Chapter 3
Linear Systems and Least Squares

In this chapter we briefly review some facts about the solution of linear systems of equations,

$$Ax = b, \tag{3.1}$$

where $A \in \mathbb{R}^{n\times n}$ is square and nonsingular. The linear system (3.1) can be solved using *Gaussian elimination* with *partial pivoting*, which is equivalent to factorizing the matrix as a product of triangular matrices.

We will also consider *overdetermined linear systems*, where the matrix $A \in \mathbb{R}^{m\times n}$ is *rectangular* with $m > n$, and their solution using the least squares method. As we are giving the results only as background, we mostly state them without proofs. For thorough presentations of the theory of matrix decompositions for solving linear systems of equations, see, e.g., [42, 92].

Before discussing matrix decompositions, we state the basic result concerning conditions for the existence of a unique solution of (3.1).

Proposition 3.1. *Let $A \in \mathbb{R}^{n\times n}$ and assume that A is nonsingular. Then for any right-hand-side b, the linear system $Ax = b$ has a unique solution.*

Proof. The result is an immediate consequence of the fact that the column vectors of a nonsingular matrix are linearly independent. ∎

3.1 LU Decomposition

Gaussian elimination can be conveniently described using *Gauss transformations*, and these transformations are the key elements in the equivalence between Gaussian elimination and LU decomposition. More details on Gauss transformations can be found in any textbook in numerical linear algebra; see, e.g., [42, p. 94]. In Gaussian elimination with partial pivoting, the reordering of the rows is accomplished by

permutation matrices, which are identity matrices with the rows reordered; see, e.g., [42, Section 3.4.1].

Consider an $n \times n$ matrix A. In the first step of Gaussian elimination with partial pivoting, we reorder the rows of the matrix so that the element of largest magnitude in the first column is moved to the $(1,1)$ position. This is equivalent to multiplying A from the left by a permutation matrix P_1. The elimination, i.e., the zeroing of the elements in the first column below the diagonal, is then performed by multiplying

$$A^{(1)} := L_1^{-1} P_1 A, \tag{3.2}$$

where L_1 is a Gauss transformation

$$L_1 = \begin{pmatrix} 1 & 0 \\ m_1 & I \end{pmatrix}, \qquad m_1 = \begin{pmatrix} m_{21} \\ m_{31} \\ \vdots \\ m_{n1} \end{pmatrix}.$$

The result of the first step of Gaussian elimination with partial pivoting is

$$A^{(1)} = \begin{pmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & & & \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{pmatrix}.$$

The Gaussian elimination algorithm then proceeds by zeroing the elements of the second column below the main diagonal (after moving the largest element to the diagonal position), and so on.

From (3.2) we see that the first step of Gaussian elimination with partial pivoting can be expressed as a matrix factorization. This is also true of the complete procedure.

Theorem 3.2 (LU decomposition). *Any nonsingular $n \times n$ matrix A can be decomposed into*

$$PA = LU,$$

where P is a permutation matrix, L is a lower triangular matrix with ones on the main diagonal, and U is an upper triangular matrix.

Proof (sketch). The theorem can be proved by induction. From (3.2) we have

$$P_1 A = L_1 A^{(1)}.$$

Define the $(n-1) \times (n-1)$ matrix

$$B = \begin{pmatrix} a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & & \\ a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{pmatrix}.$$

By an induction assumption, B can be decomposed into

$$P_B B = L_B U_B,$$

and we then see that $PA = LU$, where

$$U = \begin{pmatrix} a'_{11} & a_2^T \\ 0 & U_B \end{pmatrix}, \qquad L = \begin{pmatrix} 1 & 0 \\ P_B m_1 & L_B \end{pmatrix}, \qquad P = \begin{pmatrix} 1 & 0 \\ 0 & P_B \end{pmatrix} P_1,$$

and $a_2^T = (a'_{12}\, a'_{13}\, \ldots\, a'_{1n})$. ∎

It is easy to show that the amount of work for computing the LU decomposition is $2n^3/3$ flops, approximately. In the kth step of Gaussian elimination, one operates on an $(n-k+1) \times (n-k+1)$ submatrix, and for each element in that submatrix one multiplication and one addition are performed. Thus the total number of flops is

$$2\sum_{k=1}^{n-1}(n-k+1)^2 \approx \frac{2n^3}{3},$$

approximately.

3.2 Symmetric, Positive Definite Matrices

The LU decomposition of a symmetric, positive definite matrix A can always be computed without pivoting. In addition, it is possible to take advantage of symmetry so that the decomposition becomes symmetric, too, and requires half as much work as in the general case.

Theorem 3.3 (LDL^T decomposition). *Any symmetric, positive definite matrix A has a decomposition*

$$A = LDL^T,$$

where L is lower triangular with ones on the main diagonal and D is a diagonal matrix with positive diagonal elements.

Example 3.4. The positive definite matrix

$$A = \begin{pmatrix} 8 & 4 & 2 \\ 4 & 6 & 0 \\ 2 & 0 & 3 \end{pmatrix}$$

has the LU decomposition

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.25 & -0.25 & 1 \end{pmatrix} \begin{pmatrix} 8 & 4 & 2 \\ 0 & 4 & -1 \\ 0 & 0 & 2.25 \end{pmatrix}$$

and the LDL^T decomposition

$$A = LDL^T, \quad D = \begin{pmatrix} 8 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2.25 \end{pmatrix}. \quad \blacksquare$$

The diagonal elements in D are positive, and therefore we can put

$$D^{1/2} = \begin{pmatrix} \sqrt{d_1} & & & \\ & \sqrt{d_2} & & \\ & & \ddots & \\ & & & \sqrt{d_n} \end{pmatrix},$$

and then we get

$$A = LDL^T = (LD^{1/2})(D^{1/2}L^T) = U^T U,$$

where U is an upper triangular matrix. This variant of the LDL^T decomposition is called the *Cholesky decomposition*.

Since A is symmetric, it is only necessary to store the main diagonal and the elements above it, $n(n+1)/2$ matrix elements in all. Exactly the same amount of storage is needed for the LDL^T and the Cholesky decompositions. It is also seen that since only half as many elements as in the ordinary LU decomposition need to be computed, the amount of work is also halved—approximately $n^3/3$ flops. When the LDL^T decomposition is computed, it is not necessary to first compute the LU decomposition, but the elements in L and D can be computed directly.

3.3 Perturbation Theory and Condition Number

The *condition number* of a nonsingular matrix A is defined as

$$\kappa(A) = \|A\| \, \|A^{-1}\|,$$

where $\|\cdot\|$ denotes any operator norm. If we use a particular matrix norm, e.g., the 2-norm, then we write

$$\kappa_2(A) = \|A\|_2 \, \|A^{-1}\|_2. \tag{3.3}$$

The condition number is used to quantify how much the solution of a linear system $Ax = b$ can change, when the matrix and the right-hand side are perturbed by a small amount.

Theorem 3.5. *Assume that A is nonsingular and that*

$$\|\delta A\| \, \|A^{-1}\| = r < 1.$$

Then the matrix $A + \delta A$ is nonsingular, and

$$\|(A + \delta A)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - r}.$$

The solution of the perturbed system

$$(A + \delta A)y = b + \delta b$$

satisfies

$$\frac{\|y - x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - r}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right).$$

For a proof, see, for instance, [42, Theorem 2.7.2] or [50, Theorem 7.2]

A matrix with a large condition number is said to be *ill-conditioned*. Theorem 3.5 shows that a linear system with an ill-conditioned matrix is sensitive to perturbations in the data (i.e., the matrix and the right-hand side).

3.4 Rounding Errors in Gaussian Elimination

From Section 1.5.2, on rounding errors in floating point arithmetic, we know that any real number (representable in the floating point system) is represented with a relative error not exceeding the unit round-off μ. This fact can also be stated

$$fl[x] = x(1 + \epsilon), \qquad |\epsilon| \leq \mu.$$

When representing the elements of a matrix A and a vector b in the floating point system, there arise errors:

$$fl[a_{ij}] = a_{ij}(1 + \epsilon_{ij}), \qquad |\epsilon_{ij}| \leq \mu,$$

and analogously for b. Therefore, we can write

$$fl[A] = A + \delta A, \qquad fl[b] = b + \delta b,$$

where

$$\|\delta A\|_\infty \leq \mu\|A\|_\infty, \qquad \|\delta b\|_\infty \leq \mu\|b\|_\infty.$$

If, for the moment, we assume that no further rounding errors arise during the solution of the system $Ax = b$, we see that $\widehat{x}$ satisfies

$$(A + \delta A)\widehat{x} = b + \delta b.$$

This is an example of *backward error analysis*: the computed solution $\widehat{x}$ is the *exact solution of a perturbed problem*.

Using perturbation theory, we can estimate the error in $\widehat{x}$. From Theorem 3.5 we get

$$\frac{\|\widehat{x} - x\|_\infty}{\|x\|_\infty} \leq \frac{\kappa_\infty(A)}{1-r}\, 2\mu$$

(provided that $r = \mu\kappa_\infty(A) < 1$).

We can also analyze how rounding errors in Gaussian elimination affect the result. The following theorem holds. (For detailed error analyses of Gaussian elimination, see [50, Chapter 9] or [42, Chapters 3.3, 3.4].)

Theorem 3.6. *Assume that we use a floating point system with unit round-off μ. Let $\widehat{L}$ and $\widehat{R}$ be the triangular factors obtained from Gaussian elimination with partial pivoting, applied to the matrix A. Further, assume that $\widehat{x}$ is computed using forward and back substitution:*

$$\widehat{L}\widehat{y} = Pb, \quad \widehat{R}\,\widehat{x} = \widehat{y}.$$

Then $\widehat{x}$ is the exact solution of a system

$$(A + \delta A)\widehat{x} = b,$$

where

$$\|\delta A\|_\infty \leq k(n) g_n \mu \|A\|_\infty, \qquad g_n = \frac{\max\limits_{i,j,k} |\widehat{a}_{ij}^{(k)}|}{\max\limits_{i,j} |a_{ij}|},$$

$k(n)$ is a third-degree polynomial in n, and $\widehat{a}_{ij}^{(k)}$ are the elements computed in step $k-1$ of the elimination procedure.

We observe that g_n depends on the growth of the matrix elements during the Gaussian elimination and not explicitly on the magnitude of the multipliers. g_n can be computed, and in this way an *a posteriori* estimate of the rounding errors can be obtained.

A priori (in advance), one can show that $g_n \leq 2^{n-1}$, and matrices can be constructed where in fact the element growth is that serious (note that $g_{31} = 2^{30} \approx 10^9$). In practice, however, g_n is seldom larger than 8 in Gaussian elimination with partial pivoting.

It is important to note that there are classes of matrices for which there is no element growth during Gaussian elimination, i.e., $g_n = 1$, even if no pivoting is done. This is true, e.g., if A is symmetric and positive definite.

In almost all cases, the estimate in the theorem is much too pessimistic with regard to the third-degree polynomial $k(n)$. In order to have equality, all rounding errors must be maximally large, and their accumulated effect must be maximally unfavorable.

We want to emphasize that the main objective of this type of a priori error analysis is not to give error estimates for the solution of linear systems but rather to expose potential instabilities of algorithms and provide a basis for comparing different algorithms. Thus, Theorem 3.6 demonstrates the main weakness of Gauss transformations as compared to the orthogonal transformations that we will introduce in Chapter 4: they can cause a large growth of the matrix elements, which, in turn, induces rounding errors.

3.5 Banded Matrices

In many situations, e.g., boundary value problems for ordinary and partial differential equations, matrices arise where a large proportion of the elements are equal to zero. If the nonzero elements are concentrated around the main diagonal, then the matrix is called a band matrix. More precisely, a matrix A is said to be a *band matrix* if there are natural numbers p and q such that

$$a_{ij} = 0 \text{ if } j - i > p \text{ or } i - j > q.$$

Example 3.7. Let $q = 2$, $p = 1$. Let A be a band matrix of dimension 6:

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & 0 \\ 0 & 0 & a_{53} & a_{54} & a_{55} & a_{56} \\ 0 & 0 & 0 & a_{64} & a_{65} & a_{66} \end{pmatrix}. \quad \blacksquare$$

$w = q + p + 1$ is called the *bandwidth* of the matrix. From the example, we see that w is the maximal number of nonzero elements in any row of A.

When storing a band matrix, we do not store the elements outside the band. Likewise, when linear systems of equations are solved, one can take advantage of the band structure to reduce the number of operations.

We first consider the case $p = q = 1$. Such a band matrix is called *tridiagonal*. Let

$$A = \begin{pmatrix} \alpha_1 & \beta_1 & & & & \\ \gamma_2 & \alpha_2 & \beta_2 & & & \\ & \gamma_3 & \alpha_3 & \beta_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \gamma_{n-1} & \alpha_{n-1} & \beta_{n-1} \\ & & & & \gamma_n & \alpha_n \end{pmatrix}.$$

The matrix can be stored in three vectors. In the solution of a tridiagonal system $Ax = b$, it is easy to utilize the structure; we first assume that A is diagonally dominant, so that no pivoting is needed.

```
% LU Decomposition of a Tridiagonal Matrix.
for k=1:n-1
  gamma(k+1)=gamma(k+1)/alpha(k);
  alpha(k+1)=alpha(k+1)*beta(k);
end
% Forward Substitution for the Solution of Ly = b.
y(1)=b(1);
for k=2:n
  y(k)=b(k)-gamma(k)*y(k-1);
end
% Back Substitution for the Solution of Ux = y.
x(n)=y(n)/alpha(n);
for k=n-1:-1:1
  x(k)=(y(k)-beta(k)*x(k+1))/alpha(k);
end
```

The number of operations (multiplications and additions) is approximately $3n$, and the number of divisions is $2n$.

In Gaussian elimination *with partial pivoting*, the band width of the upper triangular matrix increases. If A has band width $w = q + p + 1$ (q diagonals under the main diagonal and p over), then, with partial pivoting, the factor U will have band width $w_U = p + q + 1$. It is easy to see that no new nonzero elements will be created in L.

The factors L and U in the LU decomposition of a band matrix A are band matrices.

Example 3.8. Let

$$A = \begin{pmatrix} 4 & 2 & & & \\ 2 & 5 & 2 & & \\ & 2 & 5 & 2 & \\ & & 2 & 5 & 2 \\ & & & 2 & 5 \end{pmatrix}.$$

A has the Cholesky decomposition $A = U^T U$, where

$$U = \begin{pmatrix} 2 & 1 & & & \\ & 2 & 1 & & \\ & & 2 & 1 & \\ & & & 2 & 1 \\ & & & & 2 \end{pmatrix}.$$

The inverse is

$$A^{-1} = \frac{1}{2^{10}} \begin{pmatrix} 341 & -170 & 84 & -40 & 16 \\ -170 & 340 & -168 & 80 & -32 \\ 84 & -168 & 336 & -160 & 64 \\ -40 & 80 & -160 & 320 & -128 \\ 16 & -32 & 64 & -128 & 256 \end{pmatrix},$$

which is dense. ∎

It turns out that the inverse of a band matrix is usually a dense matrix. Therefore, in most cases the inverse of a band matrix should not be computed explicitly.

3.6 The Least Squares Problem

In this section we will introduce the least squares method and the solution of the linear least squares problem using the normal equations. Other methods for solving the least squares problem, based on orthogonal transformations, will be presented in Chapters 5 and 6. We will also give a perturbation result for the least squares problem in Section 6.6. For an extensive treatment of modern numerical methods for linear least squares problem, see [14].

Example 3.9. Assume that we want to determine the elasticity properties of a spring by attaching different weights to it and measuring its length. From Hooke's law we know that the length l depends on the force F according to

$$e + \kappa F = l,$$

where e and κ are constants to be determined.[5] Assume that we have performed an experiment and obtained the following data:

F	1	2	3	4	5
l	7.97	10.2	14.2	16.0	21.2

The data are illustrated in Figure 3.1. As the measurements are subject to error we want to use all the data in order to minimize the influence of the errors. Thus we are lead to a system with more data than unknowns, an *overdetermined system*,

$$\begin{aligned} e + \kappa 1 &= 7.97, \\ e + \kappa 2 &= 10.2, \\ e + \kappa 3 &= 14.2, \\ e + \kappa 4 &= 16.0, \\ e + \kappa 5 &= 21.2, \end{aligned}$$

or, in matrix form,

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} e \\ \kappa \end{pmatrix} = \begin{pmatrix} 7.97 \\ 10.2 \\ 14.2 \\ 16.0 \\ 21.2 \end{pmatrix}.$$

We will determine an approximation of the elasticity constant of the spring using the least squares method. ∎

[5] In Hooke's law the *spring constant* is $1/\kappa$.

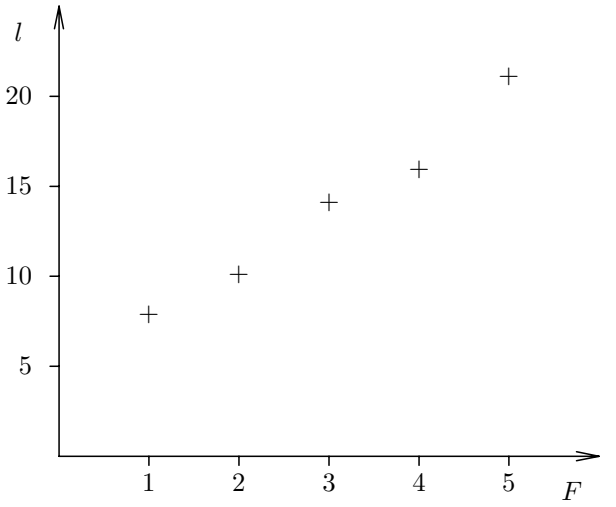

Figure 3.1. *Measured data in spring experiment.*

Let $A \in \mathbb{R}^{m\times n}$, $m > n$. The system

$$Ax = b$$

is called *overdetermined*: it has more equations than unknowns. In general such a system has no solution. This can be seen geometrically by letting $m = 3$ and $n = 2$, i.e., we consider two vectors $a_{\cdot 1}$ and $a_{\cdot 2}$ in $\mathbb{R}^3$. We want to find a linear combination of the vectors such that

$$x_1 a_{\cdot 1} + x_2 a_{\cdot 2} = b.$$

In Figure 3.2 we see that usually such a problem has no solution. The two vectors span a plane, and if the right-hand side b is not in the plane, then there is no linear combination of $a_{\cdot 1}$ and $a_{\cdot 2}$ such that $x_1 a_{\cdot 1} + x_2 a_{\cdot 2} = b$.

In this situation one obvious alternative to "solving the linear system" is to make the vector $r = b - x_1 a_{\cdot 1} - x_2 a_{\cdot 2} = b - Ax$ as small as possible. $b - Ax$ is called the *residual vector* and is illustrated in Figure 3.2.

The solution of the problem depends on how we measure the length of the residual vector. In the *least squares method* we use the standard Euclidean distance. Thus we want to find a vector $x \in \mathbb{R}^n$ that solves the minimization problem

$$\min_x \| b - Ax \|_2. \tag{3.4}$$

As the unknown x occurs linearly in (3.4), this is also referred to as the *linear least squares problem*.

In the example we know immediately from our knowledge of distances in $\mathbb{R}^3$ that the distance between the tip of the vector b and the plane is minimized if we choose the linear combination of vectors in the plane in such a way that the residual

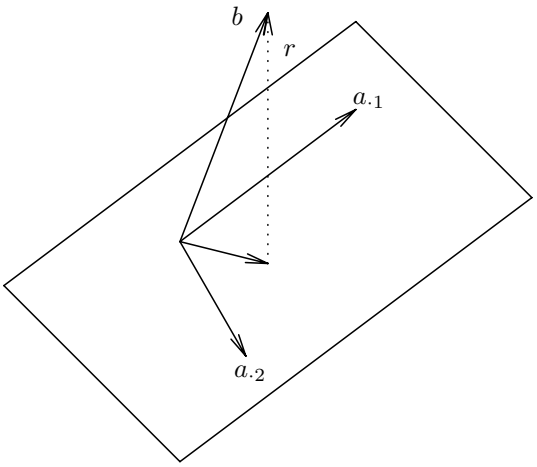

Figure 3.2. *The least squares problem, $m = 3$ and $n = 2$. The residual vector $b - Ax$ is dotted.*

vector is orthogonal to the plane. Since the columns of the matrix A span the plane, we see that we get the solution by *making r orthogonal to the columns of A*. This geometric intuition is valid also in the general case:

$$r^T a_{\cdot j} = 0, \qquad j = 1, 2, \ldots, n.$$

(See the definition of orthogonality in Section 2.3.) Equivalently, we can write

$$r^T \begin{pmatrix} a_{\cdot 1} & a_{\cdot 2} & \cdots & a_{\cdot n} \end{pmatrix} = r^T A = 0.$$

Then, using $r = b - Ax$, we get the *normal equations* (the name is now obvious)

$$A^T Ax = A^T b$$

for determining the coefficients in x.

Theorem 3.10. *If the column vectors of A are linearly independent, then the normal equations*

$$A^T Ax = A^T b$$

are nonsingular and have a unique solution.

Proof. We first show that $A^T A$ is positive definite. Let x be an arbitrary nonzero vector. Then, from the definition of linear independence, we have $Ax \neq 0$. With $y = Ax$, we then have

$$x^T A^T Ax = y^T y = \sum_{i=1}^{n} y_i^2 > 0,$$

which is equivalent to A^TA being positive definite. Therefore, A^TA is nonsingular, and the normal equations have a unique solution, which we denote $\widehat{x}$.

Then, we show that $\widehat{x}$ is the solution of the least squares problem, i.e., $\|\widehat{r}\|_2 \leq \|r\|_2$ for all $r = b - Ax$. We can write

$$r = b - A\widehat{x} + A(\widehat{x} - x) = \widehat{r} + A(\widehat{x} - x)$$

and

$$\begin{aligned}\|r\|_2^2 = r^T r &= (\widehat{r} + A(\widehat{x} - x))^T(\widehat{r} + A(\widehat{x} - x)) \\ &= \widehat{r}^T\widehat{r} + \widehat{r}^T A(\widehat{x} - x) + (\widehat{x} - x)^T A^T\widehat{r} + (\widehat{x} - x)^T A^T A(\widehat{x} - x).\end{aligned}$$

Since $A^T\widehat{r} = 0$, the two terms in the middle are equal to zero, and we get

$$\|r\|_2^2 = \widehat{r}^T\widehat{r} + (\widehat{x} - x)^T A^T A(\widehat{x} - x) = \|\widehat{r}\|_2^2 + \|A(\widehat{x} - x)\|_2^2 \geq \|\widehat{r}\|_2^2,$$

which was to be proved. ∎

Example 3.11. We can now solve the example given at the beginning of the chapter. We have

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 41 & 5 \end{pmatrix}, \qquad b = \begin{pmatrix} 7.97 \\ 10.2 \\ 14.2 \\ 16.021.2 \end{pmatrix}.$$

Using MATLAB we then get

```
>> C=A'*A          % Normal equations

C =  5    15
    15    55

>> x=C\(A'*b)

x = 4.2360
    3.2260
```

Solving the linear least squares problems using the normal equations has two significant drawbacks:

1. Forming A^TA can lead to loss of information.
2. The condition number A^TA is the square of that of A:

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

We illustrate these points in a couple of examples.

Example 3.12. Let ϵ be small, and define the matrix

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

It follows that

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

If ϵ is so small that the floating point representation of $1+\epsilon^2$ satisfies $fl[1+\epsilon^2] = 1$, then in floating point arithmetic the normal equations become singular. Thus vital information that is present in A is lost in forming $A^T A$. ∎

The condition number of a rectangular matrix A is defined using the singular value decomposition of A. We will state a result on the conditioning of the least squares problem in Section 6.6.

Example 3.13. We compute the condition number of the matrix in Example 3.9 using MATLAB:

```
A =  1     1
     1     2
     1     3
     1     4
     1     5

cond(A) = 8.3657

cond(A'*A) = 69.9857
```

Then we assume that we have a linear model

$$l(x) = c_0 + c_1 x$$

with data vector $x = (101\,102\,103\,104\,105)^T$. This gives a data matrix with large condition number:

```
A =  1   101
     1   102
     1   103
     1   104
     1   105

cond(A) = 7.5038e+03

cond(A'*A) = 5.6307e+07
```

If instead we use the model

$$l(x) = b_0 + b_1(x - 103),$$

the corresponding normal equations become diagonal and much better conditioned (demonstrate this). ∎

It occurs quite often that one has a sequence of least squares problems with the same matrix,

$$\min_{x_i} \|Ax_i - b_i\|_2, \qquad i = 1, 2, \ldots, p,$$

with solutions

$$x_i = (A^T A)^{-1} A^T b_i, \qquad i = 1, 2, \ldots, p.$$

Defining $X = \begin{pmatrix} x_1 & x_2 & \ldots & x_p \end{pmatrix}$ and $X = \begin{pmatrix} b_1 & b_2 & \ldots & b_p \end{pmatrix}$ we can write this in matrix form

$$\min_X \|AX - B\|_F \tag{3.5}$$

with the solution

$$X = (A^T A)^{-1} A^T B.$$

This follows from the identity

$$\|AX - B\|_F^2 = \sum_{i=1}^{p} \|Ax_i - b_i\|_2^2$$

and the fact that the p subproblems in (3.5) are independent.