Contents

1 Vector and matrix norms 2
  1.1 Vector norms ............................................................... 3
  1.2 Matrix norms ............................................................... 4
  1.3 Some important matrix norms ......................................... 5

2 Sensitivity of linear systems 6
  2.1 Bound with exact $A$ ................................................... 6
  2.2 Properties of the condition number .................................. 8
  2.3 Examples ...................................................................... 9
  2.4 Residuals vs. error ....................................................... 10

3 An important special case 11
  3.1 Positive definite matrices .............................................. 11
  3.2 Diagonal dominance ..................................................... 12
  3.3 Gerschgorin’s theorem (optional) .................................... 13

4 Iterative methods 13
  4.1 Jacobi method ............................................................. 15
     4.1.1 Implementation .................................................... 15
  4.2 Gauss-Seidel ............................................................... 16
  4.3 An example .................................................................. 17
  4.4 SOR (successive over-relaxation; optional) ....................... 18
  4.5 When to stop? ............................................................. 19

5 Cholesky factorization 20
  5.1 Existence of an $LU$ factorization ................................... 20
  5.2 Cholesky: theory .......................................................... 20
  5.3 Algorithm ................................................................. 22
  5.4 Example ..................................................................... 23
Theory: Motivation

To study error, we need to start with a measure of distance between two quantities. For scalars, there is not much to say; recall that the absolute and relative errors between an approximation \( \tilde{x} \) and exact value \( x \) are:

\[
\text{abs. err.} = |x - \tilde{x}|, \quad \text{rel. err.} = \frac{|x - \tilde{x}|}{|x|}.
\]

For vectors and matrices, there are more options, and not all choices are equally advantageous for error analysis.

1 Vector and matrix norms

Hereafter, we consider two flavors of vector spaces:

- \( \mathbb{R}^n \), the space of real-valued vectors of dimension \( n \),
- \( \mathbb{R}^{m \times n} \), the space of real-valued \( m \times n \) matrices (for a given \( m \) and \( n \))

Note that each \( m \) and \( n \) defines a space, e.g. \( \mathbb{R}^{2 \times 3} \) is different from \( \mathbb{R}^{3 \times 4} \).

**Definition:** A norm on a vector space \( V \) (e.g. \( V = \mathbb{R}^n \)) is a function

\[
\| \cdot \| : V \rightarrow \mathbb{R}
\]

with the following properties:

i) \( \|x\| \geq 0 \) for all \( x \in V \) and \( \|x\| = 0 \) if and only if \( x = 0 \)

ii) \( \|ax\| = |a|\|x\| \) for all \( a \in \mathbb{R} \) and \( x \in V \)

iii) \( \|x + y\| \leq \|x\| + \|y\| \) for all \( x, y \in V \) (the **triangle inequality**)

The triangle inequality is essential for breaking up norms of expressions into simpler parts!

\(^1\)The definitions have to be adjusted for vectors/matrices with complex entries; the modifications will be developed later if needed.
1.1 Vector norms

A norm on $\mathbb{R}^n$ is called a vector norm. For instance, the Euclidean norm (or ‘2-norm’) is

$$\|x\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}.$$  

The most important vector norms are extensions of this norm:

**Definition:** Let $p \in [1, \infty)$. Then the ‘$p$-norm’

$$\|x\|_p = (x_1^p + \cdots + x_n^p)^{1/p}$$

defines a norm on $\mathbb{R}^n$.

The $\infty$-norm (or max norm) is

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$  

- The unit ball in $\mathbb{R}^n$ in a given norm is

  $$\{x \in \mathbb{R}^n : \|x\| = 1\}.$$  

- A sequence $\{x_n\} \subset V$ is said to converge to $x$ (in norm) if

  $$\lim_{n \to \infty} \|x_n - x\| = 0.$$  

- Two norms $\|\cdot\|_a$ and $\|\cdot\|_b$ are called equivalent if there are constants $m$ and $M$ such that

  $$m\|x\|_a \leq \|x\|_b \leq M\|x\|_a, \quad \text{for all } x \in V.$$  

Equivalent norms measure distance in a qualitatively similar way. For instance, if a sequence $\{x_n\}$ converges to $x$ in one norm, it also converges in any other equivalent norm.

For vectors/matrices, it turns out that there is no concern about a sequence converging in one norm but not another due to the following:

**Theorem:** Any two norms on a finite dimensional vector space $V$ are equivalent.

Throughout, we will have some results that are stated for a generic vector norm (applicable to any of them). However, we will see that depending on the problem, there are reasons to favor one particular norm over another.
1.2 Matrix norms

A **matrix norm** is a norm on $\mathbb{R}^{m \times n}$, the space of $m \times n$ matrices. One way to create a matrix norm by ‘flattening’ the matrix and using a norm for vectors of dimension $mn$, e.g.

$$\|A\| = \max_{1 \leq i,j \leq n} |a_{ij}|.$$  \hfill (1)

However, this construction lacks some important structure (see example).

The more useful approach is to take a vector norm $\|\|\|$ and define the induced norm

$$\|A\| = \max_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|.$$  \hfill (2)

It is easy to verify that (2) defines a matrix norm (satisfying the properties (i)-(iii)) for any choice of vector norm. Intuitively, the subordinate matrix norm $\|A\|$ measures the amount that $A$ can stretch a vector $x$, i.e. the maximum size $\|Ax\|$ compared to $\|x\|$.

**Notation:** It is common to use the same symbol for the vector norm $\|x\|$ and the subordinate matrix norm $\|A\|$ even though they are not the same thing. There is no ambiguity because it is clear which one to use from the dimensions of the variable.

Induced matrix norms satisfy the key inequality

$$\|Ax\| \leq \|A\|\|x\|$$  \hfill (3)

and, for norms on square matrices,

$$\|AB\| \leq \|A\|\|B\|$$  \hfill (4)

Lastly, from the definition (2) we have that $\|I\| = 1$.

For this course, we will consider only induced norms and make use of their properties. However, it is worth noting that not all matrix norms are induced by a vector norm!

**Example (induced norms):** Consider the norm on $\mathbb{R}^{n \times n}$ obtained by treating the elements of $A$ like a flat vector and using the $\infty$-norm:

$$\|A\| = \max_{1 \leq i,j \leq n} |a_{ij}|.$$  

Let

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad AB = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}.$$  

Then

$$\|A\| = \|B\| = 1, \quad \|AB\| = 2.$$  

so $\|AB\| \leq \|A\|\|B\|$ does not hold; thus this norm is not induced by any vector norm.

On the other hand (see next section),

$$\|A\|_{\infty} = \|B\|_{\infty} = 2, \quad \|AB\|_{\infty} = 3.$$
1.3 Some important matrix norms

\(\infty\)-norm

The matrix \(\infty\)-norm is the norm induced by the vector \(\infty\)-norm:

\[
\|A\|_{\infty} = \max_{\|x\|_{\infty}=1} \|Ax\|_{\infty}.
\]

The matrix \(\infty\)-norm is particularly useful because it is easy to compute:

**Computing the \(\infty\)-norm:** Let \(A\) be an \(m \times n\) matrix. Then

\[
\|A\|_{\infty} = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}|.
\]

That is, \(\|A\|_{\infty}\) is the maximum absolute row sum.

**Proof.** Suppose \(x \in \mathbb{R}^{n}\) has \(\|x\|_{\infty} = 1\). Then \(|x_j| \leq 1\) for all \(j\) so

\[
|(Ax)_i| \leq \sum_{j=1}^{n} |a_{ij}| |x_j| \leq \sum_{j=1}^{n} |a_{ij}| \text{ for } 1 \leq i \leq m.
\]

Taking the maximum over \(i\), it follows that

\[
\|Ax\|_{\infty} \leq \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}|.
\]

(5)

It remains to show that the inequality can be an equality. Let \(r\) be an index for which the maximum on the right is achieved. We can construct a vector \(x^*\) such that \(\|x^*\|_{\infty} = 1\) and the upper bound is achieved by setting

\[
x^*_j = \begin{cases} 
1 & a_{rj} > 0 \\
-1 & a_{rj} < 0 
\end{cases}.
\]

Then

\[
|(Ax^*)_r| = \sum_{j=1}^{\infty} |a_{rj}| = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}|
\]

so \(\|Ax^*\|_{\infty}\) equals the upper bound in (5), which proves the result.

1-norm

The matrix 1-norm is

\[
\|A\|_{1} = \max_{\|x\|_{1}=1} \|Ax\|_{1}.
\]

By a similar argument to the one we used for \(\|A\|_{\infty}\), it can be shown that (see HW)

\[
\|Ax\|_{1} = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}| = \text{max. abs. col. sum of } A.
\]
2-norm

The 2-norm is more complicated. For an $n \times n$ real-valued matrix, define the spectral radius $\rho(A)$ to be the magnitude of the largest eigenvalue of $A$.

**Theorem:** Let $A \in \mathbb{R}^{n \times n}$ and let $\|A\|_2$ be the matrix 2-norm. Then

$$\|A\|_2 = \sqrt{\rho(A^T A)}.$$ 

If $A$ is (real-valued and) symmetric then

$$\|A\|_2 = \rho(A) = \max_{1 \leq i \leq n} |\lambda_i|.$$ 

The proof requires some work and is omitted here. In general, $\|A\|_2$ is the magnitude of the largest singular value of $A$ (which is beyond the scope of the current discussion).

## 2 Sensitivity of linear systems

We now study the sensitivity of the linear system

$$Ax = b$$

to errors in $A$ and $b$, where $A \in \mathbb{R}^{n \times n}$ is invertible and $b \in \mathbb{R}^n$. Throughout $\|x\|$ will refer to a vector norm (any one) and $\|A\|$ will be the induced matrix norm.

Let $\tilde{x}$ be an approximate solution. To start, we assume that $\tilde{x}$ is the exact solution to a perturbed system

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

where $\delta A \in \mathbb{R}^{n \times n}$ and $\delta b \in \mathbb{R}^n$ are small perturbations to $A$ and $b$.

The goal is to find relations between the error in $\tilde{x}$ and the perturbations $\delta A$ and $\delta b$.

### 2.1 Bound with exact $A$

For the simpler case, let’s first suppose that $A$ is exact, so

$$Ax = b, \quad A\tilde{x} = b + \delta b.$$ 

Subtracting and multiplying by $A^{-1}$, we get (with $\tilde{x} = x + \delta x$)

$$\delta x = A^{-1}\delta b.$$ 

Taking the norm and using the properties of induced norms (eq. (3)), we find that

$$\|\delta x\| \leq \|A^{-1}\|\|\delta b\|.$$ 

(6)
Now to get relative errors, note that from \( \|Ax\| = \|b\| \) and (3) again we have

\[
\|x\| \geq \frac{\|A\|}{\|b\|}.
\]

Dividing this into the bound (6) yields

\[
\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A) \frac{\|\Delta b\|}{\|b\|},
\]

where \( \kappa(A) \) is the condition number \( \kappa(A) \) of the matrix \( A \), defined as

\[
\kappa(A) = \|A\|\|A^{-1}\|.
\]

If \( A \) is not invertible, then \( \kappa(A) = \infty \) by convention.

The bound says that the relative error in \( x \) (the solution) can be at most \( \kappa(A) \) times as large as the relative error in \( b \).

**Bound with perturbed \( A \)**

A similar bound can be established when \( A \) is not exact. For the curious, the details can be found in subsection 7.1. The result is that if

\[
(A + \delta A)\tilde{x} = b + \delta b
\]

and \( Ax = b \) and \( \tilde{x} = x + \delta x \) then

\[
\frac{\|\delta x\|}{\|x\|} \leq \left[ \frac{\kappa(A)}{1 - \kappa(A)(\|\delta A\|/\|A\|)} \right] \left( \frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right).
\]

There are two sources of error in the factor in square brackets:

- Numerator: The relative error in either component (\( b \) or \( A \)) is amplified by \( \kappa(A) \)
- Denominator: If \( \|\delta A\|/\|A\| \approx \kappa(A) \) then the error can also be greatly amplified; this occurs (see note below) when \( A \) is close to singular.

**A word of caution:** In general, the ‘condition number’ of a problem measures the sensitivity of that problem to errors. What we call the ‘condition number’ \( \kappa(A) \) is really the condition number for the problem of inversion, i.e. the measure of sensitivity for the calculation \( A \rightarrow A^{-1} \). Other problems involving \( A \) - for example, finding the eigenvalues of \( A \) - might have another condition number.
2.2 Properties of the condition number

The condition number is usually used with one of the $p$-norms,

$$\kappa_p(A) = \|A\|_p \|A^{-1}\|_p.$$  

As with the norms, condition numbers with different norms are equivalent; if $\kappa$ is gigantic in one norm, it is gigantic in every norm (up to some constant factor).

Note that for any norm,

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

since $\|I\| = 1$ for any induced matrix norm.

A matrix with $\kappa(A) = 1$ is called **perfectly conditioned**, as it minimizes the relative error bounds. Only a few special types of matrices are perfectly conditioned.

A matrix with $\kappa(A)$ reasonably small is called **well-conditioned**: most good algorithms will not amplify errors too much, and we need not worry about rounding errors. The definition of ‘reasonably’ small depends on context - if we need, say, 8 digits of accuracy then $\kappa(A) = 10^4$ is small since a rounding error of $10^{-16}$ will tend to give a final error that is on the order of $10^{-12}$. If 12 digits of accuracy are required then $\kappa(A) = 10^4$ is probably too large.

A matrix with $\kappa(A) \gg 1$ is called **ill-conditioned**. If possible, ill-conditioned systems should be avoided. When absolutely necessary, there are a few tricks for improving the quality of solutions. For the most part, however, the difficulty in solving the system will persist, no matter algorithm is used.

If $A$ is real, symmetric and non-singular then $\kappa_2(A)$ (the **spectral condition number**) simplifies to the nice expression

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 = \max_i |\lambda_i| \min_i |\lambda_i|,$$

which is the ratio of the largest to smallest eigenvalue (in magnitude).

**Key idea:** Another useful interpretation of $\kappa$ is that $1/\kappa$ is a measure of **how close $A$ is to being singular**. To be precise, if $E$ is the matrix closest to $A$ in the 2-norm such that $A + E$ is singular then

$$\frac{1}{\kappa_2(A)} = \frac{\|E\|_2}{\|A\|_2}.$$  

That is, there is a matrix within a relative distance $1/\kappa_2(A)$ of $A$ that is singular.

If $A$ is nearly singular and there is some error $\delta A$ then $A + \delta A$ may be singular, in which case the (perturbed) linear system may not even have a solution!
2.3 Examples

Nearness to singularity: Consider, for real numbers $\epsilon$, the matrix

$$A_\epsilon = \begin{bmatrix} 1 & 1 \\ 1 - \epsilon & 1 \end{bmatrix}.$$  

For small $\epsilon$'s, this matrix is close to the singular matrix $A_0$. The eigenvalues are

$$\lambda = 1 \pm \sqrt{1 - \epsilon}.$$  

If $0 < \epsilon \ll 1$ then the condition number is

$$\kappa_2(A) = \frac{1 + \sqrt{1 - \epsilon}}{1 - \sqrt{1 - \epsilon}} \approx \frac{4}{\epsilon},$$

which says that $A_\epsilon$ is a distance $(\epsilon/4)\|A_\epsilon\|_2 \approx \epsilon/2$ from a singular matrix. Indeed, since the smaller eigenvalue is $\lambda_\epsilon \approx \epsilon/2$,

$$A_\epsilon + E_\epsilon, \quad E_\epsilon = \begin{bmatrix} -\lambda_\epsilon & 0 \\ 0 & -\lambda_\epsilon \end{bmatrix}.$$  

is singular (and is the closest singular matrix). Ill-conditioning: The matrix

$$A = \begin{bmatrix} 1 & 1 + \epsilon \\ 1 - \epsilon & 1 \end{bmatrix}$$  

for $\epsilon \ll 1$ has $\|A\|_\infty = 1 + \epsilon \approx 1$ but

$$A^{-1} = \frac{1}{\epsilon^2} \begin{bmatrix} 1 & -1 - \epsilon \\ -1 + \epsilon & 1 \end{bmatrix}$$

so

$$\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty = \frac{1}{\epsilon^2}.$$  

If $\epsilon = 10^{-3}$ then $\kappa_\infty(A) = 10^6$, so errors in computing $Ax = b$ might grow by a factor of $10^6$ (quite a large amount!). Note that $A$ is singular when $\epsilon = 0$, so the large condition number indicates that $A$ is close to a singular matrix.

Hilbert matrix: The $n \times n$ Hilbert matrix $H$ has entries

$$H_{ij} = \frac{1}{i + j - 1}, \quad 1 \leq i, j \leq n$$  

Even though the entries of $H$ are a reasonable size, the matrix is extremely ill-conditioned. The $n$-th Hilbert matrix $H_n$ has a condition number that grows faster than $4^n$. For $n = 12$, the value of $\kappa(H_n)$ is about $10^{16}$, large enough that solving a linear system with $H_{12}$ is disastrous.
2.4 Residuals vs. error

For an approximation $\tilde{x}$ to the solution of $Ax = b$, the residual is

$$r = b - A\tilde{x}.$$  

The error vector is

$$e = x - \tilde{x}.$$  

A key difference is that if we have $A, b$ and $\tilde{x}$ but do not know the exact solution, then the error cannot be computed - but we can compute the residual! Given an approximate solution, how do we know it is sufficiently accurate? It is tempting to use the residual to estimate error, since it can be computed.

Unfortunately, although it is true that $r \equiv 0$ if and only if $\tilde{x}$ is exact, the residual and error are not always the same size. Note that the residual and error vector are related by

$$Ae = r.$$  

The relative residual $\|r\|/\|A\|\|x\|$ is related to the relative error by

$$\frac{\|e\|}{\|x\|} \leq \kappa(A) \frac{\|r\|}{\|A\|\|x\|}.$$  

Thus the relative residual is a good upper bound on the relative error $\|e\|/\|x\|$ only when the matrix is well conditioned. If $\kappa(A)$ is large, then an approx. $\tilde{x}$ with a small residual might still have a large error!
3 An important special case

3.1 Positive definite matrices

An symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called positive definite if

$$x^T A x > 0 \quad \text{for all } x \in \mathbb{R}^n, \quad x \neq 0.$$ 

If the inequality is not strict then the matrix is called positive semi-definite.

We may abbreviate ‘symmetric positive definite’ by SPD. Positive definite matrices play a key role in many important problems in mathematics and in applications, so good numerical methods that can exploit their structure are in high demand.

Note: Technically, one could generalize this to non-symmetric matrices, but the symmetric kind appears to often (and enjoys nicer properties)s, so it is typically assumed $A$ is symmetric.

SPD matrices have some equivalent characterizations:

• All the eigenvalues are positive,

• $\det(A_k) > 0$ for all the principal minors $A_k$ (the $k$ by $k$ submatrix starting from the $(1,1)$ entry down to the $(k,k)$ entry)

The second condition is often easiest to check for small matrices. Note that by the previous theorem, every positive definite matrix has an $LU$ factorization (no pivoting). If the matrix is also symmetric, we can derive an efficient algorithm to do so, which is the subject of section 5.

The first assertion (on the eigenvalues) is not so hard to prove:

Proof. Suppose $A$ is real, symmetric and has positive eigenvalues. We know from the spectral theorem that $A$ has an orthogonal basis of eigenvectors $v_1, \cdots, v_n$ for the eigenvalues $\lambda_1, \cdots, \lambda_n$. Now let $x \in \mathbb{R}^n$; write it in terms of the eigenvectors basis as

$$x = \sum_{i=1}^n c_i v_i.$$ 

Then, using that $Av_i = \lambda_i v_i$ we have that

$$x^T A x = \left( \sum_{i=1}^n c_i v_i \right) \cdot \left( \sum_{i=1}^n \lambda_i c_i v_i \right) = \sum_{i,j} \lambda_i c_i c_j (v_i \cdot v_j) = \sum_{i=1}^n \lambda_i c_i^2 v_i \cdot v_i.$$ 

But $v_i \cdot v_i = \|v_i\|^2 > 0$ so each term in the sum is positive, so $x^T A x > 0$.

For the reverse, suppose $A$ is SPD. Since $A$ is real symmetric, its eigenvalues are real. Let $\lambda$ be an eigenvalue with eigenvector $v$. Then

$$0 < v^T A v = v^T \lambda v = \lambda \|v\|^2$$ 

so $\lambda > 0$. It follows that all the eigenvalues are positive. \qed
Example: We show that the matrix

\[
A = \begin{bmatrix}
2 & -1 & 0 \\
-1 & 2 & -1 \\
0 & -1 & 2 \\
\end{bmatrix}
\]

is positive definite. The principal minors are

\[A_1 = [2], \quad A_2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \quad A_3 = A.\]

The determinants are \(\det(A_1) = 2\) and \(\det(A_2) = 3\) and \(\det(A_3) = 2 \cdot 3 - (-1) \cdot (-2) = 4\). Since all these values are positive, \(A\) is positive definite.

### 3.2 Diagonal dominance

Positive definiteness is difficult to check in practice because one must either compute determinants (e.g. using Gaussian elimination) or compute eigenvalues (not easy).

A square matrix \(A\) is called **diagonally dominant** if

\[|a_{ii}| > \sum_{j=1, j \neq i}^{n} |a_{ij}|\]

i.e. the diagonal entry is larger than the (absolute) sum of all other entries in that row.

Every symmetric diagonally dominant matrix is symmetric positive definite. If the inequality is \(\geq\) then the matrix is instead semi-definite (\(\lambda \geq 0\)).

**Warning:** The converse is not true! SPD matrices are not always diagonally dominant.

However, this fact provides a convenient way to prove some matrices are SPD. Often, in applications, one needs a matrix to be SPD but it ends up being diagonally dominant as well.

Example: The matrix

\[
A = \begin{bmatrix}
2 + \delta & -1 & 0 \\
-1 & 2 + \delta & -1 \\
0 & -1 & 2 + \delta \\
\end{bmatrix}
\]

is diagonally dominant for any \(\delta > 0\) and is therefore SPD. Matrices of this form arise often when partial differential equations. Note that it is positive definite when \(\delta = 0\), but the diagonal dominance theorem only tells us that \(\lambda \geq 0\), not \(\lambda > 0\) since \(2 - |1| + |1| = 0\).
3.3 Gerschgorin’s theorem (optional)

More generally, it is true that the amount of diagonal dominance tells us how far the eigenvalues must be from zero in a specific way.

(Gerschgorin Circle Theorem): Let $A \in \mathbb{R}^{n \times n}$. For each row, define

$$r_i = \sum_{j=1, j \neq i}^{n} |a_{ij}|.$$ i.e. the sum of the absolute values of non-diagonal entries in that row. Define the ‘circles’

$$D_i = \{ z \in \mathbb{C} : |z - a_{ii}| < r_i \}.$$ Then every eigenvalue of $A$ lies in the union of the circles.

For instance, for

$$A = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 5 & 2 \\ 0 & 2 & 3 \end{bmatrix}$$ The radii are $r_1 = 1, r_2 = 3$ and $r_3 = 1$. Since $A$ is symmetric, the eigenvalues are real, so we only need to consider the intervals

$$D_1 = (2, 4), \quad D_2 = (2, 8), \quad D_3 = (1, 5).$$ Taking the union, we find that all the eigenvalues are in the interval $(1, 8)$.

4 Iterative methods

The methods we have looked at so far are direct methods, in which the system is reduced to something trivial in a finite number of steps. With exact arithmetic, an exact solution is produced after a finite amount of work.

Another approach to solving $Ax = b$ is to devise an iterative method, where we successively improve an approximation that will converge to the true solution.

**Key motivation:** Typically, a direct method has to be run to completion to have a usable answer, but this answer is likely to be as accurate as we can get. On the other hand, iterative methods may get satisfactorily ‘close’ to the exact solution within only a few (simple) steps - even though it may take an infinite number of steps to reach the exact solution.

Here we consider the most basic iterative methods, which have the form

$$M x^{(k+1)} = N x^{(k)} + b \quad (7)$$
where \( A = M - N \) is a splitting of \( A \) into the sum of two matrices. To determine what to choose, we need some theory to say when the iteration will converge. Write

\[
x^{(k+1)} = M^{-1} Nx^{(k)} + M^{-1} b = Tx^{(k)} + c
\]

where \( T = M^{-1}N \). This is the iteration matrix, which will determine the convergence properties. Let \( e^{(k)} = x^{(k)} - x \) be the error vector. Then

\[
e^{(k+1)} = Te^{(k)}.
\]

Taking norms (any one), we find that

\[
\|e^{(k+1)}\| \leq \|T\|\|e^{(k)}\| \implies \|e^{(k)}\| \leq \|T\|^k\|e^{(0)}\|.
\]

(8)

It follows that if \( \|T\| < 1 \) in any subordinate matrix norm, then the iteration will converge to \( x \), i.e. \( \|x^{(k)} - x\| \to 0 \) as \( k \to \infty \). Note that once we have established convergence in one norm, it follows that the iteration converges in all norms.

A more precise statement makes use of the following theorem:

**Theorem:** Let \( \rho(A) = \max |\lambda| \) be the spectral radius. Then

\[
\rho(A) = \inf_{\|\cdot\|} \|A\|
\]

where the infimum is taken over all induced matrix norms.

In particular, it follows that \( \rho(A) \) is the ’smallest’ norm in the sense that

i) If \( \|\cdot\| \) is any induced matrix norm then \( \rho(A) \leq \|A\| \).

ii) If \( \rho(A) < 1 \) then there is an induced matrix norm for which \( \rho(A) < \|A\| < 1 \) (i.e. a norm that, applied to \( A \), is slightly bigger than \( \rho(A) \)).

By result (ii), if the iteration matrix \( T \) satisfies \( \rho(T) < 1 \) then the iteration (7) converges, because we can find a norm for which \( \|T\| < 1 \) and (8) holds.

We summarize this result in a theorem:

**Theorem (convergence of simple iterative methods):** Let \( A = M - N \) be a splitting of \( A \) and let \( T = M^{-1}N \) be the iteration matrix for the iteration

\[
Mx^{(k+1)} = Nx^{(k)} + b.
\]

If \( \rho(T) < 1 \) and \( A \) is invertible then this iteration will converge to the solution to \( Ax = b \) for any initial vector \( x^{(0)} \).

\(^2\)See e.g. Kincaid & Cheney, Sec. 4.6
4.1 Jacobi method

Note: I’ve put the worked examples for the two methods after the theory (subsection 4.3).

Theory

Let \(L_A, U_A\) and \(D\) be the lower/upper triangular and diagonal parts of \(A\) (not the same as in the LU factorization!):

\[
L_A = \begin{bmatrix}
0 & 0 & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0
\end{bmatrix}, \quad U_A = \begin{bmatrix}
0 & a_{12} & a_{13} & \cdots & a_{1n} \\
0 & 0 & a_{23} & \ddots & a_{2n} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix}
\]

and \(D = \text{diag}(a_{11}, \ldots, a_{nn})\).

For Jacobi’s method we take \(M = D\) and \(N = -(L_A + U_A) = D - A\) (i.e. \(-A\) but without the diagonal entries), leading to

\[
x^{(k+1)} = -D^{-1}(L_A + U_A)x^{(k)} + D^{-1}b.
\]

It is straightforward to show the following:

**Theorem:** If \(A\) is diagonally dominant then

\[
\rho(D^{-1}(L_A + U_A)) < 1
\]

and so the Jacobi iteration converges to the solution to \(Ax = b\) for any starting vector \(x^{(0)}\).

**Proof.** Since \(A\) is diagonally dominant we have

\[
\|T\|_\infty = \|D^{-1}(L_A + U_A)\|_\infty = \max_{1 \leq i \leq n} \frac{1}{|a_{ii}|} \left( \sum_{j=1, j \neq i}^{n} |a_{ij}| \right) < 1.
\]

It follows from this inequality that Jacobi’s method converges; the theorem on the spectral radius shows that \(\rho(T) < \|T\|_\infty < 1\). \(\square\)

4.1.1 Implementation

In practice, there is a simple way to compute the iteration. Write out \(Ax = b\):

\[
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{n,n}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix} =
\begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_n
\end{bmatrix}
\]
Now we ‘solve’ for $x_i$ in the $i$-th equation:

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1,j \neq i}^{n} a_{ij} x_j \right).$$

Jacobi’s method uses the values of $x^{(k)}$ on the right side to get $x^{(k+1)}$:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1,j \neq i}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, \cdots n.$$

The entries ‘evaluated’ at $k+1$ are labeled in red. Note that we do not overwrite $x_i^{(k)}$ with its updated value, because the old value is still needed to update all the other components.

4.2 Gauss-Seidel

We can, however, use the updated values of $x_i$ immediately as they become available, which leads to the Gauss-Seidel method.

Implementation

Proceed as in the Jacobi iteration, but use the updated components of $x$ immediately as they become available (again, entries ‘evaluated’ at $k+1$ are labeled in red; these use the updated values):

$$
\begin{pmatrix}
  a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\
  a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  a_{n1} & a_{n2} & \cdots & \cdots & a_{n,n}
\end{pmatrix}
\begin{pmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{pmatrix}
= 
\begin{pmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_n
\end{pmatrix}.
$$

The formulas for the update are

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=i-1}^{n} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right). \quad (9)$$

Note that when computing the $i$-th component of the updated $x^{(k+1)}$, the only components of $x^{(k+1)}$ needed are the ones we have already computed (from 1 to $i-1$). **Note that** the update must be computed in the order $i = 1, \cdots n$ (or backwards, with $i$ decreasing from $n$ to 1).

**Implementation:** The update (10) can be simply coded as

$$x_i \leftarrow \frac{1}{a_{ii}} \left( b_i - \sum_{j=1,j \neq i}^{n} a_{ij} x_j \right), \quad i = 1, \cdots n$$

16
or even simpler as
\[ x_i \leftarrow x_i + \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{n} a_{ij} x_j \right). \]

The simplicity makes Gauss-Seidel an appealing choice for an iterative method. Note that using the updated values immediately is convenient. However, other, more sophisticated iterative methods are used more often in practice (see homework).

**Theory**

Gauss-Seidel can also be written in matrix form. Let \( A = L_A + D + U \) as before, and take \( M = L_A + D \):
\[ x^{(k+1)} = -(L_A + D)^{-1} U_A x^{(k)} + (L_A + D)^{-1} b. \]

The matrix for the iteration is then \( T = -(L_A + D)^{-1} U_A \). Like Jacobi, if \( A \) is diagonally dominant then Gauss-Seidel converges (see textbook). Another related result is that

**Theorem (convergence of GS):** If \( A \) is symmetric positive definite then, for the Gauss-Seidel iteration, \( \rho(T) < 1 \) so the iteration converges to \( A^{-1} b \) for any starting vector \( x^{(0)} \).

The proof involves showing that \( \rho(T) < 1 \), which takes a bit of work. The theorem is notable because SPD matrices appear so often in applications. The matrix \( A \) does not have to be symmetric or positive definite for GS to work - but the conditions for convergence are then more complicated.

**4.3 An example**

We use the Jacobi and Gauss-Seidel methods to solve \( Ax = b \) where
\[ A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ -3 \end{bmatrix}. \]

The exact solution is \( x = (3/2, -3/2)^T \). Choose \( x^{(0)} = (0, 0)^T \) as a starting vector.

For Jacobi’s method, we have
\[ x_1^{(k+1)} = \frac{1}{a_{11}} \left( b_1 - a_{12} x_2^{(k)} \right) = \frac{1}{3} \left( 3 - x_2^{(k)} \right), \]
\[ x_2^{(k+1)} = \frac{1}{a_{22}} \left( b_2 - a_{21} x_1^{(k)} \right) = \frac{1}{3} \left( -3 - x_1^{(k)} \right). \]

Computing the first few iterates gives
\[ x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad x^{(1)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad x^{(2)} = \begin{bmatrix} 4/3 \\ -4/3 \end{bmatrix}, \ldots. \]
The $\infty$-norms of the error, $\|e^{(k)}\|_\infty = \|x^{(k)} - \bar{x}\|_\infty$ are
\[\|e^{(0)}\|_\infty = 3/2, \quad \|e^{(1)}\|_\infty = 1/2, \quad \|e^{(2)}\|_\infty = 1/6, \ldots\]

For Gauss-Seidel, the 1-st component of $x^{(k+1)}$ gets used when computing $x^{(k+1)}_2$:
\[
x^{(k+1)}_1 = \frac{1}{a_{11}} \left( b_1 - a_{12}x^{(k)}_2 \right) = \frac{1}{3} \left( 3 - x^{(k)}_2 \right),
\]
\[
x^{(k+1)}_2 = \frac{1}{a_{22}} \left( b_2 - a_{11}x^{(k+1)}_1 \right) = \frac{1}{3} \left( -3 - x^{(k+1)}_1 \right).
\]

Computing the first few iterates gives
\[
x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad x^{(1)} = \begin{bmatrix} 1 \\ -4/3 \end{bmatrix}, \quad x^{(2)} = \begin{bmatrix} 1.44 \\ -1.48 \end{bmatrix}, \ldots.
\]
\[\|e^{(0)}\|_\infty = 3/2, \quad \|e^{(1)}\|_\infty = 1/2, \quad \|e^{(2)}\|_\infty = 0.06, \ldots\]

We can prove that Gauss-Seidel converges faster. Let $T_J$ and $T_G$ be the iteration matrices for the two methods. Then
\[T_J = D^{-1}(D - A) = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1/3 \\ -1/3 & 0 \end{bmatrix}\]
which has eigenvalues $\lambda = \pm 1/3$, so $\rho(T_J) = 1/3$. For Gauss-Seidel,
\[T_G = -(L_A + D)^{-1}U_A = \begin{bmatrix} 3 & 0 \\ 1 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1/3 \\ 0 & 1/9 \end{bmatrix}\]
which has eigenvalues 0 and 1/9, so $\rho(T_G) = 1/9$. Thus Gauss-Seidel converges much faster, with a rate of 1/9 (fairly good!) compared to 1/3.

### 4.4 SOR (successive over-relaxation; optional)

A disadvantage of Gauss-Seidel is that if $\rho(T)$ is close to 1, the method will converge rather slowly (too slow to be of any use). One way to correct this is to adjust the splitting. For a relaxation parameter $\omega$, use the update
\[
x^{(k+1)}_i = \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1,j<i}^n a_{ij}x^{(k+1)}_j - \sum_{j=1,j>i}^n a_{ij}x^{(k)}_j \right) + (1 - \omega)x^{(k)}_i.
\]
(10)

If $\omega = 1$ this is Gauss-Seidel; otherwise the method ‘relaxes’ the update by leaving a part of the old value (the $(1 - \omega)x^{(k)}_i$ part). In matrix terms,
\[
M_\omega x^{(k)} = N_\omega x^{(k)} + \omega b
\]
where
\[
M_\omega = D + \omega L_A, \quad N_\omega = M_\omega - \omega A.
\]
It can be shown that if $0 < \omega < 2$ and $A$ is sym. positive definite then the method converges. Ideally, one could find a value $\omega \in (0, 2)$ so that
\[
\rho(I - \omega M_\omega^{-1}A) < 1
\]
is minimized, which can greatly accelerate convergence. For some systems, such optimization is possible, but in general it is difficult to find the right value of $\omega$. 

18
4.5 When to stop?

According to the theory, we have error bounds of the form

$$\|e^{(k)}\| \leq C\|T\|^k$$

where $C$ depends on the initial error. Typically, it is true that the best bound uses the spectral radius instead of $\|T\|$ and that this is the ‘true’ decay rate of the error:

$$\|e^{(k)}\| \approx C \rho(T)^k.$$ 

Thus every step $k$ is expected to reduce the error by about a factor of $\rho(T)$. However, we do not know the constant, as we cannot calculate the error vector without the exact solution.

Knowing when to stop involves using some heuristics and educated guesses. As we saw with Newton’s method, two choices for stopping criteria are

$$\|x^{(k+1)} - x^{(k)}\| \leq \epsilon_{\text{rel}}\|x^{(k)}\|$$

and/or

$$\|x^{(k+1)} - x^{(k)}\| \leq \epsilon_{\text{abs}}$$

where $\epsilon_{\text{rel}}$ and $\epsilon_{\text{abs}}$ are relative and absolute tolerances chosen in advance. Due to the linear convergence, the ‘tolerance’ is not a bound on the error. If $\rho(T)$ is close to one, we may have to pick tolerances much less than the desired accuracy.

One can be a bit more precise about all this (e.g. monitor how fast the solution is converging, estimate the error from quantities we can compute like $\|x^{(k+1)} - x^k\|$ and so on); see the homework for one example.

A third condition is to stop when the residual is small:

$$\|Ax^{(k)} - b\| < \epsilon$$

(or a relative version). This condition is useful if a small residual is what matters, and has the benefit of being computable. As we saw, if the condition number is large, then it may be a serious underestimate of the actual error.
5 Cholesky factorization

5.1 Existence of an $LU$ factorization

It is useful to know when pivoting is not needed when factoring $PA = LU$ (so that the factorization is simply $A = LU$). A simple (in theory) sufficient condition is the following. Define the $k$-th principal minor of $A$ to be the $k$ by $k$ submatrix from the $(1, 1)$ entry to the $(k, k)$ entry, i.e.

$$A_k = \begin{bmatrix} a_{11} & \cdots & a_{kk} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix}$$

**Theorem.** If the principal minors $A_k$ for $k = 1, \cdots, n$ are non-singular, then $A$ has a factorization $A = LU$.

The factorization could be computed using Gaussian elimination with no pivoting. (Note: with pivoting on, some rows might be swapped around). Note that if $A$ is either diagonally dominant and symmetric or SPD then the theorem guarantees that is has an LU factorization without pivoting. We can exploit this to greatly improve the LU algorithm!

5.2 Cholesky: theory

When $A$ is symmetric and positive definite, the $LU$ factorization has additional structure.

**Theorem:** A symmetric positive definite $A \in \mathbb{R}^{n \times n}$ has a **Cholesky factorization**

$$A = GG^T$$

where $G$ is lower triangular, invertible, and has positive entries on the diagonal.

Moreover, the factorization is unique.

The proof of existence relies on the existence of the $LU$ factorization (without pivoting).

Given $A = LU$, it is fairly easy to show that this is in fact the Cholesky factorization in disguise.

**Claim:** If $A \in \mathbb{R}^{n \times n}$ is SPD and $A = LU$ where $L$ is unit lower triangular and $U$ is upper triangular, then

$$U = DL^T$$

where $D$ is a diagonal matrix with positive entries; then $A = GG^T$ where $G = LD^{1/2}$.

**Proof.** Since $A$ is symmetric,

$$LU = U^TL^T.$$
Since \( \det(L) = 1 \), we know \( L \) is invertible, so
\[
U(L^T)^{-1} = L^{-1}U^T.
\]
Lower triangular matrices are closed under products and inverses, so the above is
\[
\text{lower tri.} = \text{upper tri.}.
\]
But only diagonal matrices are both lower/upper triangular, so there is such a matrix,
\[
D = \text{diag}(d_1, \ldots, d_n),
\]
such that
\[
U(L^T)^{-1} = L^{-1}U^T = D.
\]
Thus \( A = LU = LDL^T \). Now (see note below) since \( A \) is SPD, \( D \) is SPD as well, hence it must have positive entries.

This means that
\[
D^{1/2} = \text{diag}(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}})
\]
exists\(^3\), so
\[
A = LD^{1/2}(L^{1/2}D^{1/2})^T = GG^T
\]
for the upper triangular matrix \( G = LD^{1/2} \).

**Note:** To show the remaining claim, suppose that \( A \) is positive definite. We show that
\[
A = LBL^T, \quad L \text{ invertible} \implies B \text{ is SPD}.
\]
Symmetry is trivial; take the transpose to get \( LBL^T = A = A^T = LB^T L^T \). Since \( L \) is invertible (as is \( L^T \)), \( B = B^T \).

Now let \( x \in \mathbb{R}^n \) with \( x \neq 0 \). Then
\[
0 < x^T Ax = x^T LBL^T x = y^T By
\]
where \( y = Lx \). Since \( L \) is invertible and the above holds for all \( x \in \mathbb{R}^n \) it also holds for all \( y \in \mathbb{R}^n \), i.e. \( B \) is positive definite.

**Uniqueness:** A similar argument proves uniqueness. Suppose \( A \) is SPD and there are two Cholesky factorizations:
\[
A = GG^T = MM^T.
\]
First, we show that \( G \) and \( M \) differ by a diagonal matrix \( (G = MD) \) and then show that the only possible scaling is the identity \( (D = I)) \).

---

\(^3\)In general, \( B = A^{1/2} \) means that \( B \) is a matrix such that \( B^2 = A \). The matrix \( A^{1/2} \) is, as you might expect, called a ‘square root’ of the matrix.
For the first part,
\[ M^{-1}G = M^T(G^T)^{-1}. \]
The LHS is lower tri. and the RHS is upper tri. so there is a diagonal matrix \( D \) such that
\[ M^{-1}G = M^T(G^T)^{-1} = D. \]
Since \( m_{ii}, g_{ii} > 0 \) and \( M^{-1}, G \) are lower triangular we have
\[ d_{ii} = g_{ii}/m_{ii} \]
(check via matrix multiplication!) so \( d_{ii} > 0 \). Comparing the two factorizations,
\[ GG^T = (MD)(MD)^T = MD^2M^T \]
so \( D^2 = 1 \). But \( d_{ii} > 0 \) so it must be that \( d_{ii} = 1 \), i.e. \( D = I \), and so \( G = M \).

5.3 Algorithm

Gaussian elimination could be used; for variety, we give an alternate derivation that yields a different method. First, we write
\[
A = GG^T = \begin{bmatrix}
g_{11} & 0 & \ldots & \ldots & 0 
g_{21} & g_{22} & 0 & \ldots & 0 
g_{31} & g_{32} & \ddots & \ddots & \vdots 
\vdots & \vdots & \ddots & g_{n-1,n-1} & 0 
g_{n1} & g_{n2} & \ldots & g_{n,n-1} & g_{nn}
\end{bmatrix}
\]
\[
\begin{bmatrix}
g_{11} & g_{21} & g_{31} & \ldots & g_{n1} 
g_{22} & 0 & g_{23} & \ldots & g_{n2} 
g_{32} & g_{33} & 0 & \ldots & g_{n3} 
\vdots & \vdots & \vdots & \ddots & \vdots 
g_{n2} & g_{n3} & \ldots & g_{n,n-1} & 0
\end{bmatrix}
\]
in component form:
\[ a_{jk} = \sum_{i=1}^{n} g_{ji}g_{ki} = \sum_{i=1}^{k} g_{ji}g_{ki}, \quad j \geq k. \]
Note that since \( A \) is symmetric, the rest of the equations are redundant.

We can solve for the entries in \( G \) in sequence, starting with the first column. Writing out the equations to see the pattern, we get
\[
a_{11} = g_{11}^2 
a_{21} = g_{21}g_{11} 
\vdots 
a_{n1} = g_{n1}g_{11}
\]
\[
a_{22} = g_{21}^2 + g_{22}^2 
a_{32} = g_{31}g_{21} + g_{32}g_{22} 
\vdots 
a_{n2} = g_{n1}g_{21} + g_{n2}g_{22}
\]
\[ \text{Note: the approach here also works for LU factorization in general, yielding a variant of GE with the loop order rearranged. The algorithms are called Crout's method or Doolittle's method.} \]
It is clear from this pattern that to obtain the \( k \)-th column of \( G \), we first solve for the diagonal entry \( g_{kk} \) and then the entries below it (since the equations for these entries depend only on the first \( k - 1 \) columns and \( g_{kk} \)).

To get the \( k \)-th column, first solve for \( g_{kk} \) using

\[
g_{kk}^2 = a_{kk} - \sum_{i=1}^{k-1} g_{ki}^2
\]

and then for \( j = k + 1, \ldots, n \):

\[
g_{jk} = \frac{1}{g_{kk}} \left( a_{jk} - \sum_{i=1}^{k-1} g_{ji}g_{ki} \right).
\]

So long as the computed \( g_{kk} \)'s are positive, the iteration will go to completion, yielding the Cholesky factorization. It can be shown that the Cholesky factorization algorithm is numerically stable (unlike GE) and no pivoting is necessary.

**Note on storage:** As with Gaussian elimination, we can save some space by updating \( A \) directly instead of having a separate \( G \). In the updates, we store \( g_{jk} \) in the entry \( a_{jk} \).

---

**Algorithm 1** Cholesky factorization (overwriting)

**Input:** \( A \in \mathbb{R}^{n \times n} \), sym. and positive definite  
**Output:** \( G \) such that \( A = GG^T \) (stored in the lower. tri. part of \( A \))

\[
\text{for } k = 1 : n \text{ do} \\
\quad a_{kk} = \sqrt{a_{kk} - \sum_{i=1}^{k-1} a_{ki}^2} \\
\quad \text{for } j = k + 1 : n \text{ do} \\
\qquad a_{jk} = \frac{1}{a_{kk}} \left( a_{jk} - \sum_{i=1}^{k-1} a_{ji}a_{ki} \right) \\
\quad \text{end for} \\
\text{end for}
\]

---

**5.4 Example**

Let

\[
A = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 5 & 6 \\ 2 & 6 & 17 \end{bmatrix}.
\]

Note that, for the principal minors, \( \det(A_1) = 1 \) and \( \det(A_2) = 5 - 1 = 4 \) and \( \det(A_3) = 36 \), all of which are positive, so \( A \) is positive definite. We apply the algorithm to obtain the Cholesky factorization:
First column: \( g_{11} = \sqrt{a_{11}} = 1 \), and

\[
g_{21} = \frac{1}{g_{11}} (a_{21}) = 1/1 = 1, \quad g_{31} = \frac{1}{g_{11}} a_{31} = 2.
\]

We now have \((x\) denoting unknown entries\)

\[
G = \begin{bmatrix}
1 & 0 & 0 \\
1 & x & 0 \\
2 & x & x
\end{bmatrix}.
\]

Second column: \( g_{22} = \sqrt{a_{22} - g_{21}^2} = 2 \) and

\[
g_{32} = \frac{1}{g_{22}} (a_{32} - g_{31} g_{21}) = 2
\]

which gives

\[
G = \begin{bmatrix}
1 & 0 & 0 \\
1 & 2 & 0 \\
2 & 2 & x
\end{bmatrix}.
\]

Finally, \( g_{33} = \sqrt{a_{33} - g_{31}^2 - g_{32}^2} = 3 \). Thus \( A = G G^T \) where

\[
G = \begin{bmatrix}
1 & 0 & 0 \\
1 & 2 & 0 \\
2 & 2 & 3
\end{bmatrix}.
\]

**Extra note:** It is possible to do pivoting to move the largest entry in the diagonal to the right place. To preserve symmetry, we must consider only **symmetric pivoting**:

\[
A \rightarrow PAP^T
\]

If \( P \) swaps rows \( i \) and \( j \) then \( PAP^T \) swaps the \((j,j)\) entry with the \((i,i)\) entry of \( A \). Pivoting is useful when obtaining the Cholesky factorization when some of the \( d_i \)'s encountered are nearly zero.
6 Full rank least squares, briefly

See Section 6.1-6.2 of the book for the details.

6.1 Derivation of the normal equations

Let $A$ be a full rank $m \times n$ matrix and $b \in \mathbb{R}^m$ with $m > n$. We wish to find

$$x_{LS} = \arg\min_{x \in \mathbb{R}^n} \|Ax - b\|_2.$$ 

Define

$$\phi(x) = \frac{1}{2} \|Ax - b\|_2^2 = \frac{1}{2} \sum_{j=1}^{n} ((Ax)_j - b_j)^2.$$ 

Then, equivalently, we seek a minimum of $\phi$. This function has a unique minimum (straightforward to show; see textbook sec. 6.1) where $\nabla \phi = 0$. To compute the gradient,

$$\frac{\partial \phi}{\partial x_j} = \sum_{i=1}^{n} (Ax - b)_i \frac{\partial}{\partial x_j} (Ax - b)_i = \sum_{i=1}^{n} (Ax - b)_j a_{ji} = (A^T(Ax - b))_j$$

so $0 = \nabla \phi = A^T(Ax - b)$ which yields the normal equations

$$A^T Ax = A^T b. \quad (11)$$

The matrix $M = A^T A$ is $n \times n$ (so it is independent of the length of the 'long' vector $b$). Moreover, we claim that

$$A \text{ full rank } \implies M \text{ is SPD.}$$

Symmetry is trivial. To show $M$ is SPD, let $x \in \mathbb{R}^n$ with $x \neq 0$. Then

$$x^T M x = (Ax)^T A x = \|Ax\|_2^2.$$ 

Since $A$ is full rank, there is no non-trivial $x \in \mathbb{R}^n$ such that $Ax = 0$ (or else the columns would be linearly dependent), so $\|Ax\| \neq 0$ and thus $x^T M x > 0$.

It follows that the normal equations (11) have a unique solution, and that we can obtain the solution using the Cholesky factorization. Simply find $G$ such that

$$M = GG^T$$

(note that $G \neq A^T$ since $A$ is not square!). Then solve

$$Gy = A^T b, \quad G^T x = y$$

to obtain $x$. The amount of work required is

$$\frac{1}{3} n^3 + O(n^2) + mn^2 = \frac{1}{3} n^3 + mn^2 + O(n^2)$$

for obtaining $G$ (first term), solving the two system ($O(n^2)$), and computing $A^T b$. Typically, $m$ is much larger than $n$ so most of the work actually goes into the $A^T b$ computation (which is the only place where length-$m$ vectors are involved).
7 Additional (optional) notes

7.1 Proof of error bound

First, we need a technical lemma regarding inverses of perturbed matrices:

**Lemma** If $A \in \mathbb{R}^{n \times n}$ and $\|A\| < 1$ then $I - A$ is invertible and  
\[
\|(I - A)^{-1}\| \leq \frac{1}{1 - \|A\|}.
\]

*Proof.* The proof involves expanding $(I - A)^{-1} = I + A + A^2 + \cdots$ and using $\|A\| < 1$ to show convergence. For the bound, use the triangle inequality and the fact that $\|A^n\| \leq \|A\|^n$ by property (4).

The main result\(^5\) is:

**Theorem** Suppose the approximation $\tilde{x}$ to $x$ solving $Ax = b$ satisfies  
\[
(A + \Delta A)\tilde{x} = b + \Delta b
\]
where $\|\Delta A\| \leq \epsilon \|A\|$, $\|\Delta b\| \leq \epsilon \|b\|$ and $\epsilon \kappa(A) < 1$. Then  
\[
\frac{\|\tilde{x} - x\|}{\|x\|} \leq \frac{2\epsilon \kappa(A)}{1 - \epsilon \kappa(A)}
\]

**Note:** The conditions $\|\Delta A\| \leq \epsilon \|A\|$ and $\epsilon \kappa(A) < 1$ are needed to guarantee that  
\[
A + \Delta A = A(I + A^{-1} \Delta A)
\]
is invertible since then $\|A^{-1} \Delta A\| < 1$, which allows us to apply the lemma from earlier. If $\kappa(A)$ is too large, then $A + \Delta A$ may fail to be invertible and the analysis fails.

*Proof.* According to the lemma, $(I + A^{-1} \Delta A)$ is invertible, so  
\[
\tilde{x} = (I + A^{-1} \Delta A)^{-1}(x + A^{-1} \Delta b).
\]
Taking the norm and using the lemma,  
\[
\|\tilde{x}\| \leq \frac{1}{1 - \|A^{-1} \Delta A\|} \left(\|x\| + \|A^{-1}\| \|\Delta b\|\right).
\]

\(^5\)see Golub and Van Loan, *Matrix Computations*
Now \( \| \Delta b \| \leq \epsilon \| b \| \leq \| A \| \| x \| \) and \( \| A^{-1} \Delta A \| \leq r \), so
\[
\| \tilde{x} \| \leq \frac{1}{1-r}(\| x \| + \epsilon \| A^{-1} \| \| x \| ) = \frac{1+r}{1-r} \| x \|. \tag{13}
\]
To bound the relative error, we write
\[
\Delta x = A^{-1} \Delta b - A^{-1} \Delta A \tilde{x}
\]
and take norms to obtain
\[
\| \Delta x \| \leq \epsilon \| A^{-1} \| \| b \| + \epsilon \kappa A \| \tilde{x} \| .
\]
Combining this with the bound (13) on \( \tilde{x} \) we obtain the result:
\[
\frac{\| \Delta x \|}{\| x \|} \leq \epsilon \kappa(A) + \epsilon \kappa A \frac{1+r}{1-r} = \frac{2\epsilon}{1-r} \kappa(A).
\]

7.2 Numerical stability: Gaussian elimination (details)

When a problem is ill-conditioned, we cannot hope to do much better than the error bound suggests, regardless of the method. But an algorithm can be unstable, even on well-conditioned problems.

For example, suppose \( Ax = b \) is solved using Gaussian elimination to obtain an approximation \( \tilde{x} \) and that
\[
(A + \Delta A) \tilde{x} = b + \Delta b.
\]
Let us now assume that \( A \) and \( b \) were represented exactly, so \( \Delta A \) and \( \Delta b \) are due to rounding errors and propagation of those errors through the algorithm.

According to the theory,
\[
\frac{\| \tilde{x} - x \|_{\infty}}{\| x \|_{\infty}} \leq C \epsilon \kappa_{\infty}(A) \tag{14}
\]
for a constant \( C \), where \( \epsilon \) is a bound on the perturbations:
\[
\| \Delta A \| \leq \epsilon \| A \|, \quad \| \Delta b \| \leq \epsilon \| b \|.
\]
The process of determining a bound on \( \Delta A \) (i.e. the \( \epsilon \) here) is called backward error analysis, developed by Wilkinson in the 1960s. The result is the following\(^6\):

**Theorem.** If \( Ax = b \) is solved using Gaussian elimination with partial pivoting to obtain a solution \( \tilde{x} \) and computed factorization \( PA \approx LU \), then
\[
(A + \Delta A) \tilde{x} = b
\]
for a matrix \( \Delta A \) such that
\[
\frac{\| \Delta A \|_{\infty}}{\| A \|_{\infty}} \leq 3n\delta + 5n^2 \delta \frac{\| U \|_{\infty}}{\| A \|_{\infty}} + O(\delta^2)
\]
where \( \delta \) is machine epsilon.

\(^6\)See Golub and Van Loan, *Matrix Computations.*
(Recall that an arithmetic operation leads to a relative error of size \( \delta \).)

The elements of \( \tilde{U} \) can be quite large relative to the elements of the original matrix \( A \) (e.g. in the example from homework, they grow as large as \( 2^{n-1} \)). Combining this result with the bound (14), we get

\[
\frac{\|x - \tilde{x}\|_\infty}{\|x\|_\infty} \leq c_n \delta \|\tilde{U}\|_\infty \kappa(A)
\]

where \( c_n \) is an unimportant factor. Note that the \( \|\tilde{U}\| \) term is due to rounding errors in the algorithm, and the \( \kappa(A) \) term (as discussed) is due to the inherent sensitivity of the problem.

Note that the numerical instability is not due to ill-conditioning (although an ill-conditioned matrix makes things worse). For a simple example without pivoting, recall we found that

\[
A = \begin{bmatrix}
\epsilon & 1 \\
1 & 1
\end{bmatrix}
\]

caused problems for Gaussian elimination. The condition number (in the \( \infty \) norm) is

\[
\kappa_\infty(A) = \|A\| \|A^{-1}\| = 1 \cdot \frac{2}{1-\epsilon} \approx 2
\]

so the matrix is well conditioned (on the other hand, the largest element in \( \tilde{U} \) is \( 1-1/\epsilon \sim -1/\epsilon \) which is quite large, and that was the problem).