted QR factorization may not reveal the rank of A.

Example 8.4.3. Consider the upper triangular matrix

$$R_n = \operatorname{diag}(1, s, s^2, \ldots, s^{n-1}) \begin{pmatrix} 1 & -c & -c & \ldots & -c \\ & 1 & -c & \ldots & -c \\ & & 1 & & \vdots \\ & & & \ddots & -c \\ & & & & 1 \end{pmatrix}, \qquad s^2 + c^2 = 1.$$

It can be verified that the elements in R_n satisfies the inequalities in (8.4.21), and that R_n is invariant under QR factorization with column pivoting. For $n = 100$, $c = 0.2$ the last diagonal element of R is $r_{nn} = s^{n-1} = 0.820$. This can be compared with the smallest singular value which is $\sigma_n = 0.368 \cdot 10^{-8}$. If the columns are reordered as $(n, 1, 2, \ldots, n-1)$ and the rank is revealed from the pivoted QR factorization!

The above example did inspire research into alternative column permutation strategies. The following theorem, which we state without proof, shows that a column permutation Π can always be found so that the numerical rank of A is revealed by the QR factorization of $A\Pi$.

Theorem 8.4.3. (H. P. Hong and C. T. Pan [1992].)

Let $A \in \mathbf{R}^{m\times n}$, $(m \geq n)$, and r be a given integer $0 < r < n$. Then there exists a permutation matrix Π_r, such that the QR factorization has the form

$$Q^T A\Pi_r = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}, \tag{8.4.22}$$

with $R_{11} \in \mathbf{R}^{r\times r}$ upper triangular, $c = \sqrt{r(n-r) + \min(r, n-r)})$, and

$$\sigma_{\min}(R_{11}) \geq \frac{1}{c}\sigma_r(A), \qquad \sigma_{\max}(R_{22}) \leq c\sigma_{r+1}(A). \tag{8.4.23}$$

Note that the bounds in this theorem are much better than those in (8.4.21).

From the interlacing properties of singular values (Theorem 8.1.13) it follows by induction that for any factorization of the form (8.4.22) we have the inequalities

$$\sigma_{\min}(R_{11}) \le \sigma_r(A), \qquad \sigma_{\max}(R_{22}) \ge \sigma_{r+1}(A). \tag{8.4.24}$$

Hence to achieve (8.4.23) we want to choose the permutation Π to maximize $\sigma_{\min}(R_{11})$ and simultaneously minimize $\sigma_{\max}(R_{22})$. These two problems are in a certain sense dual; cf. Problem 2.

Assume now that A has a well defined numerical rank $r < n$, i.e.,

$$\sigma_1 \ge \ldots \ge \sigma_r \gg \delta \ge \sigma_{r+1} \ge \ldots \ge \sigma_n.$$

Then the above theorem says that if the ratio σ_k/σ_{k+1} is sufficiently large then there is a permutation of the columns of A such that the rank of A is revealed by the QR factorization. Unfortunately, to find such a permutation may be a hard problem. The naive solution, to try all possible permutations, is not feasible since the cost prohibitive—it is exponential in the dimension n.

Many other pivoting strategies for computing rank revealing QR factorizations have been proposed. A strategy by T. F. Chan [14] makes use of approximate right singular vectors of A, which can be determined by inverse iteration (see Sec. 9.4.3). In case $r = n-1$, the column permutation Π is constructed from an approximation to the right singular vector corresponding to the smallest singular value σ_n.

8.4.4 The URV and ULV decompositions

In signal processing problems it is often the case that one wants to determine the rank of A as well as the range (signal subspace) and null space of A. Since the data analyzed arrives in real time it is necessary to update an appropriate matrix decompositions at each time step. For such applications the SVD has the disadvantage that it cannot in general be updated in less than $\mathcal{O}(n^3)$ operations, when rows and columns are added or deleted to A. Although the RRQR decomposition can be updated, it is less suitable in applications where a basis for the approximate null space of A is needed, since the matrix W in (8.4.15) cannot easily be updated.

For this reason we introduce the **URV decomposition**

$$A = URV^T = (\,U_1 \quad U_2\,) \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}, \tag{8.4.25}$$

where U and V are orthogonal matrices, $R_{11} \in \mathbb{R}^{k\times k}$, and

$$\sigma_k(R_{11}) \ge \frac{1}{c}\sigma_k, \qquad \left(\|R_{12}\|_F^2 + \|R_{22}\|_F^2\right)^{1/2} \le c\sigma_{k+1}. \tag{8.4.26}$$

Note that here both submatrices R_{12} and R_{22} have small elements.

From (8.4.25) we have

$$\|AV_2\|_2 = \left\| \begin{pmatrix} R_{12} \\ R_{22} \end{pmatrix} \right\|_F \le c\sigma_{k+1},$$

and hence the orthogonal matrix V_2 can be taken as an approximation to the numerical null space $\mathcal{N}_k$.

Algorithms for computing an URV decomposition start with an initial QR decomposition, followed by a rank revealing stage in which singular vectors corresponding to the smallest singular values of R are estimated. Assume that w is a unit vector such that $\|Rw\| = \sigma_n$. Let P and Q be a orthogonal matrices such that $Q^T w = e_n$ and $P^T RQ = \hat{R}$ where $\hat{R}$ is upper triangular. Then

$$\|\hat{R}e_n\| = \|P^T RQQ^T w\| = \|P^T Rw\| = \sigma_n,$$

which shows that the entire last column in $\hat{R}$ is small. Given w the matrices P and Q can be constructed as a sequence of Givens rotations, Algorithms can also be given for updating an URV decomposition when a new row is appended.

Like the RRQR decompositions the URV decomposition yield approximations to the singular values. In [41] the following bounds are derived

$$f\sigma_i \le \sigma_i(R_{11}) \le \sigma_i, \quad i = 1 : r,$$

and

$$\sigma_i \le \sigma_{i-k}(R_{22}) \le \sigma_i/f, \quad i = r+1 : n,$$

where

$$f = \left(1 - \frac{\|R_{12}\|_2^2}{\sigma_{\min}(R_{11})^2 - \|R_{22}\|_2^2}\right)^{1/2}.$$

Hence the smaller the norm of the off-diagonal block R_{12}, the better the bounds will be. Similar bounds can be given for the angle between the range of V_2 and the right singular subspace corresponding to the smallest $n - r$ singular values of A.

An alternative decomposition that is more satisfactory for applications where an accurate approximate null space is needed, is the rank-revealing **ULV decomposition**

$$A = U \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} V^T. \tag{8.4.27}$$

where the middle matrix has lower triangular form. For this decomposition

$$\|AV_2\|_2 = \|L_{22}\|_F, \qquad V = (V_1,\ V_2),$$

and hence the size of $\|L_{21}\|$ does not adversely affect the null space approximation. On the other hand the URV decomposition usually gives a superior approximation for the numerical range space and the updating algorithm for URV is much simpler.

We finally mention that rank-revealing QR decompositions can be effectively computed only if the numerical rank r is either high, $r \approx n$ or low, $r \ll n$. The low rank case is discussed in [15]. Matlab templates for rank-revealing UTV decompositions are described in [22].

An advantage of the complete QR factorization of A is that V_2 gives an orthogonal basis for the nullspace $\mathcal{N}(A)$. This is often useful, e.g., in signal processing applications, where one wants to determine the part of the signal that corresponds to noise. The factorization (8.4.18) can be generalized to the case when A is only

numerically rank deficient in a similar way as done above for the QR factorization. The resulting factorizations have one of the forms

$$A = Q \begin{pmatrix} R & F \\ 0 & G \end{pmatrix} V^T \qquad A = Q \begin{pmatrix} R^T & 0 \\ F^T & G^T \end{pmatrix} V^T \tag{8.4.28}$$

where R is upper triangular and

$$\sigma_k(R) > \frac{1}{c}, \qquad (\|F\|_F^2 + \|G\|_F^2)^{1/2} \le c\sigma_{k+1}.$$

An advantage is that unlike the SVD it is possible to efficiently update the factorizations (8.4.28) when rows/columns are added/deleted.

8.4.5 Bidiagonal Decomposition and Least Squares

So far we have considered methods based on the QR factorization of A for solving least squares problems. It is possible to carry this reduction further using a two-sided orthogonal factorization.

Theorem 8.4.4.
Any matrix $A \in \mathbb{R}^{m\times n}$ can be decomposed as

$$A = UBV^T, \tag{8.4.29}$$

where B is a lower bidiagonal matrix and U and V are orthogonal matrices. In the nondegenerate case the decomposition is uniquely determined by $u_1 := Ue_1$, which can be chosen arbitrarily.

Note that, since $A^T = VB^TU^T$, it follows that an arbitrary matrix A can alternatively be transformed to *upper* bidiagonal form.

This decomposition is usually the first step in computing the SVD of A; see Sec. 9.7. It is also powerful tool for solving various least squares problems. We will give a constructive proof of this theorem below.

In the Golub–Kahan algorithm the reduction is achieved by applying a sequence of Householder reflections alternately from left and right. We set $A = A^{(1)}$ and in the first step compute

$$A^{(2)} = Q_1(AP_1) = \begin{pmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ \beta_2 & \tilde{a}_{22} & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ 0 & \tilde{a}_{32} & \tilde{a}_{33} & \cdots & \tilde{a}_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & \tilde{a}_{m2} & \tilde{a}_{m3} & \cdots & \tilde{a}_{mn} \end{pmatrix}.$$

First P_1 is chosen to zero the $n-1$ elements in the first column of A above the main diagonal. Next Q_1 is chosen to zero the last $m-2$ elements in the the first row of AP_1. This transformation does not affect the zeros introduced in the first row.

All later steps are similar and in the kth step, $k = 1 : \min(m, n)$, we compute

$$A^{(k+1)} = Q_k(A^{(k)}P_k),$$

where Q_k and P_k are Householder reflections. Here P_k is chosen to zero the last $n-k$ elements in the kth row of $A^{(k)}$. Then Q_k is chosen to zero the last $m-(k+1)$ elements in the kth column of $A^{(k)}P_k$.

When $m > n$ the process ends with the factorization

$$U^TAV = \begin{pmatrix} B \\ 0 \end{pmatrix}, \quad B = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_n \\ & & & \beta_{n+1} \end{pmatrix} \in \mathbf{R}^{(n+1)\times n}, \tag{8.4.30}$$

$$U = Q_1Q_2\cdots Q_n, \qquad V = P_1P_2\cdots P_{n-1}. \tag{8.4.31}$$

Note that since Q_k, only works on rows $k+1 : m$, and P_k, only works on columns $k : m$. It follows that

$$u_1 = e_1. \quad u_k = Ue_k = Q_1\cdots Q_ke_k, \quad k = 2 : n, \tag{8.4.32}$$

$$v_k = Ve_k = P_1\cdots P_ke_k, \quad k = 1 : n-1, \quad v_n = e_n. \tag{8.4.33}$$

If $m \le n$ then we obtain

$$U^TAV = (\, B \quad 0\,), \quad B = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \ddots & & \\ & & \ddots & \alpha_{m-1} & \\ & & & \beta_m & \alpha_m \end{pmatrix} \in \mathbf{R}^{m\times m}.$$

$$U = Q_1Q_2\cdots Q_{m-2}, \qquad V = P_1P_2\cdots P_{m-1}.$$

The above process can always be carried through although some elements in B may vanish. Note that the singular values of B equal those of A; in particular $\operatorname{rank}(A) = \operatorname{rank}(B)$. Using complex Householder transformations (see Sec. 9.6.2) a complex matrix A can be reduced to *real* bidiagonal form. by the algorithm above.

The reduction to bidiagonal form is backward stable in the following sense. The computed $\bar{B}$ can be shown to be the exact result of an orthogonal transformation from left and right of a matrix $A + E$, where

$$\|E\|_F \le cn^2u\|A\|_F, \tag{8.4.34}$$

and c is a constant of order unity. Moreover, if we use the information stored to generate the products $U = Q_1\cdots Q_n$ and $V = P_1\cdots P_{n-2}$ then the computed matrices are close to the exact matrices U and V which reduce $A + E$. This will guarantee that the singular values and transformed singular vectors of $\bar{B}$ are accurate approximations to those of a matrix close to A.

The bidiagonal reduction algorithm as described above requires approximately

$$4(mn^2 - n^3/3) \text{ flops}$$

when $m \geq n$, which is twice the work for a Householder QR factorization. The Householder vectors associated with U can be stored in the lower triangular part of A and those associated with V in the upper triangular part of A. Normally U and V are not explicitly required. They can be accumulated at a cost of $4(m^2n - mn^2 + \frac{1}{3}n^3)$ and $\frac{4}{3}n^3$ flops respectively.

When $m \gg n$ it is more efficient to use a two-step procedure as originally suggested by Lawson and Hanson [38] and later analyzed by T. Chan. In the first step the QR factorization of A is computed (possibly using column pivoting)

$$AP = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad R \in \mathbf{R}^{n \times n},$$

which requires $4mn^2 - \frac{2}{3}n^3$ flops. In the second step the upper triangular matrix R is transformed to bidiagonal form using the algorithm described above. Note that no advantage can be taken of the triangular structure of R in the Householder algorithm. Already the first postmultiplication of R with P_1 will cause the lower triangular part of R to fill in. Hence the Householder reduction of R to bidiagonal form will require $\frac{4}{3}n^3$ flops. The complete reduction to bidiagonal form then takes a total of

$$2(mn^2 + n^3) \text{ flops.}$$

This is less than the original Golub–Kahan algorithm when $m/n > 5/3$. Trefethen and Bau [62, pp. 237–238] have suggested a blend of the two above approaches that reduces the operation count slightly for $1 < m/n < 2$. They note that after k steps of the Golub–Kahan reduction the aspect ratio of the reduced matrix is $(m-k)/(n-k)$. and thus increases with k. To minimize the total operation count one should switch to the Chan algorithm when $(m-k)/(n-k) = 2$. This gives the operation count

$$4(mn^2 - n^3/3 - (m-n)^3/6) \text{ flops,}$$

a modest approval over the two other methods when $n > m > 2n$.

If Givens transformation are used to reduce R to upper bidiagonal form it is possible to take advantage of the triangular form, provided that the elements are annihilated in a suitable order. In the first major step we can zero the elements in the first row from right to left, i.e. in the order $r_{1n}, \ldots, r_{13}$. To zero r_{1j} the columns $(j-1, j)$ are rotated using a Givens rotation $G_{j-1,j}$ from the right. This introduces one new non-zero element $r_{j,j-1}$ in the *lower triangular part*, which can be annihilated by a rotation of the rows $(j-1, j)$, applying a Givens rotation $\tilde{G}_{j-1,j}$ from the left. This is illustrated below, where the element r_{14} is zeroed by first rotating columns 3,4 followed by a rotation of rows 3,4.

$$\begin{pmatrix} \times & \times & \times & \otimes & 0 & 0 \\ & \times & \times & \times & \times & \times \\ & & \times & \times & \times & \times \\ & & \otimes & \times & \times & \times \\ & & & & \times & \times \\ & & & & & \times \end{pmatrix}$$

After zeroing the last $n-2$ elements in the first row we continue the reduction on the triangular submatrix in rows and columns $2:n$ in the same fashion.

Since *two* Givens rotations are needed to zero each of the $(n-1)(n-2)/2$ elements, the operation count turns out to be about the same as for the Householder reduction if standard Givens rotations are used. If the transformations are to be accumulated the Givens reduction will require more work, unless fast Givens transformations are used.

When A is a banded matrix of bandwidth $w > 2$ then R will be an upper triangular banded matrix ($w = 2$ corresponds to a bidiagonal matrix). In this case the reduction of R to bidiagonal form can be accomplished by successively reducing the bandwidth by one. (This algorithm is similar to an algorithm by Schwarz [55] for reducing a symmetric banded matrix to tridiagonal form.) Each zero element introduced generates a new nonzero element that has to be chased across the border of the matrix. Because of this the reduction is expensive unless the bandwidth is small.

Example 8.4.4.

Let $w = 3$ and $n = 7$. The figure below illustrates the steps in zeroing out the element r_{13} using Givens rotations applied alternately from the right and left

$$\begin{pmatrix} \times & \times & \times & & & & \\ & \times & \times & \times & & & \\ & & \times & \times & \times & & \\ & & & \times & \times & \times & \\ & & & & \times & \times & \times \\ & & & & & \times & \times \\ & & & & & & \times \end{pmatrix} \Rightarrow \begin{pmatrix} \times & \times & \otimes & & & & \\ & \times & \times & \times & \oplus & & \\ & \oplus & \times & \times & \times & & \\ & & & \times & \times & \times & \oplus \\ & & & \oplus & \times & \times & \times \\ & & & & & \times & \times \\ & & & & & \oplus & \times \end{pmatrix}.$$

corresponding to the transformations

$$G_{67}((G_{45}((G_{23}(RG_{23}))G_{45}))G_{67}).$$

Then the elements $r_{13}, \ldots, r_{n-2,n}$ are eliminated in this order. Such "chasing" algorithms are also commonly used in eigenvalue algorithms. Reduction of an upper triangular matrix of bandwidth w to bidiagonal form requires $\approx 4n^2(w-2)$ multiplications.

We now derive an algorithm for solving the linear least squares problem $\min \|Ax - b\|_2$, where $A \in \mathbf{R}^{m\times n}$, $m \geq n$. Let Q_0 be a Householder reflection such that

$$Q_1 b = \beta_1 e_1. \tag{8.4.35}$$

Using the Golub–Kahan algorithm $Q_1 A$ to lower triangular form. we obtain

$$U^T \begin{pmatrix} b & AV \end{pmatrix} = \begin{pmatrix} \beta_1 e_1 & B_n \\ 0 & 0 \end{pmatrix}, \tag{8.4.36}$$

where e_1 is the first unit vector, and B_n is lower bidiagonal,

$$B_n = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_n & \alpha_n \\ & & & \beta_{n+1} \end{pmatrix} \in \mathbf{R}^{(n+1)\times n}, \tag{8.4.37}$$

and

$$U = Q_1 Q_2 \cdots Q_{n+1}. \qquad V = P_1 P_2 \cdots P_{n-1}. \tag{8.4.38}$$

(Note the minor difference in notation in that Q_{k+1} now zeros elements in the kth column of A.)

Setting $x = Vy$ and using the invariance of the l_2-norm it follows that

$$\begin{aligned} \|b - Ax\|_2 &= \left\| \begin{pmatrix} b & A \end{pmatrix} \begin{pmatrix} -1 \\ x \end{pmatrix} \right\|_2 = \left\| U^T \begin{pmatrix} b & AV \end{pmatrix} \begin{pmatrix} -1 \\ y \end{pmatrix} \right\|_2 \\ &= \|\beta_1 e_1 - B_n y\|_2. \end{aligned}$$

Hence if y solves the bidiagonal least squares problem

$$\min_y \|B_n y - \beta_1 e_1\|_2, \tag{8.4.39}$$

then $x = Vy$ minimizes $\|Ax - b\|_2$.

The least squares solution to (8.4.39) is obtained by a QR factorization of B_n, which takes the form

$$G_n(B_n \mid \beta_1 e_1) = \left(\begin{array}{c|c} R_n & \begin{matrix} f_k \\ \bar{\phi}_{n+1} \end{matrix} \end{array} \right) = \left(\begin{array}{ccccc|c} \rho_1 & \theta_2 & & & & \phi_1 \\ & \rho_2 & \theta_3 & & & \phi_2 \\ & & \rho_3 & \ddots & & \phi_3 \\ & & & \ddots & \theta_n & \vdots \\ & & & & \rho_n & \phi_n \\ \hline & & & & & \phi_{n+1} \end{array} \right) \tag{8.4.40}$$

where G_n is a product of n Givens rotations. The solution is obtained by back-substitution from $R_n y = d_n$. The norm of the corresponding residual vector equals $|\bar{\phi}_{n+1}|$. To zero out the element β_2 we premultiply rows (1,2) with a rotation G_{12}, giving

$$\begin{pmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{pmatrix} \left(\begin{array}{cc|c} \alpha_1 & 0 & \beta_1 \\ \beta_2 & \alpha_2 & 0 \end{array} \right) = \left(\begin{array}{cc|c} \rho_1 & \theta_2 & \phi_1 \\ 0 & \bar{\rho}_2 & \bar{\phi}_2 \end{array} \right).$$

(Here and in the following only elements affected by the rotation are shown.) Here the elements ρ_1, θ_2 and ϕ_1 in the first row are final, but $\bar{\rho}_2$ and $\bar{\phi}_2$ will be transformed into ρ_2 and ϕ_2 in the next step.

Continuing in this way in step j the rotation $G_{j,j+1}$ is used to zero the element β_{j+1}. In steps, $j = 2 : n-1$, the rows $(j, j+1)$ are transformed

$$\left(\begin{array}{cc} c_j & s_j \\ -s_j & c_j \end{array}\right)\left(\begin{array}{cc|c} \bar{\rho}_j & 0 & \bar{\phi}_j \\ \beta_{j+1} & \alpha_{j+1} & 0 \end{array}\right) = \left(\begin{array}{cc|c} \rho_j & \theta_{j+1} & \phi_j \\ 0 & \bar{\rho}_{j+1} & \bar{\phi}_{j+1} \end{array}\right).$$

where

$$\begin{aligned} &\phi_j = c_j\bar{\phi}_j, \qquad \bar{\phi}_{j+1} = -s_j\bar{\phi}_j, \qquad \rho_j = \sqrt{\bar{\rho}_j^2 + \beta_{j+1}^2}, \\ &\theta_{j+1} = s_j\alpha_{j+1}, \qquad \bar{\rho}_{n+1} = c_j\alpha_{j+1}. \end{aligned}$$

Note that by construction $|\bar{\phi}_{j+1}| \le \bar{\phi}_j$. Finally, in step n we obtain

$$\left(\begin{array}{cc} c_n & s_n \\ -s_n & c_n \end{array}\right)\left(\begin{array}{cc|c} \bar{\rho}_n & & \bar{\phi}_n \\ \beta_{n+1} & & 0 \end{array}\right) = \left(\begin{array}{c|c} \rho_n & \phi_n \\ 0 & \bar{\phi}_{n+1} \end{array}\right).$$

After n steps, we have obtained the factorization (8.4.40) with

$$G_n = G_{n,n+1}\cdots G_{23}G_{12}.$$

Now consider the result after $k < n$ steps of the above bidiagonalization process have been carried out. At this point we have computed $Q_1, Q_2, \ldots, Q_{k+1}$, $P_1, P_2, \ldots, P_k$ such that the first k columns of A are in lower bidiagonal form, i.e.

$$Q_{k+1}\cdots Q_2Q_1\, A\, P_1P_2\cdots P_k\begin{pmatrix} I_k \\ 0 \end{pmatrix} = \begin{pmatrix} B_k \\ 0 \end{pmatrix} = \begin{pmatrix} I_{k+1} \\ 0 \end{pmatrix} B_k,$$

where $B_k \in \mathbf{R}^{(k+1)\times k}$ is a leading submatrix of B_n. Multiplying both sides with $Q_1Q_2\cdots Q_{k+1}$ and using orthogonality we obtain the relation

$$AV_k = U_{k+1}B_k = \hat{B}_k + \beta_{k+1}v_{k+1}e_k^T, \quad k = 1 : n, \tag{8.4.41}$$

where

$$\begin{aligned} &P_1P_2\cdots P_k\begin{pmatrix} I_k \\ 0 \end{pmatrix} = V_k = (v_1, \ldots, v_k), \\ &Q_1Q_2\cdots Q_{k+1}\begin{pmatrix} I_{k+1} \\ 0 \end{pmatrix} = U_{k+1} = (u_1, \ldots, u_{k+1}). \end{aligned}$$

If we consider the intermediate result after applying also P_{k+1} the first $k+1$ rows have been transformed into bidiagonal form, i.e.

$$(\, I_{k+1} \quad 0\,)\, Q_{k+1}\cdots Q_2Q_1\, A\, P_1P_2\cdots P_{k+1} = (\, B_k \quad \alpha_{k+1}e_{k+1}\,)\,(\, I_{k+1} \quad 0\,).$$

Transposing this gives a second relation

$$U_{k+1}^TA = B_kV_k^T + \alpha_{k+1}e_{k+1}v_{k+1}^T, \tag{8.4.42}$$

We now show that the bidiagonalization can be stopped prematurely if a zero element occurs in B. Assume first that the first zero element to occur is $\alpha_{k+1} = 0$. Then we have obtained the decomposition

$$\tilde{U}_{k+1}^T A \tilde{V}_k = \begin{pmatrix} B_k & 0 \\ 0 & A_k \end{pmatrix},$$

where $A_k \in \mathbf{R}^{(m-k-1)\times(n-k)}$, and

$$\tilde{U}_{k+1} = Q_{k+1} \cdots Q_2 Q_1, \qquad \tilde{V}_k = P_1 P_2 \cdots P_k,$$

are square orthogonal matrices. Then, setting $x = \tilde{V}_k y$, the transformed least squares problem takes the form

$$\min_y \left\| \begin{pmatrix} B_k & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} \right\|_2, \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \tag{8.4.43}$$

$y_1 \in \mathbf{R}^k$, $y_2 \in \mathbf{R}^{n-k}$. This problem is separable and decomposes into two independent subproblems

$$\min_{y_1} \|B_k y_1 - \beta_1 e_1\|_2, \qquad \min_{y_2} \|A_2 y_2\|_2. \tag{8.4.44}$$

By construction B_k has nonzero elements in its two diagonals. Thus it has full column rank and the solution y_1 to the first subproblem is unique. Further, the minimum norm solution of the initial problem is obtained simply by taking $y_2 = 0$. We call the first subproblem (8.4.44) a **core subproblem**. It can be solved by QR factorization exactly as outlined for the full system when $k = n$.

When $\beta_{k+1} = 0$ is the first zero element to occur then the reduced problem has a similar separable form similar to (8.4.44). The core subproblem is now

$$\hat{B}_k y_1 = \beta_1 e_1, \quad \hat{B}_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{pmatrix} \in \mathbf{R}^{k\times k}. \tag{8.4.45}$$

Here $\hat{B}_k$ is square and lower triangular, and the solution y_1 is obtained by forward substitution. Taking $y_2 = 0$ the corresponding residual $b - AVy$ is zero and hence the original system $Ax = b$ is consistent.

We give two simple examples of when premature termination occurs. First assume that $b \perp \mathcal{R}(A)$. Then the reduction will terminate with $\alpha_1 = 0$. The core system is empty and $x = Vy_2 = 0$ is the minimal norm least squares solution.

If the bidiagonalization instead terminates with $\beta_2 = 0$, then the system $Ax = b$ is consistent and the minimum norm solution equals

$$x = (\beta_1/\alpha_1)v_1, \quad v_1 = V_1 e_1 = P_1 e_1 4.$$

Paige and Strakoš [46] have shown the following important properties of the core subproblem obtained by the bidiagonalization algorithm:

Theorem 8.4.5.

Assume that the bidiagonalization of $(\,b \quad A\,)$ *terminates prematurely with* $\alpha_k = 0$ *or* $\beta_{k+1} = 0$. *Then the core corresponding subproblem* (8.4.44) *or* (8.4.45) *is minimally dimensioned. Further, the singular values of the core matrix* B_k *or* $\hat{B}_k$, *are simple and the right hand side* βe_1 *has nonzero components in along each left singular vector.*

Proof. Sketch: The minimal dimension is a consequence of the uniqueness of the decomposition (8.4.36), as long as no zero element in B appears. That the matrix $\hat{B}_k$ has simple singular values follows from the fact that all subdiagonal elements are nonzero. The same is true for the square bidiagonal matrix $(B_k\ 0)$ and therefore also for B_k. Finally, if βe_1 did not have nonzero components along a left singular vector, then the reduction must have terminated earlier. For a complete proof we refer to [46].) □

In many applications the numerical rank of the matrix A is much smaller than $\min\{m, n\}$. For example, in multiple linear regression often some columns are nearly linearly dependent. Then one wants to express the solution by restricting it to lie in a lower dimensional subspace. This can be achieved by neglecting small singular values of A and using a truncated SVD solution; see Sec. 8.4.1. In **partial least squares** (PLS) method this is achieved by a partial bidiagonalization of the matrix $(\,b \quad A\,)$. It is known that PLS often gives a faster reduction of the residual than TSVD.

We remark that the solution steps can be interleaved with the reduction to bidiagonal form. This makes it possible to compute a sequence of *approximate solutions* $x_k = P_1 P_2 \cdots P_k y_k$, where $y_k \in \mathbf{R}^k$ solves

$$\min_y \|\beta_1 e_1 - B_k y\|_2, \quad k = 1, 2, 3, \ldots. \tag{8.4.46}$$

After each (double) step in the bidiagonalization we advance the QR decomposition of B_k. The norm of the least squares residual corresponding to x_k is then given by

$$\|b - Ax_k\|_2 = |\bar{\phi}_{k+1}|.$$

The sequence of residual norms is nonincreasing. We stop and accept $x = V_k y_k$ as an approximate solution of the original least squares problem. if this residual is sufficiently small. This method is called the **Partial Least Squares** (PLS) method in statistics.

The sequential method outlined here is mathematically equivalent to a method called LSQR, which is a method of choice for solving *sparse* linear least squares. LSQR uses a Lanczos-type process for the bidiagonal reduction, which works only with the original sparse matrix. A number of important properties of the successive approximations x_k in PLS are best discussed in connection with LSQR; see Sec. 10.6.4.

Review Questions

1. When and why should column pivoting be used in computing the QR factorization of a matrix? What inequalities will be satisfied by the elements of R if the standard column pivoting strategy is used?
2. Show that the singular values and condition number of R equal those of A. Give a simple lower bound for the condition number of A in terms of its diagonal elements. Is it advisable to use this bound when no column pivoting has been performed?
3. Give a simple lower bound for the condition number of A in terms of the diagonal elements of R. Is it advisable to use this bound when no column pivoting has been performed?
4. What is meant by a Rank-revealing QR factorization? Does such a factorization always exist?
5. How is the *numerical* rank of a matrix A defined? Give an example where the numerical rank is not well determined.

Problems

1. (a) Describe how the QR factorizations of a matrix of the form

$$\begin{pmatrix} A \\ \mu D \end{pmatrix}, \quad A \in \mathbf{R}^{m\times n},$$

where $D \in \mathbf{R}^{n\times n}$ is diagonal, can be computed using Householder transformations in mn^2 flops.

(b) Estimate the number of flops that are needed for the reduction using Householder transformations in the special case that $A = R$ is upper triangular? Devise a method using Givens rotations for this special case!

Hint: In the Givens method zero one diagonal at a time in R working from the main diagonal inwards.

2. Let the vector v, $\|v\|_2 = 1$, satisfy $\|Av\|_2 = \epsilon$, and let Π be a permutation such that

$$|w_n| = \|w\|_\infty, \qquad \Pi^T v = w.$$

(a) Show that if R is the R factor of $A\Pi$, then $|r_{nn}| \le n^{1/2}\epsilon$.

Hint: Show that $\epsilon = \|Rw\|_2 \ge |r_{nn}w_n|$ and then use the inequality $|w_n| = \|w\|_\infty \ge n^{-1/2}\|w\|_2$.

(b) Show using (a) that if $v = v_n$, the right singular vector corresponding to the smallest singular value $\sigma_n(A)$, then

$$\sigma_n(A) \ge n^{-1/2}|r_{nn}|.$$

4. Consider a nonsingular 2×2 upper triangular matrix and its inverse

$$R = \begin{pmatrix} a & b \\ 0 & c \end{pmatrix}, \qquad R^{-1} = \begin{pmatrix} a^{-1} & a^{-1}bc^{-1} \\ 0 & c^{-1} \end{pmatrix}.$$

(a) Suppose we want to choose Π to *maximize* the $(1,1)$ element in the QR factorization of $R\Pi$. Show that this is achieved by taking $\Pi = I$ if $|a| \geq \sqrt{b^2 + c^2}$, else $\Pi = \Pi_{12}$, where Π_{12} interchanges columns 1 and 2.

(b) Unless $b = 0$ the permutation chosen in (a) may not *minimize* the (2,2) element in the QR factorization of $R\Pi$. Show that this is achieved by taking $\Pi = I$ if $|c^{-1}| \geq \sqrt{a^{-2} + b^2(ac)^{-2}}$ else $\Pi = \Pi_{12}$. Hence, the test compares *row* norms in R^{-1} instead of *column* norms in R.

6. To minimize $\|x\|_2$ is not always a good way to resolve rank deficiency, and therefore the following generalization of problem (8.4.13) is often useful: For a given matrix $B \in \mathbf{R}^{p\times n}$ consider the problem

$$\min_{x\in S} \|Bx\|_2, \qquad S = \{x \in \mathbf{R}^n \mid \|Ax - b\|_2 = \min\}.$$

(a) Show that this problem is equivalent to

$$\min_{x_2} \|(BC)x_2 - (Bd)\|_2,$$

where C and d are defined by (8.4.12).

(b) Often one wants to choose B so that $\|Bx\|_2$ is a measure of the smoothness of the solution x. For example one can take B to be a discrete approximation to the second derivative operator,

$$B = \begin{pmatrix} 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 1 \end{pmatrix} \in \mathbf{R}^{(n-2)\times n}.$$

Show that provided that $\mathcal{N}(A) \cap \mathcal{N}(B) = \emptyset$ this problem has a unique solution, and give a basis for $\mathcal{N}(B)$.

5. Let $A \in \mathbf{R}^{m\times n}$ with $\operatorname{rank}(A) = r$. A rank revealing LU factorizations of the form

$$\Pi_1 A \Pi_2 = \begin{pmatrix} L_{11} \\ L_{21} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \end{pmatrix},$$

where Π_1 and Π_2 are permutation matrices and $L_{11}, U_{11} \in \mathbf{R}^{r\times r}$ are triangular and nonsingular can also be used to compute pseudo-inverse solutions $x = A^{\dagger}b$. Show, using Theorem 8.1.7 that

$$A^{\dagger} = \Pi_2 \begin{pmatrix} I_r & S \end{pmatrix}^{\dagger} U_{11}^{-1} L_{11}^{-1} \begin{pmatrix} I_r \\ T \end{pmatrix}^{\dagger} \Pi_1,$$

where $T = L_{21}L_{11}^{-1}$, $S = U_{11}^{-1}U_{12}$. (Note that S is empty if $r = n$, and T empty if $r = m$.)

6. Consider the block upper-bidiagonal matrix

$$A = \begin{pmatrix} B_1 & C_1 & \\ & B_2 & C_2 \\ & & B_3 \end{pmatrix}$$

Outline an algorithm for computing the QR factorization of A, which treats one block row at a time. (It can be assumed that A has full column rank.) Generalize the algorithm to an arbitrary number of block rows!

7. (a) Suppose that we have computed the pivoted QR factorization of A,

$$Q^T A\Pi = \begin{pmatrix} R \\ 0 \end{pmatrix} \in \mathbf{R}^{m\times n},$$

of a matrix $A \in \mathbf{R}^{m\times n}$. Show that by postmultiplying the *upper* triangular matrix R by a sequence of Householder transformations we can transform R into a *lower* triangular matrix $L = RP \in \mathbf{R}^{n\times n}$ and that by combining these two factorizations we obtain

$$Q^T A\Pi P = \begin{pmatrix} L \\ 0 \end{pmatrix}. \tag{8.4.47}$$

This factorization, introduced by G. W. Stewart, who calls it the **QLP decomposition** of A.

(b) Show that the total cost for computing the QLP decomposition is roughly $2mn^2 + 2n^3/3$ flops. How does that compare with the cost for computing the bidiagonal decomposition of A?

(c) Show that the two factorizations can be interleaved. What is the cost for performing the first k steps?

8. Work out the details of an algorithm for transforming a matrix $A \in \mathbf{R}^{m\times n}$ to *lower* bidiagonal form. Consider both cases when $m > n$ and $m \le n$.

Hint: It can be derived by applying the algorithm for transformation to upper bidiagonal form to A^T.

8.5 Some Structured Least Squares Problems

8.5.1 Banded Least Squares Problems

We now consider orthogonalization methods for the special case when A is a banded matrix of row bandwidth w, see Definition 8.2.3. From Theorem 8.2.4 we know that the matrix A^TA will also be a banded matrix with only the first $r = w - 1$ superdiagonals nonzero. Since the factor R in the QR factorization equals the unique Cholesky factor of A^TA it will have only w nonzeros in each row.

Even though the *final* factor R is independent of the row ordering in A, the intermediate fill-in will vary. For banded rectangular matrices the QR factorization can be obtained efficiently by sorting the rows of A and suitably subdividing the Householder transformations. The rows of A should be sorted by leading entry

order (i.e., increasing minimum column subscript order) That is, if $f_i, i = 1, 2, \ldots, m$ denotes the column indices of the first nonzero element in row i we should have,

$$i \le k \Rightarrow f_i \le f_k.$$

Such a band matrix can then be written as

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_q \end{pmatrix}, \quad q \le n,$$

is said to be in **standard form**. where in block A_i the first nonzero element of each row is in column i. The Householder QR process is then applied to the matrix in q major steps. In the first step a QR decomposition of the first block A_1 is computed, yielding R_1. Next at step k, $k = 2 : q$, R_{k-1} will be merged with A_k yielding

$$Q_k^T \begin{pmatrix} R_{k-1} \\ A_k \end{pmatrix} = R_k.$$

Since the rows of block A_k has their first nonzero elements in column k, the first $k-1$ rows of R_{k-1} will not be affected. The matrix Q can be implicitly represented in terms of the Householder vectors of the factorization of the subblocks. This sequential Householder algorithm, which is also described in [38, Ch. 27], requires $(m + 3n/2)w(w + 1)$ multiplications or about twice the work of the less stable Cholesky approach. For a detailed description of this algorithm, see Lawson and Hanson [38, Ch. 11].

In Sec. 4.6.4 we considered the interpolation of a function f where with a linear combination of $m + k$ B-splines of degree k, see (4.6.18), on $\Delta = \{x_0 < x_1 < \cdots < x_m\}$. Assume that we are given function values $f_j = f(\tau_j)$, where $\tau_1 < \tau_2 < \ldots < \tau_n$ are distinct points and $n \ge m + k$. Then we consider the least squares approximation problem

$$\min \sum_{j=1}^{n} e_j^2, \qquad e_j = w_i \Big(f_j - \sum_{i=-k}^{m-1} c_i B_{i,k+1}(\tau_j) \Big). \tag{8.5.1}$$

where w_j are positive weights. This is an overdetermined linear system for c_i, $i = -k, \ldots, m-1$. The elements of its coefficient matrix $B_{i,k+1}(\tau_j)$ can be evaluated by the recurrence (4.6.19). The coefficient matrix has a band structure since in the jth row the ith element will be zero if $\tau_j \notin [x_i, x_{i+k+1}]$. It can be shown, see de Boor [1978, p. 200], that the coefficient matrix will have full rank equal to $m + k$ if and only if there is a subset of points τ_j satisfying

$$x_{j-k-1} < \tau_j < x_j, \qquad \forall j = 1, 2, \ldots, m + k. \tag{8.5.2}$$

Example 8.5.1.

The least squares approximation of a discrete set of data by a linear combination of cubic B-splines gives rise to a banded linear least squares problem. Let

$$s(t) = \sum_{j=1}^{n} x_j B_j(t),$$

where $B_j(t)$, $j = 1 : n$ are the normalized cubic B-splines, and let (y_i, t_i), $i = 1 : m$ be given data points. If we determine x to minimize

$$\sum_{i=1}^{m} (s(t_i) - y_i)^2 = \|Ax - y\|_2^2,$$

then A will be a banded matrix with $w = 4$. In particular if $m = 13$, $n = 8$ the matrix may have the form shown in Fig. 8.4.2. Here A consists of blocks A_k^T, $k = 1 : 7$. In the Fig. 8.4.2 we also show the matrix after the first three blocks have been reduced by Householder transformations $H_1, \ldots, H_9$. Elements which have been zeroed by H_j are denoted by j and fill-in elements by $+$. In step $k = 4$ only the indicated part of the matrix is involved.

$$\begin{pmatrix}
\times & \times & \times & \times & & & & \\
1 & \times & \times & \times & + & & & \\
1 & 2 & \times & \times & + & + & & \\
 & 3 & 4 & \times & \times & + & & \\
 & 3 & 4 & 5 & \times & + & & \\
 & & 6 & 7 & 8 & \times & & \\
 & & 6 & 7 & 8 & 9 & & \\
 & & 6 & 7 & 8 & 9 & & \\
 & & & \times & \times & \times & \times & \\
 & & & \times & \times & \times & \times & \\
 & & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times & \times
\end{pmatrix}$$

Figure 8.5.1. *A banded rectangular matrix A after $k = 3$ steps in the QR reduction.*

In the algorithm the Householder transformations can also be applied to one or several right hand sides b to produce

$$c = Q^T b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \quad c_1 \in \mathbf{R}^n.$$

The least squares solution is then obtained from $Rx = c_1$ by back-substitution. The vector c_2 is not stored but used to accumulate the residual sum of squares $\|r\|_2^2 = \|c_2\|_2^2$.

It is also possible to perform the QR factorization by treating one row at a time using Givens' rotations. Each step then is equivalent to updating a full triangular matrix formed by columns $f_i(A)$ to $l_i(A)$. Further, if the matrix A is in standard form the first $f_i(A)$ rows of R are already finished at this stage. The reader is encouraged to work through Example 8.5.1 below in order to understand how the algorithm proceeds!

8.5.2 Two-Block Least Squares Problems

In many least squares problem $\min_x \|Ax - b\|_2^2$, $A \in \mathbf{R}^{m\times n}$ the unknowns can be naturally partitioned into two groups,

$$\min_{x_1,x_2} \left\| \begin{pmatrix} A_1 & A_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - b \right\|_2, \tag{8.5.3}$$

with n_1 and n_2 components, respectively, $n = n_1 + n_2$. Assume that the matrix $A = \begin{pmatrix} A_1 & A_2 \end{pmatrix}$ has full column rank.

Let $P_{\mathcal{R}(A_1)}$ be the orthogonal projection onto $\mathcal{R}(A_1)$. For any x_2 we can split the vector $b - A_2x_2 = r_1 + r_2$ into two orthogonal components

$$r_1 = P_{\mathcal{R}(A_1)}(b - A_2x_2), \quad r_2 = (I - P_{\mathcal{R}(A_1)})(b - A_2x_2).$$

Then the problem (8.5.3) takes the form

$$\min_{x_1,x_2} \left\| (A_1x_1 - r_1) - P_{\mathcal{N}(A_1^T)}(b - A_2x_2) \right\|_2. \tag{8.5.4}$$

Here, since $r_1 \in \mathcal{R}(A_1)$ the variables x_1 can always be chosen so that $A_1x_1 - r_1$. It follows that x_2 is the solution to the **reduced least squares problem**

$$\min_{x_2} \|P_{\mathcal{N}(A_1^T)}(A_2x_2 - b)\|_2. \tag{8.5.5}$$

When this reduced problem has been solved for x_2 the unknowns x_1 can be computed from the least squares problem

$$\min_{x_1} \|A_1x_1 - (b - A_2x_2)\|_2. \tag{8.5.6}$$

Sometimes it may be advantageous to carry out a **partial** QR factorization, where only the first $k < n$ columns are orthogonalized. Suppose that after k steps of MGS, we have computed the partial factorization

$$(A, b) = (Q_k, A^{(k+1)}, b^{(k+1)}) \begin{pmatrix} R_{11} & R_{12} & z_k \\ 0 & I & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

where R_{11} is nonsingular. Then we can decompose the residual as $r = b - Ax = r_1 + r_2$, $r_1 \perp r_2$, where

$$r_1 = Q_k(z_k - R_{12}x_2 - R_{11}x_1), \qquad r_2 = b^{(k+1)} - A^{(k+1)}x_2.$$

Then x_2 is the solution to the reduced least squares problem

$$\min_{x_2} \|b^{(k+1)} - A^{(k+1)}x_2\|_2.$$

With x_2 known x_1 can been computed by back-substitution from

$$R_{11}x_1 = z_k - R_{12}x_2.$$

In some applications, e.g., when A has block angular structure (see Sec. 8.5.4), it may be preferable instead not to save R_{11} and R_{12} and instead to refactorize A_1 and solve (8.5.6) for x_1.

8.5.3 Block Triangular Form of a Rectangular Matrix

An arbitrary rectangular matrix $A \in \mathbf{R}^{m\times n}$, $m \geq n$, can by row and column permutations be brought into the block triangular form

$$PAQ = \begin{pmatrix} A_h & U_{hs} & U_{hv} \\ & A_s & U_{sv} \\ & & A_v \end{pmatrix}, \tag{8.5.7}$$

where the diagonal block A_h is underdetermined (i.e., has more columns than rows), A_s is square and A_v is overdetermined (has more rows than columns), and all three blocks have a nonzero diagonal; see the example in Figure 7.6.4. The submatrices A_v and A_h^T both have the strong Hall property. The off-diagonal blocks denoted by U are possibly nonzero matrices of appropriate dimensions. This block triangular form (8.5.7) of a sparse matrix is based on a canonical decomposition of bipartite graphs.

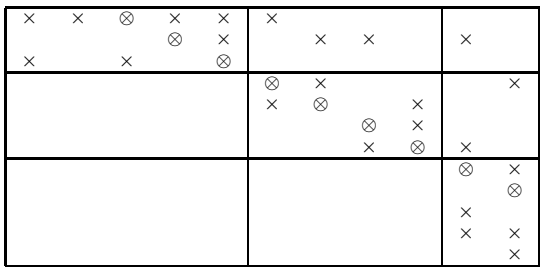

Figure 8.5.2. *The coarse block triangular decomposition of A.*

We call the decomposition of A into the submatrices A_h, A_s, and A_v the **coarse decomposition**. One or two of the diagonal blocks may be absent in the coarse decomposition. It may be possible to further decompose the diagonal blocks in (8.5.7) to obtain the **fine decompositions** of these submatrices. Each of the blocks A_h and A_v may be further decomposable into block diagonal form,

$$A_h = \begin{pmatrix} A_{h1} & & \\ & \ddots & \\ & & A_{hp} \end{pmatrix}, \quad A_v = \begin{pmatrix} A_{v1} & & \\ & \ddots & \\ & & A_{vq} \end{pmatrix},$$

where each $A_{h1}, \ldots, A_{hp}$ is underdetermined and each $A_{v1}, \ldots, A_{vq}$ is overdetermined. The submatrix A_s may be decomposable in block upper triangular form

$$A_s = \begin{pmatrix} A_{s1} & U_{12} & \ldots & U_{1,t} \\ & A_{s2} & \ldots & U_{2,t} \\ & & \ddots & \vdots \\ & & & A_{st} \end{pmatrix} \tag{8.5.8}$$

with square diagonal blocks $A_{s1}, \ldots, A_{st}$ which have nonzero diagonal elements. The resulting decomposition can be shown to be essentially unique. Any one block triangular form can be obtained from any other by applying row permutations that involve the rows of a single block row, column permutations that involve the columns of a single block column, and symmetric permutations that reorder the blocks.

An algorithm for the more general block triangular form described above due to Pothen and Fan depends on the concept of matchings in bipartite graphs. The algorithm consists of the following steps:

1. Find a maximum matching in the bipartite graph $G(A)$ with row set R and column set C.
2. According to the matching, partition R into the sets VR, SR, HR and C into the sets VC, SC, HC corresponding to the horizontal, square, and vertical blocks.
3. Find the diagonal blocks of the submatrix A_v and A_h from the connected components in the subgraphs $G(A_v)$ and $G(A_h)$. Find the block upper triangular form of the submatrix A_s from the strongly connected components in the associated directed subgraph $G(A_s)$, with edges directed from columns to rows.

The reordering to block triangular form in a preprocessing phase can save work and intermediate storage in solving least squares problems. If A has structural rank equal to n, then the first block row in (8.5.7) must be empty, and the original least squares problem can after reordering be solved by a form of block back-substitution. First compute the solution of

$$\min_{\tilde{x}_v} \|A_v \tilde{x}_v - \tilde{b}_v\|_2, \tag{8.5.9}$$

where $\tilde{x} = Q^T x$ and $\tilde{b} = Pb$ have been partitioned conformally with PAQ in (8.5.7). The remaining part of the solution $\tilde{x}_k, \ldots, \tilde{x}_1$ is then determined by

$$A_{si}\tilde{x}_i = \tilde{b}_i - \sum_{j=i+1}^{k} U_{ij}\tilde{x}_j, \quad i = k, \ldots, 2, 1. \tag{8.5.10}$$

Finally, we have $x = Q\tilde{x}$. We can solve the subproblems in (8.5.9) and (8.5.10) by computing the QR decompositions of A_v and $A_{s,i}$, $i = 1, \ldots, k$. Since $A_{s1}, \ldots, A_{sk}$ and A_v have the strong Hall property the structures of the matrices R_i are correctly predicted by the structures of the corresponding normal matrices.

If the matrix A has structural rank less than n, then we have an underdetermined block A_h. In this case we can still obtain the form (8.5.8) with a square block A_{11} by permuting the extra columns in the first block to the end. The least squares solution is then not unique, but a unique solution of minimum length can be found as outlined in Section 2.7.

8.5.4 Block Angular Least Squares Problems

There is often a substantial similarity in the structure of many large scale sparse least squares problems. In particular, the problem can often be put in the following bordered block diagonal or **block angular form**:

$$A = \left(\begin{array}{cccc|c} A_1 & & & & B_1 \\ & A_2 & & & B_2 \\ & & \ddots & & \vdots \\ & & & A_M & B_M \end{array}\right), \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \\ x_{M+1} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{pmatrix}, \tag{8.5.11}$$

where

$$A_i \in \mathbf{R}^{m_i \times n_i}, \quad B_i \in \mathbf{R}^{m_i \times n_{M+1}}, \quad i = 1, 2, \ldots, M,$$

and

$$m = m_1 + m_2 + \cdots + m_M, \qquad n = n_1 + n_2 + \cdots + n_{M+1}.$$

Note that the variables $x_1, \ldots, x_M$ are coupled only to the variables x_{M+1}, which reflects a "local connection" structure in the underlying physical problem. Applications where the form (8.5.11) arises naturally include photogrammetry, Doppler radar and GPS positioning, and geodetic survey problems.

The normal matrix of A in (8.5.11) is of doubly bordered block diagonal form,

$$A^T A = \left(\begin{array}{cccc|c} A_1^T A_1 & & & & A_1^T B_1 \\ & A_2^T A_2 & & & A_2^T B_2 \\ & & \ddots & & \vdots \\ & & & A_M^T A_M & A_M^T B_M \\ \hline B_1^T A_1 & B_2^T A_2 & \cdots & B_M^T A_M & C \end{array}\right),$$

where

$$C = \sum_{k=1}^{M} B_k^T B_k = R_{M+1}^T R_{M+1},$$

and R_{M+1} is the Cholesky factor of C. We assume in the following that $\operatorname{rank}(A) = n$, which implies that the matrices $A_i^T A_i$, $i = 1, 2, \ldots, M$, and C are positive definite. It is easily seen that then the Cholesky factor R of $A^T A$ will have a block

structure similar to that of A,

$$R = \left(\begin{array}{cccc|c} R_1 & & & & S_1 \\ & R_2 & & & S_2 \\ & & \ddots & & \vdots \\ & & & R_M & S_M \\ \hline & & & & R_{M+1} \end{array}\right) \tag{8.5.12}$$

where $R_i \in \mathbf{R}^{n_i \times n_i}$, the Cholesky factor of $A_i^T A_i$, is nonsingular and

$$S_i = (A_i R_i^{-1})^T B_i, \quad i = 1, \ldots, M+1.$$

An algorithm for least squares problems of block angular form based on QR factorization of A proceeds in the following three steps:

1. For $i = 1, 2, \ldots, M$ reduce the diagonal block A_i to upper triangular form by a sequence of orthogonal transformations applied to (A_i, B_i) and the right-hand side b_i, yielding

$$Q_i^T(A_i, B_i) = \begin{pmatrix} R_i & S_i \\ 0 & T_i \end{pmatrix}, \qquad Q_i^T b_i = \begin{pmatrix} c_i \\ d_i \end{pmatrix}.$$

   It is usually advantageous to continue the reduction in step 1 so that the matrices T_i, $i = 1, \ldots, M$, are brought into upper trapezoidal form.

2. Set

$$T = \begin{pmatrix} T_1 \\ \vdots \\ T_M \end{pmatrix}, \qquad d = \begin{pmatrix} d_1 \\ \vdots \\ d_M \end{pmatrix}$$

   and compute the QR decomposition

$$\tilde{Q}_{M+1}^T \begin{pmatrix} T & d \end{pmatrix} = \begin{pmatrix} R_{M+1} & c_{M+1} \\ 0 & d_{M+1} \end{pmatrix}.$$

   The solution to $\min_{x_{M+1}} \|Tx_{M+1} - d\|_2$ is then obtained from the triangular system

$$R_{M+1} x_{M+1} = c_{M+1},$$

   and the residual norm is given by $\rho = \|d_{M+1}\|_2$.

3. For $i = M, \ldots, 1$ compute $x_M, \ldots, x_1$ by back-substitution in the triangular systems

$$R_i x_i = c_i - S_i x_{M+1}.$$

In steps 1 and 3 the computations can be performed in parallel on the M subsystems. There are alternative ways to organize this algorithm. Note that when

x_{M+1} has been computed in step 2, then the vectors x_i, $i = 1, \ldots, M$, solves the least squares problem

$$\min_{x_i} \|A_i x_i - g_i\|_2, \quad g_i = b_i - B_i x_{M+1}.$$

Hence it is possible to discard the R_i, S_i and c_i in step 1 if the QR factorizations of A_i are recomputed in step 3. In some practical problems this modification can reduce the storage requirement by an order of magnitude, while the recomputation of R_i may only increase the operation count by a few percent.

Using the structure of the R-factor in (8.5.12), the diagonal blocks of the variance-covariance matrix $C = (R^T R)^{-1} = R^{-1} R^{-T}$ can be written

$$\begin{aligned} &C_{M+1,M+1} = R_{M+1}^{-1} R_{M+1}^{-T}, \\ &C_{i,i} = R_i^{-1}(I + W_i^T W_i) R_i^{-T}, \quad W_i^T = S_i R_{M+1}^{-1}, \quad i = 1, \ldots, M. \end{aligned} \tag{8.5.13}$$

If we compute the QR decompositions

$$Q_i \begin{pmatrix} W_i \\ I \end{pmatrix} = \begin{pmatrix} U_i \\ 0 \end{pmatrix}, \quad i = 1, \ldots, M,$$

we have $I + W_i^T W_i = U_i^T U_i$ and then

$$C_{i,i} = (U_i R_i^{-T})^T (U_i R_i^{-T}), \quad i = 1, \ldots, M.$$

This assumes that all the matrices R_i and S_i have been retained.

8.5.5 Kronecker Product Problems

Sometimes least squares problems occur which have a highly regular block structure. Here we consider least squares problems of the form

$$\min_x \|(A \otimes B)x - d\|_2, \tag{8.5.14}$$

where the $A \otimes B$ is the **Kronecker product** of $A \in \mathbf{R}^{m \times n}$ and $B \in \mathbf{R}^{p \times q}$. This product is the $mp \times nq$ block matrix,

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}.$$

Problems of Kronecker structure arise in several application areas including signal and image processing, photogrammetry, and multidimensional approximation. It applies to least squares fitting of multivariate data on a rectangular grid. Such problems can be solved with great savings in storage and operations. Since often the size of the matrices A and B is large, resulting in models involving several hundred thousand equations and unknowns, such savings may be essential.

We recall from Sec. 7.7.3 the important rule (7.7.14) for the inverse of a Kronecker product

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}.$$

It follows that if P and Q are orthogonal $n \times n$ matrices then $P \otimes Q$ is an orthogonal $n^2 \times n^2$ matrix. This rule for the inverse holds also for pseudo-inverses.

Lemma 8.5.1.
Let $A^\dagger$ and $B^\dagger$ be the pseudo-inverses of A and B. Then

$$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger.$$

Proof. The theorem follows by verifying that $X = A^\dagger \otimes B^\dagger$ satisfies the four Penrose conditions in (8.1.11)–(8.1.12). □

Using Lemmas 7.7.6 and 8.5.1 the solution to the Kronecker least squares problem (8.5.14) can be written

$$x = (A \otimes B)^\dagger \mathrm{vec}\, C = (A^\dagger \otimes B^\dagger)\mathrm{vec}\, C = \mathrm{vec}\,(B^\dagger C (A^\dagger)^T). \tag{8.5.15}$$

This formula leads to a great reduction in the cost of solving Kronecker least squares problems. For example, if A and B are both $m \times n$ matrices, the cost of computing is reduced from $O(m^2 n^4)$ to $O(mn^2)$.

In some areas the most common approach to computing the least squares solution to (8.5.14) is to use the normal equations. If we assume that both A and B have full column rank, then we can use the expressions

$$A^\dagger = (A^T A)^{-1} A^T, \qquad B^\dagger = (B^T B)^{-1} B^T.$$

However, because of the instability associated with the explicit formation of $A^T A$ and $B^T B$, an approach based on orthogonal decompositions should generally be preferred. If we have computed the complete QR decompositions of A and B,

$$A\Pi_1 = Q_1 \begin{pmatrix} R_1 & 0 \\ 0 & 0 \end{pmatrix} V_1^T, \qquad B\Pi_2 = Q_2 \begin{pmatrix} R_2 & 0 \\ 0 & 0 \end{pmatrix} V_2^T,$$

with R_1, R_2 upper triangular and nonsingular, then from Section 2.7.3 we have

$$A^\dagger = \Pi_1 V_1 \begin{pmatrix} R_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} Q_1^T, \qquad B^\dagger = \Pi_2 V_2 \begin{pmatrix} R_2^{-1} & 0 \\ 0 & 0 \end{pmatrix} Q_2^T.$$

These expressions can be used in (8.5.15) to compute the pseudo-inverse solution of problem (8.5.14) even in the rank deficient case.

We finally note that the singular values and singular vectors of the Kronecker product $A \otimes B$ can be simply expressed in terms of the singular values and singular vectors of A and B.

Lemma 8.5.2. *Let A and B have the singular value decompositions*

$$A = U_1\Sigma_1 V_1^T, \qquad B = U_2\Sigma_2 V_2^T.$$

Then we have

$$A \otimes B = (U_1 \otimes U_2)(\Sigma_1 \otimes \Sigma_2)(V_1 \otimes V_2)^T.$$

Proof. The proof follows from Lemma 8.5.1. □

Review Questions

1. What is meant by the standard form of a banded rectangular matrix A? Why is it important that a banded matrix is permuted into standard form before its orthogonal factorization is computed?
2. In least squares linear regression the first column of A often equals the vector $a_1 = e = (1, 1, \ldots, 1)^T$ (cf. Example 8.2.1). Setting $A = (\, e \quad A_2\,)$, show that performing one step in MGS is equivalent to "subtracting out the means".

Problems

1. Consider the two-block least squares problem (8.5.3). Work out an algorithm to solve the reduced least squares problem $\min_{x_2} \|P_{\mathcal{N}(A_1^T)}(A_2x_2 - b)\|_2$ using the method of normal equations.

   Hint: First show that $P_{\mathcal{N}(A_1^T)}(A) = I - A_1(R_1^TR_1)^{-1}A_1^T$, where R_1 is the Cholesky factor of $A_1^TA_1$.
2. (a) Suppose we want to fit two set of points $(x_i, y_i) \in \mathbf{R}^2$, $i = 1, \ldots, p$, and $i = p+1, \ldots, m$, to two *parallel* lines

   $$cx + sy = h_1, \qquad cx + sy = h_2, \quad c^2 + s^2 = 1,$$

   so that the sum of orthogonal distances are minimized. Generalize the approach of Example 8.6.3 to sketch an algorithm for solving this problem.

   (b) Modify the algorithm in (a) to fit two *orthogonal* lines.
3. Use the Penrose conditions to prove the formula

   $$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger,$$

   where $\otimes$ denotes the Kronecker product

8.6 Generalized Least Squares

8.6.1 Generalized Least Squares

Let $A \in \mathbf{R}^{m\times n}$, $m \geq n$, and let $B \in \mathbf{R}^{m\times m}$ be symmetric positive semidefinite. Augmented linear systems of the form

$$\begin{pmatrix} B & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} s \\ x \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix} \tag{8.6.1}$$

in (8.6.1) occur in many application areas since they represent the condition for equilibrium of a physical system. The system (8.6.1) is often called a **saddle-point system** or, in optimization, a KKT (Karush–Kuhn–Tucker) system. The system matrix in (8.6.1) is symmetric but in general indefinite; it is nonsingular if and only if

1. A has full column rank;
2. the matrix $(B\ A)$ has full row rank.

A unified formulation of generalized least squares and minimum norm problems can be obtained as follows.

Theorem 8.6.1. *If B is positive definite then the linear system (8.6.1) is nonsingular and gives the condition for the solution of the two problems:*

$$\min_x \tfrac{1}{2}\|Ax - b\|^2_{B^{-1}} + c^T x, \tag{8.6.2}$$

$$\min_s \tfrac{1}{2}\|s - b\|_B, \quad \textit{subject to} \quad A^T s = c, \tag{8.6.3}$$

where $\|x\|^2_G = x^T G x$ for any symmetric positive definite matrix G.

Proof. If B is symmetric positive definite so is B^{-1}. The system (8.6.1) can be obtained by differentiating (8.6.2) to give

$$A^T B^{-1}(Ax - b) + c = 0,$$

and setting $s = B^{-1}(b - Ax)$. The system can also be obtained by differentiating the Lagrangian

$$L(x, s) = \frac{1}{2}s^T B s - s^T b + x^T(A^T s - c)$$

of (8.6.3), and equating to zero. Here x is the vector of Lagrange multipliers. □

Remark: Theorem 8.6.1 can be generalized to to the semidefinite case, see Gulliksson and Wedin [31, Theorem 3.2]. A case when B is indefinite and nonsingular is considered in Sec. 8.6.4

If we take $c = 0$ in Theorem 8.6.1, then the solution x gives the best linear unbiased estimate for the linear model

$$Ax + \epsilon = b, \quad \mathcal{V}(\epsilon) = \sigma^2 B^{-1}.$$

The standard linear least squares problem (8.1.1) is obtained by taking $B = I$.

Taking $B = I$ in problem (8.6.3), we have $s = r = b - Ax$ and this problem becomes

$$\min_s \frac{1}{2}\|r - b\|_2, \quad \text{subject to} \quad A^T r = c, \tag{8.6.4}$$

i.e. to find the point s closest to b in the set of solutions to the underdetermined linear system $A^T r = c$. This problem frequently occurs as a subproblem in linearly constrained optimization. Another application, for which $c = 0$, is in structural optimization, where A^T is called the equilibrium matrix, B the element flexibility matrix, y is the force, and x a Lagrange multiplier vector.

There are two different approaches to the solution of systems of the form (8.6.1). In the **range space method** the y variables are eliminated to obtain the **generalized normal equations**

$$A^T B^{-1} A x = A^T B^{-1} b - c. \tag{8.6.5}$$

From the assumptions in Theorem 8.6.1, it follows that the matrix $A^T B^{-1} A$ is symmetric, positive definite. The normal equations can be solved for x, and then y obtained by solving $Bs = b - Ax$. Setting $B = W$ and $c = 0$ in (8.6.2) gives a weighted linear least squares problem; see Sec. 8.6.2.

$$A^T B^{-1} A x = A^T B^{-1} b - c, \qquad y = B^{-1}(b - Ax). \tag{8.6.6}$$

For $B = I$ the first equation in (8.6.6) is the normal equations for the least squares problem. If B is positive definite then one way to solve these equations is to compute the Cholesky factorization $B = R^T R$ and then solve

$$\min_x \|R^{-1}(Ax - b)\|_2 \tag{8.6.7}$$

using the QR factorization of $R^{-1}A$. However, a more stable approach is to use a generalized QR (GQR) factorization of the matrix pair A, B; to be described in Sec. 8.6.3.

Using (8.6.5) the solution to problem (8.6.4) can be written

$$r = b - Ax = P_{\mathcal{N}(A^T)} b + A(A^T A)^{-1} c. \tag{8.6.8}$$

In particular, taking $b = 0$, this is the **minimum norm solution** of the system $A^T y = c$.

In the **null space method** the solution y to (8.6.6) is split as

$$y = y_1 + y_2, \quad y_1 \in \mathcal{R}(A), \quad y_2 \in \mathcal{N}(A^T). \tag{8.6.9}$$

Let y_1 be the minimum norm solution of $A^T y = c$. This can be computed using the QR factorization of A. If we set $Q = (\, Q_1 \quad Q_2 \,)$ then

$$y_1 = Q_1 z_1, \quad z_1 = R^{-T} c.$$

Next y_2 is obtained by solving the reduced system

$$Q_2^T B Q_2 z_2 = Q_2^T (b - B y_1), \qquad y_2 = Q_2 z_2. \tag{8.6.10}$$

Finally, form

$$y = Q_1 z_1 + Q_2 z_2 \quad \text{and} \quad x = R^{-1} Q_1^T (b - Vy).$$

In the special case that $B = I$ the generalized least squares problems associated with (8.6.1) simplify to

$$\min_y \|y - b\|_2^2 \quad \text{subject to} \quad A^T y = c, \tag{8.6.11}$$

$$\min_x \{\|b - Ax\|_2^2 + 2c^T x\}. \tag{8.6.12}$$

When a Householder QR factorization is available the algorithm is as follows:

$$z = R^{-T} c, \qquad \begin{pmatrix} d \\ f \end{pmatrix} = Q^T b, \qquad r = Q \begin{pmatrix} z \\ f \end{pmatrix}, \qquad x = R^{-1}(d - z).$$

Assuming that the matrix $M = A^T B^{-1} A$ has rank n, a first order perturbation analysis for the generalized least squares problem can be obtained. We assume that B is not perturbed and for simplicity take $c = 0$. Proceeding as in Sec. 8.1.5, we denote the perturbed data $A + \delta A$ and $b + \delta b$ and the perturbed solution $x + \delta x$ and $s + \delta s$.

The perturbed solution satisfies the system

$$\begin{pmatrix} B & A + \delta A \\ (A + \delta A)^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{s} \\ \tilde{x} \end{pmatrix} = \begin{pmatrix} b + \delta b \\ 0 \end{pmatrix}. \tag{8.6.13}$$

Subtracting the equations (8.6.13) and neglecting second order quantities the perturbations $\delta s = B^{-1}\delta r$ and δx satisfy the linear system

$$\begin{pmatrix} B & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \delta s \\ \delta x \end{pmatrix} = \begin{pmatrix} \delta b - \delta A x \\ -\delta A^T s \end{pmatrix}. \tag{8.6.14}$$

From the Schur–Banachiewicz formula (see Sec. 7.1.5) it follows that the inverse of the matrix in this system equals

$$\begin{pmatrix} B & A \\ A^T & 0 \end{pmatrix}^{-1} = \begin{pmatrix} (I - B^{-1}AM^{-1}A^T)B^{-1} & B^{-1}AM^{-1} \\ M^{-1}A^T B^{-1} & -M^{-1} \end{pmatrix}. \tag{8.6.15}$$

where $M = A^T B^{-1} A$ is the negative Schur complement. Hence we obtain

$$\delta x = M^{-1} A^T B^{-1} (\delta b - \delta A\, \tilde{x}) + M^{-1} \delta A^T \tilde{s}, \tag{8.6.16}$$

$$\delta r = (B - AM^{-1}A^T) B^{-1} (\delta b - \delta A\, \tilde{x}) - AM^{-1} \delta A^T \tilde{s}, \tag{8.6.17}$$

Taking norms in (8.6.17) and (8.6.17) we obtain

$$\|\delta x\|_2 \lessapprox \|M^{-1} A^T B^{-1}\| \left(\|\delta b\| + \|\delta A\|\, \|x\|\right) + \|M^{-1}\|\, \|\delta A\|\, \|s\|, \tag{8.6.18}$$

$$\|\delta r\| \lessapprox \|(B - AM^{-1}A^T)\|\, \|B^{-1}\| (\|\delta b\| + \|\delta A\|\, \|x\|) \tag{8.6.19}$$

$$+ \|AM^{-1}\|\, \|\delta A\| \|s\|, \tag{8.6.20}$$

8.6.2 Weighted Problems

We now consider a simple special case of the generalized least squares problem. In the general univariate linear model (8.1.5) the covariance matrix W is a positive diagonal matrix

$$W = \sigma^2 \mathrm{diag}\,(w_1, w_2, \ldots, w_m) > 0.$$

The corresponding problem then is the **weighted** linear least squares problem (8.1.5)

$$\min_x \|D(Ax - b)\|_2, \quad D = W^{-1/2} = \mathrm{diag}\,(d_1, d_2, \ldots, d_m). \tag{8.6.21}$$

When the ith component of the error vector in the linear model has small variance then $d_i = 1/\sqrt{w_{ii}}$ will be large. In the limit when some d_i tend to infinity, the corresponding ith equation becomes a linear constraint.

We assume in the following that the matrix A is row equilibrated, that is,

$$\max_{1 \le j \le n} |a_{ij}| = 1, \quad i = 1 : m.$$

and that the rows of A are ordered so that the weights satisfy

$$\infty > d_1 \ge d_2 \ge \cdots \ge d_m > 0. \tag{8.6.22}$$

We call a weighted least squares problems **stiff** if $d_1 \gg d_m$; see Example 8.2.2. For stiff problems the condition number $\kappa(DA)$ will be large. An upper bound is given by

$$\kappa(DA) \le \kappa(D)\kappa(A) = \gamma\kappa(A).$$

It is important to note that this does *not* mean that the problem of computing x from given data $\{D, A, b\}$ necessarily is ill-conditioned. Problems with extremely ill-conditioned weight matrices arise, e.g., in electrical networks, certain classes of finite element problems, and interior point methods for constrained optimization.

In many cases it is possible to compute $\tilde{A} = DA$, $\tilde{b} = Db$ and solve this as a standard least squares problem

$$\min_x \|\tilde{A}x - \tilde{b}\|_2.$$

However, if the weights $d_1, \ldots, d_m$ vary widely in magnitude this is not in general a numerically stable approach. Special care may be needed in solving stiff weighted linear least squares problems. In general the method of normal equations is not well suited for solving stiff problems. To illustrate this, we consider the important special case where only the first p equations are weighted:

$$\min_x \left\| \begin{pmatrix} \gamma A_1 \\ A_2 \end{pmatrix} x - \begin{pmatrix} \gamma b_1 \\ b_2 \end{pmatrix} \right\|_2^2, \tag{8.6.23}$$

$A_1 \in \mathbf{R}^{p \times n}$ and $A_2 \in \mathbf{R}^{(m-p) \times n}$. Such problems occur, for example, when the method of weighting is used to solve least squares problems with the linear equality

constraints $A_1x = b_1$; see Section 5.1.4. For this problem the matrix of normal equations becomes

$$B = (\,\gamma A_1^T \quad A_2^T\,)\begin{pmatrix}\gamma A_1\\ A_2\end{pmatrix} = \gamma^2 A_1^T A_1 + A_2^T A_2.$$

If $\gamma > u^{-1/2}$ (u is the unit roundoff) and $A_1^T A_1$ is dense, then $B = A^T A$ will be completely dominated by the first term and the data contained in A_2 may be lost. However, if the number p of very accurate observations is less than n, then the solution depends critically on the less precise data in A_2. (The matrix in Example 2.2.1 is of this type.) We conclude that for weighted least squares problems with $\gamma \gg 1$ the method of normal equations generally is not well behaved.

We now consider the use of methods based on the QR decomposition of A for solving weighted problems. We first examine the Householder QR method, and show by an example that this method can give poor accuracy for stiff problems unless the algorithm is extended to include *row interchanges*.

Example 8.6.1.

Consider the least squares problem ([51]) with

$$A = \begin{pmatrix}0 & 2 & 1\\ \gamma & \gamma & 0\\ \gamma & 0 & \gamma\\ 0 & 1 & 1\end{pmatrix}, \qquad b = \begin{pmatrix}3\\ 2\gamma\\ 2\gamma\\ 2\end{pmatrix}.$$

The exact solution is equal to $x = (1, 1, 1)$. Using exact arithmetic we obtain after the first step of QR decomposition of A by Householder transformations the reduced matrix

$$\tilde{A}^{(2)} = \begin{pmatrix}\frac{1}{2}\gamma - 2^{1/2} & -\frac{1}{2}\gamma - 2^{-1/2}\\ -\frac{1}{2}\gamma - 2^{1/2} & \frac{1}{2}\gamma - 2^{-1/2}\\ 1 & 1\end{pmatrix}.$$

If $\gamma > u^{-1}$ the terms $-2^{1/2}$ and $-2^{-1/2}$ in the first and second rows are lost. However, this is equivalent to the loss of all information present in the first row of A. This loss is disastrous because the number of rows containing large elements is less than the number of components in x, so there is a substantial dependence of the solution x on the first row of A. (However, compared to the method of normal equations, which fails already when $\gamma > u^{-1/2}$, this is an improvement!)

The Householder algorithm can be extended to include *row interchanges*. In each step a pivot column is first selected in the reduced matrix, and then the element of largest absolute value in the pivot column is permuted to the top. The resulting algorithm has good stability properties for stiff problems as well.

There is no need to perform row pivoting in Householder QR, provided that an initial **row sorting** is performed, where the rows are sorted after decreasing so that

$$\max_j |a_{1j}| \geq \max_j |a_{1j}| \geq \cdots \geq \max_j |a_{1j}|. \tag{8.6.24}$$

For example, in Example 8.6.1 the two large rows will be permuted to the top of the matrix A. The Householder algorithm then works well without any further row interchanges.

The stability of row sorting has been shown by Cox and Higham [18]. Note that row sorting has the advantage over row pivoting in that after sorting the rows any library routine can be used for the QR factorization. In particular this allows for the use of BLAS 3 subroutines, which is not the case for row pivoting.

It is also essential that *column pivoting* is performed when QR decomposition is used for weighted problems. To illustrate the need for column pivoting, consider an example of the form (8.6.23), where

$$A_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix},$$

Then stability is lost without column pivoting because the first two columns of the matrix A_1 are linearly dependent.

When column pivoting is introduced this difficulty disappears. With QR factorization with **complete pivoting** we will mean that both row sorting (or row pivoting) and column pivoting is used.

Another suitable transformation for weighted problems is to make a preliminary LU factorization of the matrix A. If the problem has the form (8.6.23) with $\operatorname{rank}(A_1) = p$, then p steps of Gaussian elimination are performed on the weighted system using row and column pivoting. The resulting factorization can be written

$$\Pi_r \begin{pmatrix} \gamma A_1 \\ A_2 \end{pmatrix} \Pi_c = LDU, \tag{8.6.25}$$

where Π_r and Π_c are permutation matrices, and

$$L = \begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix} \in \mathbf{R}^{m\times n}, \quad U = \begin{pmatrix} U_{11} & U_{12} \\ & I \end{pmatrix} \in \mathbf{R}^{n\times n}.$$

Here $L_{11} \in \mathbf{R}^{p\times p}$ is unit lower triangular, and $U_{11} \in \mathbf{R}^{p\times p}$ is unit upper triangular. Assuming that A has full rank, D is nonsingular. Then (4.4.1) is equivalent to

$$\min_y \|Ly - \Pi_r b\|_2, \qquad DU\Pi_c^T x = y.$$

The least squares problem in y is usually well-conditioned, since any ill-conditioning from the weights is usually reflected in D. We illustrate the method in a simple example. For a fuller treatment of weighted and the general linear model, see Björck [11, Chap.3].

Example 8.6.2. In Example 8.2.2 it was shown that the method of normal equations can fail. After multiplication with $\gamma = \epsilon^{-1}$ this becomes

$$A = \begin{pmatrix} 1 & 1 & 1 \\ \epsilon & & \\ & \epsilon & \\ & & \epsilon \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

which is of the form (8.6.23) with $p = 1$. After one step of Gaussian elimination we have the factorization $A = LDU$, where

$$L = \begin{pmatrix} 1 & & \\ \epsilon & -1 & -1 \\ & 1 & \\ & & 1 \end{pmatrix}, \qquad D = \begin{pmatrix} 1 & & \\ & \epsilon & \\ & & \epsilon \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & 1 & 1 \\ & 1 & \\ & & 1 \end{pmatrix}.$$

As is easily verified L and U are well-conditioned. Setting $DUx = y$, the solution can be accurately computed by first solving the normal equations $L^T Ly = L^T b$ for y and then finding x by back-substitution and scaling. □

8.6.3 Generalized Orthogonal Decompositions

The motivation for introducing different generalizations of orthogonal decompositions is basically to avoid the explicit computation of matrix products and quotients of matrices. For example, let A and B be square and nonsingular matrices and assume we need the SVD of AB^{-1} (or AB). Then the explicit calculation of AB^{-1} (or AB) may result in a loss of precision and should be avoided.

Consider a pair of matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times p}$. The generalized QR (GQR) decomposition of A and B is written

$$A = QR, \qquad B = QTZ, \tag{8.6.26}$$

where $Q \in \mathbb{R}^{m\times m}$ and $Z \in \mathbb{R}^{p\times p}$ are orthogonal matrices and R and T have one of the forms

$$R = \begin{pmatrix} R_{11} \\ 0 \end{pmatrix} \quad (m \geq n), \qquad R = \begin{pmatrix} R_{11} & R_{12} \end{pmatrix} \quad (m < n), \tag{8.6.27}$$

and

$$T = \begin{pmatrix} 0 & T_{12} \end{pmatrix} \quad (m \leq p), \qquad T = \begin{pmatrix} T_{11} \\ T_{21} \end{pmatrix} \quad (m > p). \tag{8.6.28}$$

If B is square and nonsingular GQR implicitly gives the QR factorization of $B^{-1}A$. There is also a similar generalized RQ factorization related to the QR factorization of AB^{-1}. Routines for computing a GQR decomposition of are included in LAPACK. These decompositions allow the solution of very general formulations of several least squares problems.

8.6.4 Indefinite Least Squares

The indefinite least squares problem (ILS) has the form

$$\min_x (b - Ax)^T J(b - Ax), \tag{8.6.29}$$

where $A \in \mathbf{R}^{m\times n}$, $m \geq n$, and $b \in \mathbf{R}^m$ are given and J is a **signature matrix**, i.e. a diagonal matrix with elements equal to ± 1. In the following we assume for

simplicity that

$$J = \begin{pmatrix} I_p & 0 \\ 0 & -I_q \end{pmatrix}, \quad p+q = m, \tag{8.6.30}$$

While the standard least squares is obtained if $p = 0$ or $q = 0$, for $pq \neq 0$ the problem is to minimize an indefinite quadratic form.

The normal equations

$$A^T J(b - Ax) = 0 \tag{8.6.31}$$

give first order conditions for optimality. In the following we assume that the Hessian matrix $A^T JA$ is positive definite. Then the ILS problem has a unique solution.

Chandrasekaran, Gu and Sayed [16] proposed a QR-Cholesky method for solving the ILS problem. It uses a QR factorization

$$A = QR = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} R, \quad Q_1 \in \mathbf{R}^{p\times n}, \quad Q_2 \in \mathbf{R}^{q\times n}. \tag{8.6.32}$$

Then

$$A^T JA = R^T (Q_1^T Q_1 - Q_2^T Q_2) R,$$

and it follows that R is nonsingular and $Q_1^T Q_1 - Q_2^T Q_2$ is positive definite. Using (8.6.32) the normal equations (8.6.31) can be written

$$(Q_1^T Q_1 - Q_2^T Q_2) Rx = Q^T Jb. \tag{8.6.33}$$

Using the Cholesky factorization $Q_1^T Q_1 - Q_2^T Q_2 = U^T U$, this becomes

$$U^T URx = Q^T Jb,$$

which can be solved by one forward and two backward substitutions.

The operation count for the QR–Cholesky algorithm is approximately $n^2(5m - n)$, which can be compared to the normal equations $n^2(m + n/3)$.

Sometimes it is useful to consider **hyperbolic rotations** $\breve{G}$ of the form

$$\breve{G} = \begin{pmatrix} c & -s \\ -s & c \end{pmatrix}, \quad c = \cosh\theta, \quad s = \sinh\theta, \tag{8.6.34}$$

and $c^2 - s^2 = 1$. The matrix $\breve{G}$ is S-orthogonal, $\breve{G}^T S\breve{G} = I$, for the signature matrix $S = \operatorname{diag}(1, -1)$.

A hyperbolic rotation can be used to zero a selected component in a vector. Provided that $|\alpha| > |\beta|$ and

$$s = \beta/\alpha, \quad c = \sqrt{(1+s)(1-s)}, \quad \sigma = \alpha c,$$

we have $s^2 + c^2 = 1$ and

$$\breve{G}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{c}\begin{pmatrix} 1 & -s \\ -s & 1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \sigma \\ 0 \end{pmatrix}.$$

A matrix representing a rotation a hyperbolic rotation in $\mathbf{R}^m$ in the plane spanned by the unit vectors e_i and e_j, $i < j$, is obtained as for Givens' rotations; see (8.3.34).

Thee condition number of $\breve{G}$ in (8.6.34) is not bounded, and to form a product $\breve{G}x$ the straightforward way is not numerically stable. Instead we note the equivalence of

$$\breve{G}\begin{pmatrix} x_i \\ x_j \end{pmatrix} = \begin{pmatrix} \hat{x}_i \\ \hat{x}_j \end{pmatrix}, \qquad G\begin{pmatrix} \hat{x}_i \\ x_j \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} \hat{x}_i \\ x_j \end{pmatrix} = \begin{pmatrix} x_i \\ \hat{x}_j \end{pmatrix}, \tag{8.6.35}$$

where G is an orthogonal Givens rotation. The mixed method where $\hat{x}_i$ is determined from the hyperbolic rotation and then $\hat{x}_j$ from the equivalent Givens rotation

$$\hat{x}_i = (x_i - sx_j)/c, \qquad \hat{x}_j = -s\hat{x}_i + cx_j, \tag{8.6.36}$$

has been shown to be numerically more stable.

A hyperbolic QR factorization method for solving the indefinite least squares problem has been devised by Bojanczyk, Higham and Patel [13]. We first use Householder transformations to compute the factorization

$$\begin{pmatrix} Q_1^T & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} R_1 \\ 0 \\ A_2 \end{pmatrix}.$$

where A has been partitioned conformally with J.

8.6.5 Orthogonal Regression

We consider here the following **orthogonal regression** problem. Let $y_i \in \mathbf{R}^n$, $i = 1 : m$, be $m > n$ given points. We want to determine a hyperplane M in $\mathbf{R}^n$ such that the sum of squares of the orthogonal distances from the given points to M is minimized. The equation for the hyperplane can be written

$$c^T z = h, \quad z, c \in \mathbf{R}^n, \quad \|c\|_2 = 1,$$

where $c \in \mathbf{R}^n$ is the normal vector of M, and $|h|$ is the orthogonal distance form the origin to the plane. Then the orthogonal projections of the points y_i onto M are given by

$$z_i = y_i - (c^T y_i - h)c. \tag{8.6.37}$$

It is readily verified that the point z_i lies on M and the residual $(z_i - y_i)$ is parallel to c and hence orthogonal to M. It follows that the problem is equivalent to minimizing

$$\sum_{i=1}^{m}(c^T y_i - h)^2, \quad \text{subject to} \quad \|c\|_2 = 1.$$

If we put $Y = (y_1, \ldots, y_m) \in \mathbf{R}^{n\times m}$ and $e = (1, \ldots, 1)^T$, this problem can be written in matrix form

$$\min_{c,h} \left\| \begin{pmatrix} Y^T & -e \end{pmatrix}\begin{pmatrix} c \\ h \end{pmatrix} \right\|_2, \quad \text{subject to} \quad \|c\|_2 = 1. \tag{8.6.38}$$

For a fixed c, this expression is minimized when the residual vector $(Y^Tc - he)$ is orthogonal to e, that is $e^T(Y^Tc - he) = e^TY^Tc - he^Te = 0$. Since $e^Te = m$ it follows that

$$h = \frac{1}{m}c^TYe = c^T\bar{y}, \qquad \bar{y} = \frac{1}{m}Ye, \tag{8.6.39}$$

where $\bar{y}$ is the mean value of the given points y_i. Hence h is determined by the condition that the mean value $\bar{y}$ lies on the optimal plane M.

We now subtract the mean value $\bar{y}$ from the each given point, and form the matrix

$$\bar{Y} = (\bar{y}_1, \ldots, \bar{y}_m), \qquad \bar{y}_i = y_i - \bar{y}, \quad i = 1, \ldots, m.$$

Since by (8.6.39)

$$(Y^T, -e)\begin{pmatrix} c \\ h \end{pmatrix} = Y^Tc - e\bar{y}^Tc = (Y^T - e\bar{y}^T)c = \bar{Y}^Tc,$$

problem (8.6.38) is equivalent to

$$\min_c \|\bar{Y}^Tc\|_2, \qquad \|c\|_2 = 1 \tag{8.6.40}$$

By the min-max characterization of the singular values (Theorem 8.1.11) a solution to (8.6.40) is $c = v_n$, where v_n is a right singular vector of $\bar{Y}^T$ corresponding to the singular value σ_n. Hence, a solution to problem (8.6.38) is given by

$$c = v_n, \qquad h = v_n^T\bar{y}, \qquad \sum_{i=1}^{m}(v_n^Ty_i - h)^2 = \sigma_n,$$

The fitted points $z_i \in M$ are obtained from

$$z_i = \bar{y}_i - (v_n^T\bar{y}_i)v_n + \bar{y},$$

i.e., by first orthogonalizing the shifted points $\bar{y}_i$ against v_n, and then adding the mean value back.

Note that in contrast to the TLS problem the orthogonal regression problem always has a solution. The solution is unique when $\sigma_{n-1} > \sigma_n$, and the minimum sum of squares equals σ_n^2. We have $\sigma_n = 0$, if and only if the given points y_i, $i = 1, \ldots, m$ all lie on the hyperplane M. In the extreme case, all points coincide and then $\bar{Y} = 0$, and any plane going through $\bar{y}$ is a solution.

The above method solves the problem of fitting a $(n-1)$ dimensional linear manifold to a given set of points in R. It is readily generalized to the fitting of an $(n-p)$ dimensional manifold by orthogonalizing the shifted points y against the p left singular vectors of Y corresponding to p smallest singular values. A least squares problem that often arises is to fit to given data points a geometrical element, which may be defined in implicit form. For example, the problem of fitting circles, ellipses, spheres, and cylinders arises in applications such as computer graphics, coordinate meteorology, and statistics. Such problems are nonlinear and will be discussed in Sec. 11.4.7.

Example 8.6.3.

Suppose we want to fit by orthogonal regression m pair of points $(x_i, y_i) \in \mathbf{R}^2$, $i = 1, \ldots, m$, to a straight line

$$cx + sy = h, \quad c^2 + s^2 = 1.$$

First compute the mean values of x_i and y_i and the QR factorization of the matrix of shifted points

$$\bar{Y}^T = \begin{pmatrix} \bar{x}_1 & \bar{y}_1 \\ \bar{x}_2 & \bar{y}_2 \\ \vdots & \vdots \\ \bar{x}_m & \bar{y}_m \end{pmatrix} = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where R is an upper triangular 2×2 matrix. Since the singular values and right singular vectors of $\bar{Y}^T$ and R are the same, it suffices to compute the SVD

$$R = \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix} = (u_1 \ u_2) \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \begin{pmatrix} v_1^T \\ v_2^T \end{pmatrix},$$

where $\sigma_1 \geq \sigma_2 \geq 0$. (A stable algorithm for computing the SVD of an upper triangular matrix is given in Algorithm 9.4.2; see also Problem 9.4.5.) Then the coefficients in the equation of the straight line are given by

$$(c \ s) = v_2^T, \qquad h = v_2^T \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}.$$

If $\sigma_2 = 0$ but $\sigma_1 > 0$ the matrix $\bar{Y}$ has rank one. In this case the given points lie on a straight line. If $\sigma_1 = \sigma_2 = 0$, then $\bar{Y} = 0$, and $x_i = \bar{x}$, $y_i = \bar{y}$ for all $i = 1, \ldots, m$. Note that u_2 is uniquely determined if and only if $\sigma_1 \neq \sigma_2$. It is left to the reader to discuss the case $\sigma_1 = \sigma_2 \neq 0$!

8.6.6 Linear Equality Constraints

In some least squares problems in which the unknowns are required to satisfy a system of linear equations exactly. One source of such problems is in curve and surface fitting, where the curve is required to interpolate certain data points.

Given matrices $A \in \mathbf{R}^{m \times n}$ and $B \in \mathbf{R}^{p \times n}$ we consider the problem **LSE** to find a vector $x \in \mathbf{R}^n$ which solves

$$\min_x \|Ax - b\|_2 \quad \text{subject to} \quad Bx = d. \tag{8.6.41}$$

A solution to problem (8.6.41) exists if and only if the linear system $Bx = d$ is consistent. If $\operatorname{rank}(B) = p$ then B has linearly independent rows, and $Bx = d$ is consistent for any right hand side d. A solution to problem (8.6.41) is unique if and only if the null spaces of A and B intersect only trivially, i.e., if $\mathcal{N}(A) \cap \mathcal{N}(B) = \{0\}$, or equivalently

$$\operatorname{rank} \begin{pmatrix} A \\ B \end{pmatrix} = n. \tag{8.6.42}$$

If (8.6.42) is not satisfied then there is a vector $z \neq 0$ such that $Az = Bz = 0$. Hence if x solves (8.6.41) then $x + z$ is a different solution. In the following we therefore assume that $\operatorname{rank}(B) = p$ and that (8.6.42) is satisfied.

A robust algorithm for problem LSE should check for possible inconsistency of the constraints $Bx = d$. If it is not known a priori that the constraints are consistent, then problem LSE may be reformulated as a sequential least squares problem

$$\min_{x \in S} \|Ax - b\|_2, \qquad S = \{x \mid \|Bx - d\|_2 = \min\ \}. \tag{8.6.43}$$

The most natural way to solve problem LSE is to derive an equivalent unconstrained least squares problem of lower dimension. There are basically two different ways to perform this reduction: **direct elimination** and **the null space method**. We describe both these methods below.

In the method of direct elimination we start by reducing the matrix B to upper trapezoidal form. It is essential that column pivoting is used in this step. In order to be able to solve also the more general problem (8.6.43) we will compute a QR factorization of B. By Theorem 8.4.1 (see next section) there is an orthogonal matrix $U \in \mathbf{R}^{p \times p}$ and a permutation matrix Π_B such that

$$Q_B^T B \Pi_B = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix}, \qquad R_{11} \in \mathbf{R}^{r \times r}, \tag{8.6.44}$$

where $r = \operatorname{rank}(B) \leq p$ and R_{11} is upper triangular and nonsingular. Using this factorization, and setting $\bar{x} = \Pi_B^T x$, the constraints become

$$(R_{11}, R_{12})\bar{x} = R_{11}\bar{x}_1 + R_{12}\bar{x}_2 = \bar{d}_1, \qquad \bar{d} = Q_B^T d = \begin{pmatrix} \bar{d}_1 \\ \bar{d}_2 \end{pmatrix}, \tag{8.6.45}$$

where $\bar{d}_2 = 0$ if and only if the constraints are consistent. If we apply the permutation Π_B also to the columns of A and partition the resulting matrix conformally with (8.6.44), $\bar{A}\Pi_B = (A_1, A_2)$. then $Ax - b = A_1\bar{x}_1 + A_2\bar{x}_2 - b$. Solving (8.6.45) for $\bar{x}_1 = R_{11}^{-1}(\bar{d}_1 - R_{12}\bar{x}_2)$, and substituting, we find that the unconstrained least squares problem

$$\begin{aligned} &\min_{\bar{x}_2} \|\hat{A}_2\bar{x}_2 - \hat{b}\|_2, \qquad \hat{A}_2 \in \mathbf{R}^{m \times (n-r)} \\ &\hat{A}_2 = \bar{A}_2 - \bar{A}_1 R_{11}^{-1} R_{12}, \qquad \hat{b} = b - \bar{A}_1 R_{11}^{-1}\bar{d}_1. \end{aligned} \tag{8.6.46}$$

is equivalent to the original problem LSE. Here $\hat{A}_2$ is the Schur complement of R_{11} in

$$\begin{pmatrix} R_{11} & R_{12} \\ \bar{A}_1 & \bar{A}_2 \end{pmatrix}.$$

It can be shown that if the condition in (8.6.42) is satisfied, then $\operatorname{rank}(A_2) = r$. Hence the unconstrained problem has a unique solution, which can be computed from the QR factorization of $\hat{A}_2$.

In the null-space method, assuming that $\operatorname{rank}(B) = p$, we compute the QR factorization

$$B^T = U \begin{pmatrix} R_B \\ 0 \end{pmatrix}, \qquad R_B \in \mathbf{R}^{p \times p}, \tag{8.6.47}$$

where R_B is upper triangular and nonsingular. Using Theorem 8.3.7 we find that the general solution of the system $Bx = d$ can be written as

$$x = x_1 + Q_2 y_2, \qquad x_1 = B^{\dagger} d = Q_1 R_B^{-T} d. \tag{8.6.48}$$

where $U = (Q_1, Q_2)$, $Q_1 \in \mathbf{R}^{n\times p}$, and $Q_2 \in \mathbf{R}^{n\times(n-p)}$. (Note that Q_2 gives an orthogonal basis for the null space of B.) Hence, $Ax - b = Ax_1 + AQ_2y_2 - b$, $y_2 \in \mathbf{R}^{n-p}$, and it remains to solve the unconstrained least squares problem

$$\min_{y_2} \|(AQ_2)y_2 - (b - Ax_1)\|_2. \tag{8.6.49}$$

Let $y_2 = (AQ_2)^I(b - Ax_1)$ be the minimum length solution to (8.6.49), and let x be defined by (8.6.48). Then since $x_1 \perp Q_2y_2$ it follows that

$$\|x\|_2^2 = \|x_1\|_2^2 + \|Q_2y_2\|_2^2 = \|x_1\|_2^2 + \|y_2\|_2^2$$

and x is the minimum norm solution to problem LSE.

If (8.6.42) is satisfied it follows that rank $(AQ_2) = n-p$. Then we can compute the QR factorization

$$Q_A^T(AQ_2) = \begin{pmatrix} R_A \\ 0 \end{pmatrix},$$

where R_A is upper triangular and nonsingular. The unique solution to (8.6.49) can then be computed from

$$R_A y_2 = c_1, \qquad c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = Q_A^T(b - Ax_1), \tag{8.6.50}$$

and we finally obtain $x = x_1 + Q_2y_2$, the unique solution to problem LSE.

The method of direct elimination and the null space method both have good numerical stability. The operation count for the method of direct elimination is slightly lower because Gaussian elimination is used to derive the reduced unconstrained problem.

Review Questions

1. What is meant by a saddle-point system? Which two optimization problems give rise to saddle-point systems?
2. Show the equivalence of the hyperbolic and the Givens rotations in (8.6.35).

Problems

1. Consider the overdetermined linear system $Ax = b$ in Example 8.2.2. Assume that $\epsilon^2 \le u$, where u is the unit roundoff, so that $fl(1 + \epsilon^2) = 1$.

(a) Show that the condition number of A is $\kappa = \epsilon^{-1}\sqrt{3+\epsilon^2} \approx \epsilon^{-1}\sqrt{3}$.

(b) Show that if no other rounding errors are made then the maximum deviation from orthogonality of the columns computed by CGS and MGS, respectively, are

$$\text{CGS}: \quad |q_3^T q_2| = 1/2, \qquad \text{MGS}: \quad |q_3^T q_1| = \frac{\epsilon}{\sqrt{6}} \leq \frac{\kappa}{3\sqrt{3}}u.$$

Note that for CGS orthogonality has been completely lost!

2. Assume that $A \in \mathbf{R}^{m\times m}$ is symmetric and positive definite and $B \in \mathbf{R}^{m\times n}$ a matrix with full column rank. Show that

$$M = \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ B^T A^{-1} & I \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & -S \end{pmatrix} \begin{pmatrix} I & A^{-1}B \\ 0 & I \end{pmatrix},$$

where $S = B^T A^{-1} B$ is the Schur complement (cf. (6.2.12)). Conclude that M is indefinite! (M is called a saddle point matrix.)

8.7 Total Least Squares

8.7.1 The Total Least Squares Problem

In the standard linear model (8.1.3) it is assumed that the vector $b \in \mathcal{R}^m$ is related to the unknown parameter vector $x \in \mathcal{R}^n$ by a linear relation $Ax = b + e$, where $A \in \mathcal{R}^{m\times n}$ is an exactly known matrix and e a vector of random errors. If the components of e are uncorrelated, have zero means and the same variance, then by the Gauss–Markoff theorem (Theorem 8.1.4) the best unbiased estimate of x is obtained by solving the least squares problem

$$\min_x \|r\|_2, \qquad Ax = b + r. \tag{8.7.1}$$

The assumption in the least squares problem that all errors are confined to the right hand side b is frequently unrealistic, and sampling or modeling errors often will affect also the matrix A. In the **errors-in-variables model** it is assumed that a linear relation

$$(A + E)x = b + r,$$

where the rows of the errors $(E,\ r)$ are *independently and identically distributed with zero mean and the same variance*. If this assumption is not satisfied it might be possible to find scaling matrices $D = \text{diag}\,(d_1, \ldots, d_m)$, $T = \text{diag}\,(d_1, \ldots, d_{n+1})$, such that $D(A,\ b)T$ satisfies this assumptions.

Estimates of the unknown parameters x in this model can be obtained from the solution of the **total least squares** (TLS) problem[4]

$$\min_{E,\,r} \|(r,\ E)\|_F, \qquad (A + E)x = b + r, \tag{8.7.2}$$

[4]The term "total least squares problem" was coined by Golub and Van Loan in [28]. The concept has been independently developed in other areas. For example, in statistics this is also known as "latent root regression".

where $\|\cdot\|_F$ denotes the Frobenius matrix norm defined by

$$\|A\|_F^2 = \sum_{i,j} a_{ij}^2 = \text{trace}\,(A^T A).$$

The constraint in (8.7.2) implies that $b + r \in \mathcal{R}(A + E)$. Thus the total least squares is equivalent to the problem of finding the "nearest" compatible linear system, where the distance is measured by the Frobenius norm. If a minimizing perturbation $(E,\ r)$ has been found for the problem (8.7.2) then any x satisfying $(A + E)x = b + r$ is said to solve the TLS problem.

The TLS solution will depend on the scaling of the data $(A,\ b)$. In the following we assume that this scaling has been carried out in advance, so that any statistical knowledge of the perturbations has been taken into account. In particular, the TLS solution depends on the relative scaling of A and b. If we scale x and b by a factor γ we obtain the **scaled TLS problem**

$$\min_{E,\,r} \|(E,\ \gamma r)\|_F \qquad (A + E)x = b + r.$$

Clearly, when γ is small perturbations in b will be favored. In the limit when $\gamma \to 0$ we get the ordinary least squares problem. Similarly, when γ is large perturbations in A will be favored. In the limit when $1/\gamma \to 0$, this leads to the **data least squares** (DLS) problem

$$\min_{E} \|E\|_F, \qquad (A + E)x = b, \tag{8.7.3}$$

where it is assumed that the errors in the data is confined to the matrix A.

8.7.2 Total Least Squares Problem and the SVD

In the following we assume that $b \notin \mathcal{R}(A)$, for otherwise the system is consistent. The constraint in (8.7.2) can be written

$$(\,b + r \quad A + E\,)\begin{pmatrix} -1 \\ x \end{pmatrix} = 0.$$

This constraint is satisfied if the matrix $(b+r\ A+E)$ is rank deficient and $(\,-1 \quad x\,)^T$ lies in its nullspace. Hence the TLS problem involves finding a perturbation matrix having minimal Frobenius norm, which lowers the rank of the matrix $(\,b \quad A\,)$.

The total least squares problem can be analyzed in terms of the SVD

$$(\,b \quad A\,) = U\Sigma V^T = \sum_{i=1}^{k+1} \sigma_i u_i v_i^T, \tag{8.7.4}$$

where $\sigma_1 \geq \ldots \geq \sigma_n \geq \sigma_{n+1} \geq 0$ are the singular values of $(\,b \quad A\,)$. By Theorem 8.1.13 the singular values of $\hat{\sigma}_i$ of A interlace those of $(\,b \quad A\,)$, i.e.,

$$\sigma_1 \geq \hat{\sigma}_1 \geq \sigma_2 > \cdots \geq \sigma_n \geq \hat{\sigma}_n \geq \sigma_{n+1}. \tag{8.7.5}$$

Assume first that rank $(A) = n$ and that $\hat{\sigma}_n > \sigma_{n+1}$, which implies that $\sigma_n > \sigma_{n+1}$. Then by Theorem 8.1.14 the unique perturbation of minimum norm $\|\,(\,r \quad E\,)\,\|_F$ that makes $(A+E)x = b+r$ consistent is the rank one perturbation

$$(\,r \quad E\,) = -\sigma_{n+1} u_{n+1} v_{n+1}^T \tag{8.7.6}$$

for which $\min_{E,\,r} \|\,(\,r \quad E\,)\,\|_F = \sigma_{n+1}$. From (8.7.5), using the orthogonality of the right singular vectors we find that $\sigma_{n+1} u_{n+1} = (\,b \quad A\,)\,v_{n+1}$. Multiplying (8.7.6) from the right with v_{n+1} gives

$$(\,b \quad A\,)\,v_{n+1} = -\,(\,r \quad E\,)\,v_{n+1}. \tag{8.7.7}$$

Writing the relation $(A+E)x = b+r$ in the form

$$(\,b \quad A\,)\begin{pmatrix} 1 \\ -x \end{pmatrix} = -\,(\,r \quad E\,)\begin{pmatrix} 1 \\ -x \end{pmatrix}$$

and comparing with (8.7.7) it is easily seen that the TLS solution can be written in terms of the right singular vector v_{n+1} as

$$x = -\omega^{-1} y, \quad v_{n+1} = \begin{pmatrix} \omega \\ y \end{pmatrix}, \tag{8.7.8}$$

If $\omega = 0$ then the TLS problem has no solution. Note that this is the case if and only if b has no component along u_{n+1}. (This case can only occur when $\hat{\sigma}_n = \sigma_{n+1}$, since otherwise it can be shown that the TLS problem has a unique solution.) In this "nongeneric" case the theory and solution methods become more complicated.

Suppose now that σ_{n+1} is a repeated singular value,

$$\sigma_1 \geq > \cdots \geq \sigma_k \;\; > \hat{\sigma}_{k+1} = \cdots = \sigma_{n+1}.$$

and let $v = V_2 z$ be any unit vector in the subspace $\mathcal{R}(V_2)$, where $V_2 = (v_{k+1}, \ldots, v_{n+1})$ is the matrix consisting of the right singular vectors corresponding to the minimal singular values. Let Q be a Householder transformation such that.

$$V_2 Q = \begin{pmatrix} \omega & 0 \\ y & V_2' \end{pmatrix}$$

Then is $\omega \neq 0$ a TLS solution of minimum norm is given by (8.7.8). Otherwise we have a nongeneric problem.

One way to avoid the complications of nongeneric problems is to compute a regular core TLS problem by bidiagonalizing of the matrix $(\,b \quad A\,)$. This will be discussed in Sec. 8.7.4.

8.7.3 Conditioning of the TLS Problem

We now consider the conditioning of the total least squares problem and its relation to the least squares problem. We denote those solutions by x_{TLS} and x_{LS} respectively.

The TLS solution can also be characterized by

$$\begin{pmatrix} b^T b & b^T A \\ A^T b & A^T A \end{pmatrix} \begin{pmatrix} -1 \\ x_{TLS} \end{pmatrix} = \sigma_{n+1}^2 \begin{pmatrix} -1 \\ x_{TLS} \end{pmatrix}, \tag{8.7.9}$$

i.e., $(\,-1 \quad x_{TLS}\,)^T$ is an eigenvector corresponding to the smallest eigenvalue $\lambda_{n+1} = \sigma_{n+1}^2$ of the "square" of $(\,b \quad A\,)$. This eigenvector is characterized by the property that it minimizes the Rayleigh quotient, that is $\min_x \rho(x) = \sigma_{n+1}^2$, where

$$\rho(x) = \frac{(b - Ax)^T (b - Ax)}{x^T x + 1} = \frac{\|r\|_2^2}{\|x\|_2^2 + 1}, \tag{8.7.10}$$

This also shows that whereas the LS solution minimizes $\|r\|_2$ the TLS solution minimizes $\|r\|_2/(\|x\|_2^2 + 1)^{1/2}$.

From the last block row of (8.7.9) it follows that

$$(A^T A - \sigma_{n+1}^2 I) x_{TLS} = A^T b. \tag{8.7.11}$$

Hence, if we assume that $\hat{\sigma}_n > \sigma_{n+1}$ it follows that the matrix $(A^T A - \sigma_{n+1}^2 I)$ is symmetric positive definite, which ensures that the TLS problem has a unique solution.

This can be compared with the corresponding normal equations for the least squares solution x_{LS},

$$A^T A x_{LS} = A^T b. \tag{8.7.12}$$

In (8.7.11) a positive multiple of the unit matrix is *subtracted* from the matrix $A^T A$ of normal equations. Thus TLS can be considered as a *deregularizing* procedure. (Compare Sec. 8.4.1, where a multiple of the unit matrix was added to improve the conditioning.) Hence the TLS solution is always *worse conditioned* than the LS problem, From a statistical point of view this can be interpreted as removing the bias by subtracting the error covariance matrix (estimated by $\sigma_{n+1}^2 I$ from the data covariance matrix $A^T A$. Subtracting (8.7.12) from (8.7.12) we get

$$x_{TLS} - x_{LS} = \sigma_{n+1}^2 (A^T A - \sigma_{n+1}^2 I)^{-1} x_{LS}.$$

Taking norms we obtain

$$\frac{\|x_{TLS} - x_{LS}\|_2}{\|x_{LS}\|_2} \leq \frac{\sigma_{n+1}^2}{\hat{\sigma}_n^2 - \sigma_{n+1}^2},$$

which shows that when the difference $\hat{\sigma}_n - \sigma_{n+1} \ll \hat{\sigma}_n$ is small then the TLS solution can differ much from the LS solution. It can be shown that an approximate condition number for the TLS solution is

$$\kappa_{TLS} \approx \frac{\hat{\sigma}_1}{\hat{\sigma}_n - \sigma_{n+1}} = \kappa(A) \frac{\hat{\sigma}_n}{\hat{\sigma}_n - \sigma_{n+1}}. \tag{8.7.13}$$

When $\hat{\sigma}_n - \sigma_{n+1} \ll \hat{\sigma}_n$ the TLS condition number can be much worse than for the LS problem.

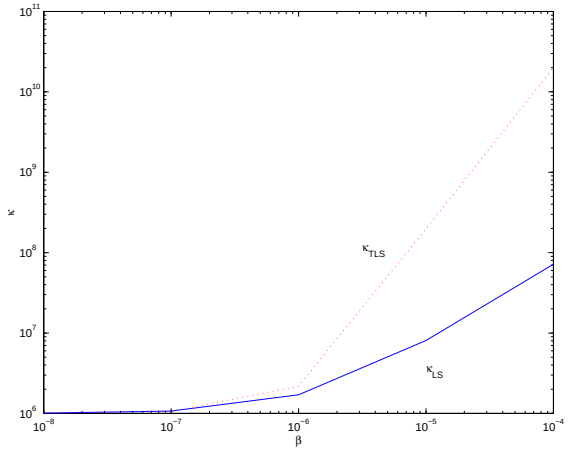

Figure 8.7.1. *Condition numbers κ_{LS} and κ_{TLS} as function of $\beta = \|r_{LS}\|_2$.*

Example 8.7.1.

Consider the overdetermined system

$$\begin{pmatrix} \hat{\sigma}_1 & 0 \\ 0 & \hat{\sigma}_2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \beta \end{pmatrix}. \tag{8.7.14}$$

Trivially, the LS solution is $x_{LS} = (c_1/\hat{\sigma}_1, c_2/\hat{\sigma}_2)^T$, $\|r_{LS}\|_2 = |\beta|$. If we take $\hat{\sigma}_1 = c_1 = 1$, $\hat{\sigma}_2 = c_2 = 10^{-6}$, then $x_{LS} = (1,\ 1)^T$ independent of β, and hence does not reflect the ill-conditioning of A. However,

$$\kappa_{LS}(A, b) = \kappa(A) \left(1 + \frac{\|r_{LS}\|_2}{\|\hat{\sigma}_1 x_{LS}\|_2}\right)$$

will increase proportionally to β. The TLS solution is of similar size as the LS solution as long as $|\beta| \leq \hat{\sigma}_2$. However, when $|\beta| \gg \hat{\sigma}_2$ then $\|x_{TLS}\|_2$ becomes large.

In Figure 8.7.1 the two condition numbers are plotted as a function of $\beta \in [10^{-8}, 10^{-4}]$. For $\beta > \hat{\sigma}_2$ the condition number κ_{TLS} grows proportionally to β^2. It can be verified that $\|x_{TLS}\|_2$ also grows proportionally to β^2.

Setting $c_1 = c_2 = 0$ gives $x_{LS} = 0$. If $|\beta| \geq \sigma_2(A)$, then $\sigma_2(A) = \sigma_3(A, b)$ and the TLS problem is nongeneric.

8.7.4 Bidiagonalization and TLS Problems.

Consider the total least squares (TLS) problem

$$\min_{E,r} \|(E,\ r)\|_F, \qquad (A + E)x = b + r.$$

It was shown in Sec. 8.4.5 that we can always find square orthogonal matrices $\tilde{U}_{k+1}$ and $\tilde{V}_k = P_1 P_2 \cdots P_k$, such that

$$\tilde{U}_{k+1}^T \begin{pmatrix} b & A\tilde{V}_k \end{pmatrix} = \begin{pmatrix} \beta_1 e_1 & B_k & 0 \\ 0 & 0 & A_k \end{pmatrix}, \tag{8.7.15}$$

where

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \in \mathbf{R}^{(k+1)\times k},$$

and

$$\beta_j \alpha_j \neq 0, \quad j = 1:k. \tag{8.7.16}$$

Setting $x = \tilde{V}_k \begin{pmatrix} y \\ z \end{pmatrix}$, the approximation problem $Ax \approx b$ then decomposes into the two subproblems

$$B_k y \approx \beta_1 e_1, \qquad A_k z \approx 0.$$

It seems reasonable to simply take $z = 0$, and separately solve the first subproblem, which is the minimally dimensioned **core subproblem**. Setting

$$V_k = \tilde{V}_k \begin{pmatrix} I_k \\ 0 \end{pmatrix}, \qquad U_{k+1} = \tilde{U}_{k+1} \begin{pmatrix} I_{k+1} \\ 0 \end{pmatrix},$$

it follows that

$$\begin{pmatrix} b & AV_k \end{pmatrix} = U_{k+1} \begin{pmatrix} \beta_1 e_1 & B_k \end{pmatrix}.$$

If $x = V_k y \in \mathcal{R}(V_k)$ then

$$(A + E)x = (A + E)V_k y = (U_{k+1} B_k + EV_k)y = \beta_1 U_{k+1} e_1 + r,$$

Hence the consistency relation $(A + E_k)x = b + r$ becomes

$$(B_k + F)y = \beta_1 e_1 + s, \quad F = U_{k+1}^T E V_k, \quad s = U_{k+1}^T r. \tag{8.7.17}$$

Using the orthogonality of U_{k+1} and V_k it follows that

$$\|(E, r)\|_F = \|(F, s)\|_F. \tag{8.7.18}$$

Hence to minimize $\|(E, r)\|_F$ we should take y_k to be the solution to the TLS core subproblem

$$\min_{F,s} \|(F,\ s)\|_F, \qquad (B_k + F)y = \beta_1 e_1 + s. \tag{8.7.19}$$

From (8.7.16) and Theorem 8.4.5 it follows that the singular values of the matrix B_k are simple and that the right hand side βe_1 has nonzero components along each

left singular vector. This TLS problem therefore must have a unique solution. Note that we can assume that $\beta_{k+1} \neq 0$, since otherwise the system is compatible.

To solve this subproblem we need to compute the SVD of the bidiagonal matrix

$$(\beta_1 e_1,\ B_k) = \begin{pmatrix} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & & \beta_3 & \ddots & \\ & & & \ddots & \alpha_k \\ & & & & \beta_{k+1} \end{pmatrix} \in \mathbf{R}^{(k+1)\times(k+1)}. \tag{8.7.20}$$

The SVD of this matrix

$$(\beta_1 e_1,\ B_k) = P\mathrm{diag}\,(\sigma_1, \ldots, \sigma_{k+1})Q^T, \quad P, Q \in \mathbf{R}^{(k+1)\times(k+1)}$$

can be computed, e.g., by the implicit QR-SVD algorithm; see Sec. 9.7.6. (Note that the first stage in this is a transformation to bidiagonal form, so the work in performing the reduction (8.7.15) has not been wasted!) Then with

$$q_{k+1} = Qe_{k+1} = \begin{pmatrix} \omega \\ z \end{pmatrix}.$$

Here it is always the case that $\omega \neq 0$ and the solution to the original TLS problem (8.7.19) equals

$$x_{TLS} = V_k y = -\omega^{-1} V_k z.$$

Further the norm of the perturbation equals

$$\min_{E,r} \|(E, r)\|_F = \sigma_{k+1}.$$

8.7.5 Some Generalized TLS Problems

We now consider the more general TLS problem with $d > 1$ right-hand sides

$$\min_{E,\ F} \| \begin{pmatrix} E & F \end{pmatrix} \|_F, \qquad (A + E)X = B + F, \tag{8.7.21}$$

where $B \in \mathbf{R}^{m\times d}$. The consistency relations can be written

$$\begin{pmatrix} B + F & A + E \end{pmatrix} \begin{pmatrix} -I_d \\ X \end{pmatrix} = 0,$$

Thus we now seek perturbations $(E,\ F)$ that reduces the rank of the matrix $(A,\ B)$ by d. We call this a multidimensional TLS problem. As remarked before, for this problem to be meaningful the rows of the error matrix $(E,\ F)$ should be independently and identically distributed with zero mean and the same variance.

Note that the multidimensional problem is different from solving d one-dimensional TLS problems with right-hand sides $b_1, \ldots, b_d$. This is because in the multidimensional problem we require that the matrix A be similarly perturbed for all right-hand sides. This is in contrast to the usual least squares solution and may lead to improved predicted power of the TLS solution.

The solution to the TLS problem with multiple right-hand sides can be expressed in terms of the SVD

$$(\, B \quad A\,) = (U_1\ U_2) \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}, \tag{8.7.22}$$

where $\Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$, $\Sigma_2 = \operatorname{diag}(\sigma_{n+1}, \ldots, \sigma_{n+d})$, and U and V are partitioned conformally. The minimizing perturbation is given by

$$(\, F \quad E\,) = -U_2 \Sigma_2 V_2^T = -\,(\, B \quad A\,)\, V_2 V_2^T,$$

for which $\|\,(\, F \quad E\,)\,\|_F = \sum_{j=1}^{d} \sigma_{n+j}^2$ and

$$(\, B + F \quad A + E\,)\, V_2 = 0, \qquad V_2 = \begin{pmatrix} V_{12} \\ V_{22} \end{pmatrix}.$$

where $V_{12} \in \mathbf{R}^{d\times d}$. If $\hat{\sigma}_n > \sigma_{n+1}$, where $\hat{\sigma}_i$, $i = 1, \ldots, n$, are the singular values of A, it can be shown that V_{12} is nonsingular. Then the solution to the TLS problem is unique and given by

$$X = -V_{22} V_{12}^{-1} \in \mathbf{R}^{n\times d}.$$

Otherwise assume that $\sigma_k > \sigma_{k+1} = \cdots = \sigma_{n+1}$, $k \le n$, and set $V_2 = (v_{k+1}, \ldots, v_{n+d})$. Let Q be a product of Householder transformations such that

$$V_2 Q = \begin{pmatrix} \Gamma & 0 \\ Z & Y \end{pmatrix},$$

where $\Gamma \in \mathbf{R}^{d\times d}$ is lower triangular. If Γ is nonsingular, then the TLS solution of minimum norm is given by

$$X = -Z\Gamma^{-1}.$$

In many parameter estimation problems, some of the columns are known exactly. It is no restriction to assume that the error-free columns are in leading positions in A. In the multivariate version of this **mixed LS-TLS problem** one has a linear relation

$$(A_1, A_2 + E_2)X = B + F, \qquad A_1 \in \mathbf{R}^{m\times n_1},$$

where $A = (A_1,\ A_2) \in \mathbf{R}^{m\times n}$, $n = n_1 + n_2$. It is assumed that the rows of the errors $(E_2,\ F)$ are independently and identically distributed with zero mean and the same variance. The mixed LS-TLS problem can then be expressed

$$\min_{E_2, F} \|(E_2,\ F)\|_F, \qquad (A_1,\ A_2 + E_2)X = B + F. \tag{8.7.23}$$

When A_2 is empty, this reduces to solving an ordinary least squares problem. When A_1 is empty this is the standard TLS problem. Hence this mixed problem includes both extreme cases.

The solution of the mixed LS-TLS problem can be obtained by first computing a QR factorization of A and then solving a TLS problem of reduced dimension.

Algorithm 8.7.1 Mixed LS-TLS problem
Let $A = (A_1,\ A_2) \in \mathbf{R}^{m\times n}$, $n = n_1 + n_2$, $m \geq n$, and $B \in \mathbf{R}^{m\times d}$. Assume that the columns of A_1 are linearly independent. Then the following algorithm solves the mixed LS-TLS problem (8.7.23).

Step 1. Compute the QR factorization

$$(A_1,\ A_2,\ B) = Q\begin{pmatrix} R \\ 0 \end{pmatrix}, \quad R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

where Q is orthogonal, and $R_{11} \in \mathbf{R}^{n_1\times n_1}$, $R_{22} \in \mathbf{R}^{(n_2+d)\times(n_2+d)}$ are upper triangular. If $n_1 = n$, then the solution X is obtained by solving $R_{11}X = R_{12}$ (usual least squares); otherwise continue (solve a reduced TLS problem).

Step 2. Compute the SVD of R_{22}

$$R_{22} = U\Sigma V^T, \qquad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{n_2+d}),$$

where the singular values are ordered in decreasing order of magnitude.

Step 3a. Determine $k \leq n_2$ such that

$$\sigma_k > \sigma_{k+1} = \cdots = \sigma_{n_2+d} = 0,$$

and set $V_{22} = (v_{k+1}, \ldots, v_{n_2+d})$. If $n_1 > 0$ then compute V_2 by back-substitution from

$$R_{11}V_{12} = -R_{12}V_{22}, \qquad V_2 = \begin{pmatrix} V_{12} \\ V_{22} \end{pmatrix},$$

else set $V_2 = V_{22}$.

Step 3b. Perform Householder transformations such that

$$V_2Q = \begin{pmatrix} \Gamma & 0 \\ Z & Y \end{pmatrix},$$

where $\Gamma \in \mathbf{R}^{d\times d}$ is upper triangular. If Γ is nonsingular then the solution is

$$X = -Z\Gamma^{-1}.$$

Otherwise the TLS problem is nongeneric and has no solution.

Note that the QR factorization in the first step would be the first step in computing the SVD of A.

8.7.6 Iteratively Reweighted Least Squares.

In some applications it might be more adequate to solve the problem

$$\min \|Ax - b\|_p \tag{8.7.24}$$

for some l_p-norm with $p \neq 2$. For $p = 1$ the solution may not be unique, while for $1 < p < \infty$ the problem (8.7.24) is strictly convex and hence has exactly one solution. Minimization in the l_1-norm or l_∞-norm is more complicated since the function $f(x) = \|Ax - b\|_p$ is not differentiable for $p = 1, \infty$.

Example 8.7.2. *To illustrate the effect of using a different norm we consider the problem of estimating the scalar x from m observations $b \in \mathbf{R}^m$. This is equivalent to minimizing $\|Ax - b\|_p$, with $A = e = (1, 1, \ldots, 1)^T$. It is easily verified that if $b_1 \geq b_2 \geq \ldots \geq b_m$, then the solution x_p for some different values p are*

$$\begin{aligned} x_1 &= b_{\frac{m+1}{2}}, \quad (m \text{ odd}) \\ x_2 &= \frac{1}{m}(b_1 + b_2 + \ldots + b_m), \\ x_\infty &= \frac{1}{2}(b_1 + b_m). \end{aligned}$$

These estimates correspond to the median, mean, and midrange respectively. Note that the estimate x_1 is insensitive to the extreme values of b_i, while x_∞ only depends on the extreme values. The l_∞ solution has the property that the absolute error in at least n equations equals the maximum error.

The simple example above shows that the l_1 norm of the residual vector has the advantage of giving a solution that is **robust**, i.e., a small number of isolated large errors will usually not change the solution much. A similar effect is also achieved with p greater than but close to 1.

For solving the l_p norm problem when $1 < p < 3$, the **iteratively reweighted least squares** (IRLS) method (see Osborne [44, 1985]) can be used to reduce the problem to a sequence of weighted least squares problems.

We start by noting that, provided that $|r_i(x)| = |b - Ax|_i > 0$, $i = 1, \ldots, m$, the problem (8.7.24) can be restated in the form $\min_x \psi(x)$, where

$$\psi(x) = \sum_{i=1}^{m} |r_i(x)|^p = \sum_{i=1}^{m} |r_i(x)|^{p-2} r_i(x)^2. \tag{8.7.25}$$

This can be interpreted as a weighted least squares problem

$$\min_x \|D(r)^{(p-2)/2}(b - Ax)\|_2, \qquad D(r) = \operatorname{diag}(|r|), \tag{8.7.26}$$

where $\operatorname{diag}(|r|)$ denotes the diagonal matrix with ith component $|r_i|$.

The diagonal weight matrix $D(r)^{(p-2)/2}$ in (8.7.26) depends on the unknown solution x, but we can attempt to use the following iterative method.

Algorithm 8.7.2
IRLS for l_p Approximation $1 < p < 2$
Let $x^{(0)}$ be an initial approximation such that $r_i^{(0)} = (b - Ax^{(0)})_i \neq 0$, $i = 1, \ldots, n$.

$$
\begin{aligned}
&\text{for } k = 0, 1, 2, \ldots \\
&\quad r_i^{(k)} = (b - Ax^{(k)})_i; \\
&\quad D_k = \operatorname{diag}\left(\left(|r_i^{(k)}|\right)^{(p-2)/2}\right); \\
&\quad \text{solve } \delta x^{(k)} \text{ from} \\
&\qquad \min_{\delta x} \left\| D_k (r^{(k)} - A\delta x) \right\|_2; \\
&\quad x^{(k+1)} = x^{(k)} + \delta x^{(k)}; \\
&\text{end}
\end{aligned}
$$

Since $D_k b = D_k(r^{(k)} - Ax^{(k)})$, it follows that $x^{(k+1)}$ in IRLS solves $\min_x \|D_k(b - Ax)\|_2$, but the implementation above is to be preferred. It has been assumed that in the IRLS algorithm, at each iteration $r_i^{(k)} \neq 0$, $i = 1, \ldots, n$. In practice this cannot be guaranteed, and it is customary to modify the algorithm so that

$$D_k = \operatorname{diag}\left(\left(100ue + |r^{(k)}|\right)^{(p-2)/2}\right),$$

where u is the machine precision and $e^T = (1, \ldots, 1)$ is the vector of all ones. Because the weight matrix D_k is not constant, the simplest implementations of IRLS recompute, e.g., the QR factorization of $D_k A$ in each step. It should be pointed out that the iterations can be carried out entirely in the r space without the x variables. Upon convergence to a residual vector r_{opt} the corresponding solution can be found by solving the consistent linear system $Ax = b - r_{\text{opt}}$.

It can be shown that in the l_p case any fixed point of the IRLS iteration satisfies the necessary conditions for a minimum of $\psi(x)$. The IRLS method is convergent for $1 < p < 3$, and also for $p = 1$ provided that the l_1 approximation problem has a unique nondegenerate solution. However, the IRLS method can be extremely slow when p is close to unity.

Review Questions

1. Formulate the total least squares (TLS) problem. The solution of the TLS problem is related to a theorem on matrix approximation. Which?

Problems and Computer Exercises

1. Consider a TLS problem where $n = 1$ and

$$C = (A, b) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

Show that the unique minimizing ΔC gives

$$C + \Delta C = (A + E, b + r) = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}$$

so the perturbed system is not compatible, but that an arbitrary small perturbation ϵ in the (2,1) element will give a compatible system with solution $x = 2/\epsilon$.

2. Write a MATLAB program for fitting a straight line $c_1 x + c_2 y = h$ to given points $(x_i, y_i) \in \mathbf{R}^2$, $i = 1, 2, \ldots, m$. Follow the outline in Example 8.6.3. Use the Algorithm 10.4.2 to compute the SVD of R. The program should handle all exceptional cases, e.g., $c_1 = 0$ or and/or $c_2 = 0$.

3. (a) Let $A \in \mathbf{R}^{m\times n}$, $m \geq n$, $b \in \mathbf{R}^m$, and consider the **total least squares** (TLS) problem. $\min_{E,r} \|(E, r)\|_F$, where $(A+E)x = b+r$. If we have the QR factorization

$$Q^T(A,\ b) = \begin{pmatrix} S \\ 0 \end{pmatrix}, \qquad S = \begin{pmatrix} R & z \\ 0 & \rho \end{pmatrix}.$$

then the ordinary least squares solution is $x_{LS} = R^{-1}z$, $\|r\|_2 = \rho$.
Show that if a TLS solution x_{TLS} exists, then it holds

$$\begin{pmatrix} R^T & 0 \\ z^T & \rho \end{pmatrix} \begin{pmatrix} R & z \\ 0 & \rho \end{pmatrix} \begin{pmatrix} x_{TLS} \\ -1 \end{pmatrix} = \sigma_{n+1}^2 \begin{pmatrix} x_{TLS} \\ -1 \end{pmatrix},$$

where σ_{n+1} is the smallest singular value of $(A,\ b)$.

(b) Write a program using inverse iteration to compute x_{TLS}, i.e., for $k = 0, 1, 2, \ldots$, compute a sequence of vectors $x^{(k+1)}$ by

$$\begin{pmatrix} R^T & 0 \\ z^T & \rho \end{pmatrix} \begin{pmatrix} R & z \\ 0 & \rho \end{pmatrix} \begin{pmatrix} y^{(k+1)} \\ -\alpha \end{pmatrix} = \begin{pmatrix} x^{(k)} \\ -1 \end{pmatrix}, \qquad x^{(k+1)} = y^{(k+1)}/\alpha.$$

As starting vector use $x^{(0)} = x_{LS}$ on the assumption that x_{TLS} is a good approximation to x_{LS}. Will the above iteration always converge? Try to make it fail!

(c) Study the effect of scaling the right hand side in the TLS problem by making the substitution $z := \theta z$, $\rho := \theta\rho$. Plot $\|x_{TLS}(\theta) - x_{LS}\|_2$ as a function of θ and verify that when $\theta \to 0$, then $x_{TLS} \to x_{LS}$.

Hint For generating test problems it is suggested that you use the function qmult(A) from the MATLAB collection of test matrices by N. Higham to generate a matrix $C = (A,\ b) = Q_1 * D * Q_2^T$, where Q_1 and Q_2 are random real orthogonal matrices and D a given diagonal matrix. This allows you to generate problems where C has known singular values and vectors.

Notes

Several of the great mathematicians at the turn of the 19th century worked on methods for solving overdetermined linear systems. Laplace in 1799 used the principle of

minimizing the sum of absolute errors $|r_i|$. This leads to a solution x that satisfies at least n equations exactly. The method of least squares was first published as an algebraic procedure by Legendre 1805 in [39]. Gauss justified the least squares principle as a statistical procedure in [24], where he claimed to have used the method since 1795. This led to one of the most famous priority dispute in the history of mathematics. Gauss further developed the statistical aspects in 1821–1823. For an interesting accounts of the history of the invention of least squares, see Stiegler [60, 1981].

Because of its success in analyzing astronomical data the method of least squares rapidly became the method of choice when analyzing observation. Geodetic calculations was another early area of application of the least squares principle. In the last decade applications in control and signal processing has been a source of inspiration for developments in least squares calculations.

The singular value decomposition was independently developed by E. Beltrami 1873 and C. Jordan 1874; see G. W. Stewart [57, 1993] for an interesting account of the early history of the SVD. The first stable algorithm for computing the SVD the singular value was developed by Golub, Kahan and Wilkinson in the late 1960's. Several other applications of the SVD to matrix approximation can be found in Golub and Van Loan [29, Sec. 12.4].

A good introduction to generalized inverses Ben-Israel and Greville [5]. These should be used with caution since they tend to hide the computational difficulties involved with rank deficient matrices. A more complete and thorough treatment is given in the monograph by the same authors [6]. The use of generalized inverses in geodetic calculations is treated in Bjerhammar [8].

Peters and Wilkinson [49, 1970] developed methods based on Gaussian elimination from a uniform standpoint and the excellent survey by Noble [43, 1976]. Sautter [53, 1978] gives a detailed analysis of stability and rounding errors of the LU algorithm for computing pseudo-inverse solutions.

The different computational variants of Gram–Schmidt have an interesting history. The "modified" Gram–Schmidt (MGS) algorithm was in fact already derived by Laplace in 1816 as an elimination method using weighted row sums. Laplace did not interpret his algorithm in terms of orthogonalization, nor did he use it for computing least squares solutions! Bienaymé in 1853 gave a similar derivation of a slightly more general algorithm; see Björck [10, 1994]. What is now called the "classical" Gram–Schmidt (CGS) algorithm first appeared explicitly in papers by Gram 1883 and Schmidt 1908. Schmidt treats the solution of linear systems with infinitely many unknowns and uses the orthogonalization as a theoretical tool rather than a computational procedure.

In the 1950's algorithms based on Gram–Schmidt orthogonalization were frequently used, although their numerical properties were not well understood at the time. Björck [9] analyzed the modified Gram–Schmidt algorithm and showed its stability for solving linear least squares problems.

The systematic use of orthogonal transformations to reduce matrices to simpler form was initiated by Givens [25, 1958] and Householder [36, 1958]. The application of these transformations to linear least squares is due to Golub [26, 1965], where it was shown how to compute a QR factorization of A using Householder

transformations.

How to find the optimal backward error for the linear least squares problem was an open problem for many years, until it was elegantly answered by Karlsson et al. [64]; see also [37]. Gu [30] gives several approximations to that are optimal up to a factor less than 2. Optimal backward perturbation bounds for underdetermined systems are derived in [61]. The extension of backward error bounds to the case of constrained least squares problems is discussed by Cox and Higham [19].

The QR algorithm for banded rectangular matrices was first given by Reid [52]. Rank-revealing QR (RRQR) decompositions have been studied by a number of authors. A good survey can be found in Hansen [32]. The URV and ULV decompositions were introduced by G. W. Stewart [56, 58].

The systematic use of GQR as a basic conceptual and computational tool are explored by [45]. These generalized decompositions and their applications are discussed in [1]. Algorithms for computing the bidiagonal decomposition are due to Golub and Kahan [27, 1965]. The partial least squares (PLS) method, which has become a standard tool in chemometrics, goes back to Wold et al. [66].

The term "total least squares problem", which was coined by Golub and Van Loan [28], renewed the interest in the "errors in variable model". A thorough and rigorous treatment of the TLS problem is found in Van Huffel and Vandewalle [63]. The important role of the core problem for weighted TLS problems was discovered by Paige and Strakoš [47].

Modern numerical methods for solving least squares problems are surveyed in the two comprehensive monographs [38] and [11]. The latter contains a bibliography of 860 references, indicating the considerable research interest in these problems. Hansen [32] gives an excellent survey of numerical methods for the treatment of numerically rank deficient linear systems arising, for example, from discrete ill-posed problems.

Bibliography

[1] Edward Anderssen, Zhaojun Bai, and J. J. Dongarra. Generalized QR factorization and its applications. *Linear Algebra Appl.*, 162–164:243–271, 1992.

[2] Mario Arioli, James W. Demmel, and Iain S. Duff. Solving sparse linear systems with sparse backward error. *SIAM J. Matrix Anal. Appl.*, 10:165–190, 1989.

[3] Jesse L. Barlow. More accurate bidiagonal reduction algorithm for computing the singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 23:3:761–798, 2002.

[4] Jesse L. Barlow, Nela Bosner, and Zlatko Dramăk. A new stable bidiagonal reduction algorithm. *Linear Algebra Appl.*, 397:35–84, 2005.

[5] Adi Ben-Israel and T. N. E. Greville. Some topics in generalized inverses of matrices. In M. Z. Nashed, editor, *Generalized Inverses and Applications*, pages 125–147. Academic Press, Inc., New York, 1976.

[6] Adi Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications*. Springer, Berlin–Heidelberg–New York, 2003.

[7] Commandant Benoit. Sur la méthode de résolution des équations normales, etc. (procédés du commandant Cholesky). *Bull. Géodesique*, 2:67–77, 1924.

[8] Arne Bjerhammar. *Theory of Errors and Generalized Inverse Matrices*. Elsevier Scientific Publishing Co., Amsterdam, 1973.

[9] Å. Björck. Solving linear least squares problems by Gram–Schmidt orthogonalization. *BIT*, 7:1–21, 1967.

[10] Åke Björck. Numerics of Gram–Schmidt orthogonalization. *Linear Algebra Appl.*, 197–198:297–316, 1994.

[11] Åke Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, PA, 1996.

[12] Åke Björck and Christopher C. Paige. Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm. *SIAM J. Matrix Anal. Appl.*, 13:176–190, 1992.

[13] Adam Bojanczyk, Nicholas J. Higham, and H. Patel. Solving the indefinite least squares problem by hyperbolic qr factorization. *SIAM J. Matrix Anal. Appl.*, 24:4:914–931, 2003.

[14] Tony F. Chan. Rank revealing QR factorizations. *Linear Algebra Appl.*, 88/89:67–82, 1987.

[15] Tony F. Chan and Per Christian Hansen. Low-rank revealing QR factorizations. *Numer. Linear Algebra Appl.*, 1:33–44, 1994.

[16] S. Chandrasekaran, Ming Gu, and A. H. Sayed. A stable and efficient algorithm for the indefinite linear least-squares problem. *SIAM J. Matrix Anal. Appl.*, 20:2:354–362, 1998.

[17] R. E. Cline and Robert J. Plemmons. ℓ_2-solutions to underdetermined linear systems. *SIAM Review*, 18:92–106, 1976.

[18] Anthony J. Cox and Nicholas J. Higham. Stability of Householder qr factorization for weighted least squares problems. In D. F. Griffiths, D. J. Higham, and G. A. Watson, editors, *Numerical Analysis 1997: Proceedings of the 17th Dundee Biennial Conference*, Pitman Research Notes in mathematics, vol. 380, pages 57–73. Longman Scientific and Technical, Harlow, Essex, UK, 1998.

[19] Anthony J. Cox and Nicholas J. Higham. Backward error bounds for constrained least squares problems. *BIT*, 39:2:210–227, 1999.

[20] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrica*, 1:211–218, 1936.

[21] Ky Fan and Alan J. Hoffman. Some metric inequalities in the space of matrices. *Proc. Amer. Math. Soc.*, 6:111–116, 1955.

[22] R. D. Fierro, P. C. Hansen, and P. S. K. Hansen. UTV tools: Matlab templates for rank-revealing UTV decompositions. *Numerical Algorithms*, to appear, 2003.

[23] C. F. Gauss. *The Theory of the Combination of Observations Least Subject to Errors. Pars Prior.* SIAM, Philadelphia, PA, G. W. Stewart, Translation 1995, 1821.

[24] Carl Friedrich Gauss. *Theory of the Motion of of the Heavenly Bodies Moving about the Sun in Conic Sections.* Dover, New York, (1963), C. H. Davis, Translation, 1809.

[25] Wallace G. Givens. Computation of plane unitary rotations transforming a general matrix to triangular form. *SIAM J. Appl. Math.*, 6:26–50, 1958.

[26] Gene H. Golub. Numerical methods for solving least squares problems. *Numer. Math.*, 7:206–216, 1965.

[27] Gene H. Golub and W. Kahan. Calculating the singular values and pseudoinverse of a matrix. *SIAM J. Numer. Anal. Ser. B*, 2:205–224, 1965.

[28] Gene H. Golub and Charles F. Van Loan. An analysis of the total least squares problem. *SIAM J. Numer. Anal.*, 17:883–893, 1980.

[29] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[30] Ming Gu. Backward perturbation bounds for linear least squares problems. *SIAM J. Matrix. Anal. Appl.*, 20:2:363–372, 1998.

[31] M. Gulliksson and P.-Å. Wedin. Perturbation theory for generalized and constrained linear least squares. *Numer. Linear Algebra Appl.*, 7:181–196, 2000.

[32] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion.* SIAM, Philadelphia, 1998.

[33] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, second edition, 2002.

[34] Nicholas J. Higham. J-Orthogonal matrices: Properties and generation. *SIAM Review*, 45:3:504–519, 2003.

[35] Y. T. Hong and C. T. Pan. Rank-revealing QR decompositions and the singular value decomposition. *Math. Comp.*, 58:213–232, 1992.

[36] A. S. Householder. Unitary triangularization of a nonsymmetric matrix. *J. Assoc. Comput. Mach.*, 5:339–342, 1958.

[37] Rune Karlsson and Bertil Waldén. Estimation of optimal backward perturbation bounds for the linear least squares problem. *BIT*, 37:4:862–869, 1997.

[38] Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974. Reprinted by SIAM, Philadelphia, PA, 1995.

[39] Adrien-Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*. Courcier, Paris, 1805.

[40] A. A. Markov. *Wahrscheinlichheitsrechnung*. Liebmann, Leipzig, second edition, 1912.

[41] Roy Mathias and G. W. Stewart. A block QR algorithm and the singular value decompositions. *Linear Algebra Appl.*, 182:91–100, 1993.

[42] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math. Oxford*, 11:50–59, 1960.

[43] Ben Noble. Methods for computing the moore–penrose generalized inverse and related matters. In M. Z. Nashed, editor, *Generalized Inverses and Applications*, pages 245–302. Academic Press, Inc., New York, 1976.

[44] M. R. Osborne. *Finite Algorithms in Optimization and Data Analysis*. John Wiley, New York, 1985.

[45] Christopher C. Paige. Some aspects of generalized QR factorizations. In M. G. Cox and S. J. Hammarling, editors, *Reliable Numerical Computation*, pages 71–91. Clarendon Press, Oxford, UK, 1990.

[46] Christopher C. Paige and Z. Strakoš. Unifying least squares, total least squares and data least squares. In S. Van Huffel and P. Lemmerling, editors, *Total Least Squares and Errors-in-Variables Modeling*, pages 25–34. Kluwer Academic Publishers, Dordrecht, 2002.

[47] Christopher C. Paige and Zdeněk Strakoš. Scaled total least squares fundamentals. *Numer. Math.*, 91:1:117–146, 2002a.

[48] Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Classics in Applied Mathematics 20. SIAM, Philadelphia, PA, 1998.

[49] G. Peters and James H. Wilkinson. The least squares problem and pseudo-inverses. *Comput. J.*, 13:309–316, 1970.

[50] Robert J. Plemmons. Linear least squares by elimination and MGS. *J. Assoc. Comput. Mach.*, 21:581–585, 1974.

[51] M. J. D. Powell and J. K. Reid. On applying Householder's method to linear least squares problems. In A. J. M. Morell, editor, *Proceedings of the IFIP Congress 68*, pages 122–126. North-Holland, Amsterdam, 1969.

[52] J. K. Reid. A note on the least squares solution of a band system of linear equations by Householder reductions. *Comput J.*, 10:188–189, 1967.

[53] Werner Sautter. Fehleranalyse für die Gauss-Elimination zur Berechnung der Lösung minimaler Länge. *Numer. Math.*, 30:165–184, 1978.

[54] W. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrica.*, 31:1–10, 1966.

[55] H. R. Schwartz. Tridiagonalizing of a symmetric band matrix. *Numer. Math.*, 12:231–241, 1968. See also Wilkinson and Reinsch, 1971, 273–283.

[56] G. W. Stewart. An updating algorithm for subspace tracking. *IEEE Trans. Signal Process.*, 40:1535–1541, 1992.

[57] G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35:4:551–556, 1993.

[58] G. W. Stewart. Updating a rank-revealing ULV decomposition. *SIAM J. Matrix Anal. Appl.*, 14:494–499, 1993.

[59] G. W. Stewart. The QLP approximation to the singular value decomposition. *SIAM J. Sci. Comput.*, 20:4:1336–1348, 1999.

[60] S. M. Stiegler. Gauss and the invention of least squares. *Ann. Statist.*, 9:465–474, 1981.

[61] Ji-guang Sun and Zheng Sun. Optimal backward perturbation bounds for underdetermined systems. *SIAM J. Matrix Anal. Appl.*, 18:393–402, 1997.

[62] Lloyd N. Trefethen and III David Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[63] Sabine Van Huffel and Joos Vandewalle. *The Total Least Squares Problem; Computational Aspects and Analysis*. SIAM, Philadelphia, PA, 1991.

[64] Bertil Waldén, Rune Karlsson, and Ji guang. Sun. Optimal backward perturbation bounds for the linear least squares problem. *Numer. Linear Algebra Appl.*, 2:271–286, 1995.

[65] Per-Åke Wedin. Perturbation theory for pseudo-inverses. *BIT*, 9:217–232, 1973.

[66] Svante Wold, Axel Ruhe, Herman Wold, and W. J. Dunn. The collinearity problem in linear regression, the partial least squares (pls) approach to generalized inverses. *SIAM J. Sci. Statist. Comput.*, 5:735–743, 1984.

[67] M. Zelen. Linear estimation and related topics. In J. Todd, editor, *Survey of Numerical Analysis*, pages 558–584. McGraw-Hill, New York, 1962.

Index