Basics in Applied Mathematics | Winter term 2026/27 | University of Freiburg

Part I: Numerics

Lecture notes

JProf. Dr. Diyora Salimova

Department of Applied Mathematics, University of Freiburg

Course homepage:
https://pfaffelh.github.io/hp/2026WS_BAM.html

Version of October 25, 2026.

Contents

Preface iv

1 Introduction, floating-point arithmetic, conditioning and norms 1
1.1 What this part of the course is about . . . 1
1.2 Floating-point numbers . . . 2
1.2.1 Representation . . . 2
1.2.2 Rounding and the standard model . . . 3
1.2.3 Cancellation . . . 4
1.3 Conditioning of a problem and stability of an algorithm . . . 5
1.3.1 Condition numbers . . . 5
1.3.2 Stability of an algorithm . . . 7
1.3.3 Cost of an algorithm . . . 8
1.4 Vector and matrix norms . . . 8
1.4.1 Vector norms . . . 8
1.4.2 Matrix norms . . . 10
1.5 Summary . . . 13

2 Linear systems: condition number, Gaussian elimination, LU and Cholesky 14
2.1 The problem . . . 14
2.2 Conditioning of linear systems . . . 14
2.2.1 Perturbation of the right-hand side . . . 14
2.2.2 Perturbation of the matrix . . . 15
2.2.3 Properties and examples . . . 16
2.3 Triangular systems . . . 17
2.4 Gaussian elimination and the LU decomposition . . . 18
2.4.1 The algorithm . . . 18
2.4.2 Solving $Ax = b$ with the LU decomposition . . . 20
2.4.3 Pivoting . . . 21
2.5 Symmetric positive definite matrices and the Cholesky decomposition . . . 23
2.6 Exploiting structure: banded and sparse matrices* . . . 25
2.7 Summary . . . 27

3 Least squares problems and the QR decomposition 28
3.1 Overdetermined systems and data fitting . . . 28
3.2 The normal equations and the geometry of least squares . . . 29
3.2.1 Solving the normal equations, and why one should be careful . . . 30
3.3 Orthogonal matrices and the QR decomposition . . . 32
3.4 Gram–Schmidt orthogonalisation* . . . 34
3.5 Householder reflections . . . 35
3.6 Application: polynomial regression and the choice of basis* . . . 38
3.7 Summary . . . 40

4 The singular value decomposition **41**
4.1 The theorem . . . 41
4.2 What the SVD tells us about a matrix . . . 42
4.3 The pseudoinverse and rank-deficient least squares problems . . . 43
4.4 Best low-rank approximation . . . 44
4.4.1 Experiment: image compression . . . 46
4.4.2 Principal component analysis (PCA)* . . . 47
4.5 Computing the SVD* . . . 47
4.6 Outlook: regularisation of ill-posed problems* . . . 48
4.7 Summary . . . 49

5 Eigenvalue problems **50**
5.1 Eigenvalues: recall and first observations . . . 50
5.2 Localisation: Gershgorin discs . . . 51
5.3 The symmetric case: Rayleigh quotient and conditioning . . . 53
5.4 The power method . . . 55
5.5 Inverse iteration, shifts and the Rayleigh quotient iteration . . . 57
5.6 Application: PageRank . . . 58
5.7 The QR algorithm* . . . 60
5.8 Summary . . . 61

6 Iterative methods for linear systems **63**
6.1 Why iterate? . . . 63
6.2 Fixed point iterations and the Banach fixed point theorem . . . 64
6.3 Linear iterations: splitting methods . . . 65
6.4 Convergence theorems . . . 67
6.5 Gradient descent and the conjugate gradient method . . . 70
6.6 Summary . . . 74

7 Nonlinear equations **76**
7.1 The problem . . . 76
7.2 Bisection . . . 77
7.3 Fixed-point iteration . . . 78
7.4 Newton's method . . . 79
7.4.1 Derivation and algorithm . . . 79
7.4.2 The secant method* . . . 82
7.5 Newton's method for systems . . . 83
7.5.1 Experiment: basins of attraction* . . . 84
7.6 Summary . . . 85

8 Polynomial interpolation and splines **86**
8.1 The interpolation problem . . . 86
8.2 Existence, uniqueness and the Lagrange form . . . 86
8.3 The Newton form and divided differences* . . . 87
8.4 The interpolation error . . . 89
8.5 The Runge phenomenon and Chebyshev nodes . . . 90
8.6 Piecewise polynomial interpolation and splines . . . 92
8.7 Summary . . . 95

9 Numerical integration **96**
9.1 Quadrature rules . . . 96
9.2 Composite rules . . . 99

9.3 Gaussian quadrature . . . 100
9.4 Extrapolation and adaptivity . . . 103
9.5 High dimensions: the curse of dimensionality and Monte Carlo integration . . . 103
9.6 Summary . . . 105

10 Numerical solution of ordinary differential equations 106
10.1 Initial value problems . . . 106
10.2 Euler's method . . . 107
10.3 Convergence of Euler's method . . . 108
10.4 Runge–Kutta methods . . . 110
10.5 Stiff problems and implicit methods . . . 112
10.6 A system: the pendulum* . . . 114
10.7 Outlook* . . . 115
10.8 Summary . . . 116

Preface

These notes cover the numerics part of *Basics in Applied Mathematics*.

Sections and results marked with an asterisk (*) are not presented in the lecture. They are included for self-study and for reference; they are not part of the testable core.

This material was prepared with the help of AI. The selection and structure of the material and every final formulation are mine, as is the responsibility for any remaining errors; please report them to me.

Literature

- S. Bartels, *Numerical Mathematics 3×9*, Springer 2025 – the reference text for this part.
- A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, Springer – broad and detailed.
- L. N. Trefethen, D. Bau, *Numerical Linear Algebra*, SIAM – Chapters 3–6; an elegant treatment of QR, SVD and conditioning.
- E. Süli, D. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press – very accessible, with full proofs.
- P. Deuflhard, A. Hohmann, *Numerical Analysis in Modern Scientific Computing*, Springer – condition and stability.
- N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM 2002 – the reference for rounding error analysis (Chapters 1–3).
- G. H. Golub, C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press 2013 – the reference for matrix algorithms (Chapters 2–6).
- E. Hairer, S. P. Nørsett, G. Wanner, *Solving Ordinary Differential Equations I*, and E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer – for Chapter 10.

1 Introduction, floating-point arithmetic, conditioning and norms

Learning goals. After this chapter you can

(i) name the four sources of error in a numerical computation,

(ii) explain how real numbers are stored in a computer, why $0.1+0.2 \neq 0.3$ in floating-point arithmetic, and what cancellation is,

(iii) distinguish a well-conditioned problem from a stable algorithm,

(iv) measure the size of vectors and matrices with norms and compute $\|A\|_1$, $\|A\|_\infty$ and $\|A\|_2$.

1.1 What this part of the course is about

Numerical mathematics is concerned with algorithms that compute approximate solutions of mathematical problems using finitely many arithmetic operations on numbers of finite precision. The problems treated in this part of the course are the following:

- solving a linear system $Ax = b$ (Chapters 2 and 6), also when there are more equations than unknowns (Chapter 3);
- computing eigenvalues and singular values of a matrix (Chapters 4 and 5);
- solving a nonlinear equation $f(x) = 0$ (Chapter 7);
- reconstructing a function from finitely many values (Chapter 8) and computing an integral (Chapter 9);
- solving an ordinary differential equation (Chapter 10).

For each of these problems three questions recur throughout the course.

1. **Accuracy.** How far is the computed result from the exact solution, and how does the error depend on the parameters of the method (step size, number of iterations, number of nodes)?
2. **Cost.** How many arithmetic operations and how much memory does the method require as a function of the problem size n?
3. **Robustness.** Does the method still work when the data are perturbed, for instance by measurement errors or by rounding in the computer?

Four sources of error. In a computation based on a mathematical model one distinguishes

- the **model error**: the equations do not describe reality exactly;
- the **data error**: the input data are known only approximately;
- the **method error** (discretisation error): the exact mathematical problem is replaced by a computable approximation, for instance a series by a finite sum, an integral by a finite

sum, a limit by an iterate;

- the **rounding error**: the computer stores numbers with finite precision.

Numerical analysis is concerned with the method error and the rounding error, and with the question of how an algorithm propagates data errors. The following example shows that two mathematically identical formulas can behave very differently in floating-point arithmetic.

Example 1.1 (two formulas for one function)**.** For $x > 0$ the two expressions

$$f(x) = \frac{1}{x} - \frac{1}{x+1} \qquad \text{and} \qquad g(x) = \frac{1}{x(x+1)}$$

define the same function. The table below lists the values of both expressions computed in double precision arithmetic for $x_k = 10^k$, $k = 1, \ldots, 16$, together with the relative difference $|f(x_k) - g(x_k)|/|g(x_k)|$.

k	x	f(x)	g(x)	rel. difference
1	1e+01	9.0909090909e-03	9.0909090909e-03	3.82e-16
2	1e+02	9.9009900990e-05	9.9009900990e-05	8.21e-16
3	1e+03	9.9900099900e-07	9.9900099900e-07	2.16e-14
4	1e+04	9.9990001000e-09	9.9990001000e-09	6.39e-14
5	1e+05	9.9999000010e-11	9.9999000010e-11	6.25e-13
6	1e+06	9.9999899989e-13	9.9999900000e-13	1.08e-10
7	1e+07	9.9999989907e-15	9.9999990000e-15	9.28e-10
8	1e+08	1.0000000003e-16	9.9999999000e-17	1.03e-08
9	1e+09	1.0000001492e-18	9.9999999900e-19	1.50e-07
10	1e+10	1.0000006145e-20	9.9999999990e-21	6.15e-07
11	1e+11	9.9999996827e-23	9.9999999999e-23	3.17e-08
12	1e+12	1.0000484359e-24	1.0000000000e-24	4.84e-05
13	1e+13	1.0009067165e-26	1.0000000000e-26	9.07e-04
14	1e+14	9.9396474058e-29	1.0000000000e-28	6.04e-03
15	1e+15	9.8607613153e-31	1.0000000000e-30	1.39e-02
16	1e+16	0.0000000000e+00	1.0000000000e-32	1.00e+00

For $x = 10^k$ the value computed from f agrees with the exact value in only about $16 - k$ decimal digits, and for $x = 10^{16}$ it is 0, because $x + 1$ is rounded to x. The value computed from g is correct to all displayed digits in every row. The explanation is the following. For $k \le 15$ the sum $x + 1$ is an integer below 2^{53} and is therefore computed exactly, and the two reciprocals $1/x \approx 10^{-k}$ and $1/(x+1) \approx 10^{-k}$ are each computed with a relative error of at most $u \approx 1.1 \cdot 10^{-16}$ (Theorem 1.4 below), that is, with an absolute error of at most about 10^{-16-k}. Their exact difference is $1/(x(x+1)) \approx 10^{-2k}$, so an absolute error of 10^{-16-k} amounts to a relative error of about 10^{k-16} of the result: up to k of the 16 significant digits are lost. This is a rule of thumb for the worst case; the actual error depends on the particular rounding errors of the two reciprocals and need not grow monotonically with k (compare the rows $k = 10$ and $k = 11$). Nothing is wrong with f mathematically; the loss of accuracy is caused by the subtraction of two nearly equal numbers in floating-point arithmetic, called *cancellation*. The precise analysis is given in Section 1.2.3.

1.2 Floating-point numbers

1.2.1 Representation

A computer stores a real number with finitely many digits. The standard representation is the floating-point format.

Definition 1.2 (floating-point numbers)**.** Let $\beta \ge 2$ (the *base*), $t \ge 1$ (the *precision*) and $e_{\min} \le e_{\max}$ be integers. The set of **floating-point numbers** is

$$\mathbb{F} = \mathbb{F}(\beta, t, e_{\min}, e_{\max}) = \Big\{ \pm \beta^e \sum_{i=1}^{t} d_i \, \beta^{-i} \;:\; d_i \in \{0, \ldots, \beta-1\},\ d_1 \neq 0,\ e_{\min} \le e \le e_{\max} \Big\} \cup \{0\}.$$

For $x = \pm\beta^e \sum_{i=1}^{t} d_i \beta^{-i} \in \mathbb{F} \setminus \{0\}$ the number $m = \sum_{i=1}^{t} d_i \beta^{-i} \in [\beta^{-1}, 1)$ is the *mantissa* and e the *exponent* of x.

The condition $d_1 \neq 0$ (*normalisation*) makes the representation of $x \in \mathbb{F} \setminus \{0\}$ unique: if $\beta^e m = \beta^{e'} m'$ with $m, m' \in [\beta^{-1}, 1)$, then $\beta^{e-e'} = m'/m \in (\beta^{-1}, \beta)$, hence $e = e'$ and $m = m'$; and two different digit strings (d_i), (d_i') with first difference at position i_0 give mantissas that differ by at least $\beta^{-i_0} - \sum_{i=i_0+1}^{t} (\beta - 1)\beta^{-i} = \beta^{-t} > 0$. The set $\mathbb{F}$ consists of the normalised numbers and 0; IEEE arithmetic additionally contains *subnormal* numbers ($d_1 = 0$ with $e = e_{\min}$), $\pm\infty$ and the symbol `NaN` ("not a number").

Current computers implement the IEEE 754 standard. Its **double precision** format has $\beta = 2$, $t = 53$, $e_{\min} = -1021$ and $e_{\max} = 1024$; a number is stored in 64 bits (one sign bit, eleven bits for the exponent and 52 bits for the mantissa, since the leading digit $d_1 = 1$ need not be stored). The single precision format (`float32`) has $t = 24$. The largest double is $(1 - 2^{-53})\, 2^{1024} \approx 1.80 \cdot 10^{308}$, the smallest positive normalised double is $2^{-1022} \approx 2.23 \cdot 10^{-308}$.

Two properties of $\mathbb{F}$ are essential.

- $\mathbb{F}$ is a **finite** set (there are $2 \cdot 2^{52} \cdot 2046 \approx 1.8 \cdot 10^{19}$ normalised doubles). In particular, $\mathbb{F}$ is not closed under the operations $+, -, \cdot, /$.
- The elements of $\mathbb{F}$ are **not equally spaced**: in the interval $[\beta^{e-1}, \beta^e)$ they form a grid with spacing β^{e-t}. For doubles, the interval $[1, 2)$ contains 2^{52} elements of $\mathbb{F}$ with spacing 2^{-52}, and in $[2^k, 2^{k+1})$ the spacing is 2^{k-52} for every exponent in the admissible range, $-1022 \leq k \leq 1023$. Relative to the numbers themselves the spacing is nearly constant: for $x \in [2^k, 2^{k+1}) \cap \mathbb{F}$ the quotient $2^{k-52}/x$ lies in $(2^{-53}, 2^{-52}]$.

```
spacing of doubles near    1e+00: 2.220e-16   (relative: 2.220e-16)
spacing of doubles near    1e+03: 1.137e-13   (relative: 1.137e-16)
spacing of doubles near    1e+06: 1.164e-10   (relative: 1.164e-16)
spacing of doubles near    1e+15: 1.250e-01   (relative: 1.250e-16)
0.1 + 0.2 == 0.3 ? False    0.1+0.2 = 0.30000000000000004
```

The last line of the output is not a defect of the software: 0.1 is not a double (its binary expansion is infinite and periodic), so the literal `0.1` already denotes the double nearest to 0.1, and the sum of two rounded numbers, rounded once more, need not coincide with the double nearest to 0.3.

1.2.2 Rounding and the standard model

Definition 1.3 (rounding, unit roundoff)**.** Let $x_{\max} := (1 - \beta^{-t})\, \beta^{e_{\max}}$ denote the largest element of $\mathbb{F}$. The **rounding** $\mathrm{fl}\colon [-x_{\max}, x_{\max}] \to \mathbb{F}$ maps x to an element of $\mathbb{F}$ nearest to x; if there are two nearest elements, one of them is selected by a fixed tie-breaking rule (in IEEE arithmetic the one whose last digit d_t is even). The number

$$u := \tfrac{1}{2}\beta^{1-t}$$

is called the **unit roundoff** (or machine precision). For doubles, $u = 2^{-53} \approx 1.11 \cdot 10^{-16}$.

Theorem 1.4 (rounding error)**.** *Let $x \in \mathbb{R} \setminus \{0\}$ satisfy*

$$\beta^{e_{\min}-1} \leq |x| \leq x_{\max}. \tag{1.1}$$

(The left bound is the smallest positive normalised number and the right bound the largest floating-point number; the two conditions are called no underflow *and* no overflow*.) Then*

$$\frac{|\,\mathrm{fl}(x) - x|}{|x|} \leq u.$$

Equivalently, there is a number $\delta = \delta(x) \in \mathbb{R}$ with $|\delta| \leq u$ such that $\mathrm{fl}(x) = x\,(1 + \delta)$.

Proof. Since $x \neq 0$, there is exactly one integer e with $\beta^{e-1} \le |x| < \beta^e$, namely $e = \lfloor \log_\beta |x| \rfloor + 1$. The hypothesis (1.1) gives $e_{\min} \le e \le e_{\max}$: from $\beta^e > |x| \ge \beta^{e_{\min}-1}$ we get $e > e_{\min} - 1$, and from $\beta^{e-1} \le |x| \le x_{\max} < \beta^{e_{\max}}$ we get $e - 1 < e_{\max}$.

The elements of $\mathbb{F}$ with exponent e are the numbers $\pm\beta^e m$ with $m \in \{\beta^{-1}, \beta^{-1} + \beta^{-t}, \dots, 1 - \beta^{-t}\}$; on the interval $[\beta^{e-1}, \beta^e)$ they form a grid with spacing β^{e-t} whose smallest point is β^{e-1} and whose largest point is $\beta^e(1 - \beta^{-t})$. If $e < e_{\max}$, then $\beta^e = \beta^{e+1} \cdot \beta^{-1}$ also belongs to $\mathbb{F}$, so $|x|$ lies between two consecutive elements of $\mathbb{F}$ at distance β^{e-t}; if $e = e_{\max}$, then $|x| \le x_{\max} = \beta^e(1 - \beta^{-t})$ is the largest grid point, and again $|x|$ lies between two consecutive grid points at distance β^{e-t} (or coincides with one). Since $\mathrm{fl}(x)$ is an element of $\mathbb{F}$ nearest to x, its distance to x is at most half the spacing:

$$|\,\mathrm{fl}(x) - x| \le \tfrac{1}{2}\beta^{e-t}.$$

Using $|x| \ge \beta^{e-1}$ we conclude

$$\frac{|\,\mathrm{fl}(x) - x|}{|x|} \le \frac{\frac{1}{2}\beta^{e-t}}{\beta^{e-1}} = \tfrac{1}{2}\beta^{1-t} = u.$$

The exponent e has cancelled: the bound is the same for every magnitude of x. For the second statement set $\delta := (\mathrm{fl}(x) - x)/x$; then $\mathrm{fl}(x) = x(1 + \delta)$ and $|\delta| \le u$ by the first statement. □

The IEEE standard requires the four basic operations to be *correctly rounded*: for $a, b \in \mathbb{F}$ and $\circ \in \{+, -, \cdot, /\}$ the computed result of $a \circ b$ is $\mathrm{fl}(a \circ b)$. Together with Theorem 1.4 this gives the **standard model of floating-point arithmetic**,

$$\mathrm{fl}(a \circ b) = (a \circ b)(1 + \delta), \qquad |\delta| \le u, \tag{1.2}$$

valid for all $a, b \in \mathbb{F}$ for which the exact result $a \circ b$ is nonzero and satisfies the no-underflow and no-overflow condition (1.1) (and $b \neq 0$ in the case of division); if $a \circ b = 0$ the result is exact. Every single operation is thus accurate to a relative error of at most u. The difficulties arise when many operations are combined, as the following output shows.

```
floating point addition is not associative: (1 + 1e-16) - 1 = 0.0 ,  1 + (1e-16 - 1) =
    1.1102230246251565e-16
```

The two results differ because floating-point addition is not associative. The spacing of the doubles in $[1, 2)$ is $2^{-52} \approx 2.2 \cdot 10^{-16}$, so $1 + 10^{-16}$ is rounded to 1 and the first expression evaluates to 0. In the second expression, $10^{-16} - 1 = -(1 - 10^{-16})$ is rounded to $-(1 - 2^{-53})$, because the spacing of the doubles in $[\frac{1}{2}, 1)$ is $2^{-53} \approx 1.1 \cdot 10^{-16}$ and $1 - 2^{-53}$ is the double nearest to $1 - 10^{-16}$; the second expression therefore evaluates to $2^{-53} \approx 1.11 \cdot 10^{-16}$.

1.2.3 Cancellation

Consider the subtraction of two numbers that are themselves results of earlier computations and therefore carry relative errors: $\tilde{a} = a(1 + \delta_a)$ and $\tilde{b} = b(1 + \delta_b)$ with $|\delta_a|, |\delta_b| \le u$ and $a \neq b$. Even if the subtraction itself is carried out exactly, the result $\tilde{a} - \tilde{b}$ has the relative error

$$\frac{|(\tilde{a} - \tilde{b}) - (a - b)|}{|a - b|} = \frac{|a\delta_a - b\delta_b|}{|a - b|} \le u\,\frac{|a| + |b|}{|a - b|}. \tag{1.3}$$

The factor $(|a| + |b|)/|a - b| \ge 1$ is the worst-case amplification of the relative errors of the operands by the subtraction (a particular pair of errors δ_a, δ_b may be amplified less, or may even cancel), and it is large precisely when $|a - b|$ is small compared with $|a| + |b|$, that is, when a and b are nearly equal relative to their size. This phenomenon is called **cancellation**: the leading digits of a and b agree and cancel, and the trailing digits, which carry the rounding

errors of $\tilde{a}$ and $\tilde{b}$, become the leading digits of the result. For $a = 1/x$ and $b = 1/(x+1)$ with $x = 10^k$ (Example 1.1) the amplification factor is $(|a| + |b|)/|a - b| = (2x + 1) \approx 2 \cdot 10^k$, in accordance with the loss of up to k digits observed there.

Example 1.5 (quadratic equation)**.** Let $p, q \in \mathbb{R}$ with $p \neq 0$ and $p^2/4 - q > 0$. The two roots of $x^2 + px + q = 0$ are

$$x_\pm = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}.$$

If $p^2 \gg |q|$, then $\sqrt{p^2/4 - q} \approx |p|/2$, and the root of smaller modulus is obtained as the difference of two nearly equal numbers (x_+ if $p > 0$, x_- if $p < 0$). The remedy is to compute the root of larger modulus,

$$x_1 = -\frac{p}{2} - \operatorname{sign}(p)\sqrt{\frac{p^2}{4} - q},$$

in which the two terms have the same sign, so that no cancellation occurs in this addition, and the second root from Vieta's formula $x_1 x_2 = q$, that is, $x_2 = q/x_1$. (The discriminant $p^2/4 - q$ itself suffers from cancellation when $p^2/4 \approx q$, that is, for nearly double roots; these are ill-conditioned, see Remark 7.2.) For $p = 10^8$ and $q = 1$ the two procedures give the following results.

```
direct formula: smaller root = -7.4505805969e-09   larger root = -1.0000000000e+08
stable        : smaller root = -1.0000000000e-08   larger root = -1.0000000000e+08
```

Expanding $\sqrt{p^2/4 - q} = \frac{p}{2}\sqrt{1 - 4q/p^2}$ in powers of $1/p$ gives the exact roots $x_1 = -10^8 + 10^{-8} + 10^{-24} + O(10^{-40})$ and $x_2 = -10^{-8} - 10^{-24} + O(10^{-40})$. The direct formula returns $-7.45 \cdot 10^{-9}$ for the root of smaller modulus, a relative error of about 25 %. The reason is that $\sqrt{p^2/4 - q}$ is a double of magnitude $5 \cdot 10^7$, where the spacing of the doubles is $2^{-27} \approx 7.45 \cdot 10^{-9}$; the subtraction $-p/2 + \sqrt{p^2/4 - q}$ can therefore only produce a multiple of this spacing. The stable procedure returns -10^{-8} correct to all displayed digits. Consequently, the subtraction of nearly equal quantities that are themselves affected by rounding errors should be avoided where possible: if a formula contains such a subtraction, an algebraically equivalent formula without it should be sought. (If both operands are exact floating-point numbers, the subtraction is as accurate as any other operation by the standard model; the danger lies in the amplification of errors already present in the operands.)

1.3 Conditioning of a problem and stability of an algorithm

1.3.1 Condition numbers

Before an algorithm is blamed for an inaccurate result, one should ask whether the *problem itself* is sensitive to perturbations of its data. This is a property of the problem and does not depend on any algorithm.

Definition 1.6 (condition numbers)**.** Let $f\colon \mathbb{R}^n \to \mathbb{R}^m$ describe a problem (input x, output $f(x)$) and let $\|\cdot\|$ denote norms on $\mathbb{R}^n$ and $\mathbb{R}^m$ (norms are introduced in Section 1.4; for $n = m = 1$ the norm is the absolute value). The **absolute** and the **relative condition number** of f at x are

$$\kappa_{\mathrm{abs}}(x) = \lim_{\varepsilon \to 0} \sup_{0 < \|\Delta x\| \le \varepsilon} \frac{\|f(x + \Delta x) - f(x)\|}{\|\Delta x\|},$$

$$\kappa_{\mathrm{rel}}(x) = \lim_{\varepsilon \to 0} \sup_{0 < \|\Delta x\| \le \varepsilon} \frac{\|f(x + \Delta x) - f(x)\|/\|f(x)\|}{\|\Delta x\|/\|x\|}$$

(for the relative condition number we assume $x \neq 0$ and $f(x) \neq 0$). The limits exist in $[0, \infty]$ because the suprema are nondecreasing functions of ε. A problem is called **well-conditioned** at x if $\kappa_{\mathrm{rel}}(x)$ is of moderate size (as a rule of thumb, at most of the order 10^2) and **ill-conditioned** if $\kappa_{\mathrm{rel}}(x)$ is large; where the boundary lies depends on the accuracy required.

Lemma 1.7 (condition numbers of differentiable functions)**.** *Let $f\colon \mathbb{R} \to \mathbb{R}$ be differentiable at x. Then $\kappa_{\mathrm{abs}}(x) = |f'(x)|$, and if $x \neq 0$ and $f(x) \neq 0$,*

$$\kappa_{\mathrm{rel}}(x) = \frac{|f'(x)|\,|x|}{|f(x)|}.$$

Proof. By the definition of the derivative, the difference quotient $D(\Delta x) := |f(x+\Delta x) - f(x)|/|\Delta x|$ converges to $|f'(x)|$ as $\Delta x \to 0$. Let $\eta > 0$. There is $\varepsilon_0 > 0$ such that $|D(\Delta x) - |f'(x)|| \leq \eta$ for all $0 < |\Delta x| \leq \varepsilon_0$, hence $\sup_{0<|\Delta x|\leq\varepsilon} D(\Delta x) \in [|f'(x)| - \eta, |f'(x)| + \eta]$ for all $\varepsilon \leq \varepsilon_0$. Since $\eta > 0$ was arbitrary, the limit of the suprema equals $|f'(x)|$, which is the first claim. The quotient in the definition of $\kappa_{\mathrm{rel}}(x)$ equals $D(\Delta x)\,|x|/|f(x)|$, and the second claim follows from the first. □

Example 1.8.

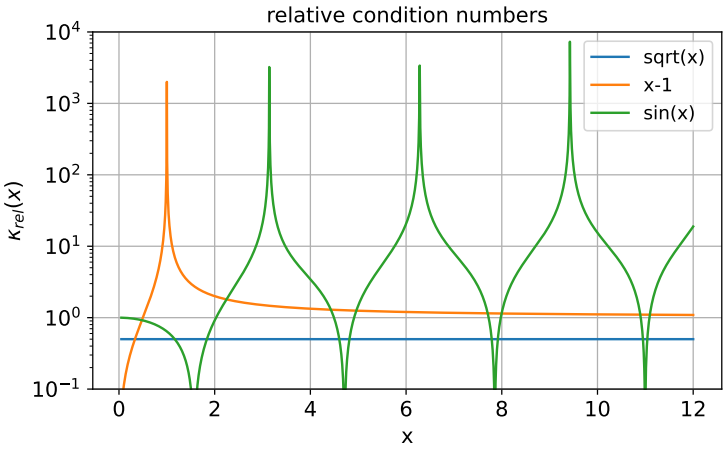
(i) $f(x) = \sqrt{x}$, $x > 0$: $\kappa_{\mathrm{rel}}(x) = \dfrac{x}{2\sqrt{x}\,\sqrt{x}} = \dfrac{1}{2}$. Taking square roots is well-conditioned for every $x > 0$.

(ii) $f(x) = x - 1$: $\kappa_{\mathrm{rel}}(x) = \dfrac{|x|}{|x-1|} \to \infty$ as $x \to 1$. This is cancellation, viewed as the ill-conditioning of a problem.

(iii) $f(x) = e^x$: $\kappa_{\mathrm{rel}}(x) = |x|$; the exponential function is ill-conditioned for large $|x|$.

(iv) $f(x) = \sin x$: $\kappa_{\mathrm{rel}}(x) = |x \cot x|$, which is large near the roots $x = k\pi$, $k \in \mathbb{Z} \setminus \{0\}$. A small relative change of x near 1000π changes $\sin x$ by a large amount relative to its small value.

Remark 1.9 (interpretation of the relative condition number)*.* $\kappa_{\mathrm{rel}}(x)$ is a worst-case amplification factor. If the input is known only up to a relative error ε—and a relative error of up to u arises already when an input that is not itself a floating-point number is stored in $\mathbb{F}$ (Theorem 1.4)—then the output can deviate from the exact value by a relative error of about $\kappa_{\mathrm{rel}}(x)\,\varepsilon$, and no algorithm that receives only the perturbed input can detect or compensate for this deviation. A particular perturbation may of course be harmless, and for some inputs the actual error is much smaller than the bound.



1.3.2 Stability of an algorithm

An **algorithm** for the problem f is a finite sequence of elementary operations that maps the input x to a computed result $\tilde{f}(x)$; in floating-point arithmetic each operation is subject to the standard model (1.2).

Definition 1.10 (forward and backward stability)**.** An algorithm $\tilde{f}$ for the problem f is called

(i) **forward stable** if there is a constant C of moderate size (independent of x and of u, but possibly depending on the dimensions n and m) such that

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq C\,\bigl(1 + \kappa_{\mathrm{rel}}(x)\bigr)\,u$$

for all admissible inputs x, that is, if the computed result is as accurate as the conditioning of the problem permits (the term 1 accounts for the rounding of the result itself, which is unavoidable because the result has to be stored in $\mathbb{F}$);

(ii) **backward stable** if there is a constant C of moderate size (again independent of x and of u, but possibly depending on n and m) such that for every admissible input x there is $\tilde{x}$ with

$$\tilde{f}(x) = f(\tilde{x}) \qquad \text{and} \qquad \frac{\|\tilde{x} - x\|}{\|x\|} \leq C\,u,$$

that is, if the computed result is the *exact* result for a slightly perturbed input.

Remark 1.11. Backward stability is the stronger property. Fix an input x with $\kappa_{\mathrm{rel}}(x) < \infty$. If $\tilde{f}(x) = f(\tilde{x})$ with $\|\tilde{x} - x\|/\|x\| \leq Cu$, then by the definition of the relative condition number,

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = \frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} \leq \bigl(\kappa_{\mathrm{rel}}(x) + o(1)\bigr)\frac{\|\tilde{x} - x\|}{\|x\|} \leq \bigl(\kappa_{\mathrm{rel}}(x) + o(1)\bigr)\,C\,u$$

as $u \to 0$; a backward stable algorithm therefore satisfies the forward error bound of Definition 1.10(i) at every fixed input up to terms of higher order in u (the $o(1)$ term depends on x, so this argument alone does not provide a constant that is uniform in x). The rule to keep in mind is

$$\text{error of the result} \lesssim \underbrace{\text{conditioning of the problem}}_{\text{cannot be influenced}} \times \underbrace{\text{backward error of the algorithm}}_{\text{choose a good algorithm}}.$$

Example 1.12. The problem "compute $f(x) = 1/x - 1/(x+1) = 1/(x(x+1))$ for $x > 0$" is well-conditioned: by Lemma 1.7, with $f'(x) = -(2x+1)/(x(x+1))^2$,

$$\kappa_{\mathrm{rel}}(x) = \frac{|x\,f'(x)|}{|f(x)|} = \frac{2x+1}{x+1} \leq 2.$$

The algorithm "evaluate $1/x$, evaluate $1/(x+1)$, subtract" is *unstable* for large x: by (1.3), the subtraction amplifies the relative errors of the two intermediate results by the factor $2x+1$, so the relative error of the result can be of the order $x\,u$, whereas the condition number is at most 2; Example 1.1 shows that errors of this order do occur. The algorithm "evaluate $x+1$, multiply by x, take the reciprocal" consists of three operations, each with a relative error of at most u by (1.2), and none of them amplifies the relative errors of its operands, since no subtraction of nearly equal numbers occurs: with $|\delta_1|, |\delta_2|, |\delta_3| \leq u$ the computed result is

$$\tilde{g}(x) = \frac{1+\delta_3}{x\,(x+1)\,(1+\delta_1)(1+\delta_2)} = g(x)\,\bigl(1 + \delta_3 - \delta_1 - \delta_2 + O(u^2)\bigr),$$

so its relative error is at most $3u + O(u^2)$ (provided that $x(x+1)$ does not overflow), and this algorithm is forward stable. This is precisely what Example 1.1 shows.

The recurring lesson of this course is that *the error of a computation is governed by the product of the conditioning of the problem and the backward error of the algorithm.* We will compute condition numbers for linear systems (Chapter 2), least squares problems (Chapter 3), eigenvalues (Chapter 5) and root finding (Chapter 7), and we will design algorithms that are backward stable.

1.3.3 Cost of an algorithm

The second quality criterion is cost. We count **floating-point operations (flops)**—additions, subtractions, multiplications, divisions—as a function of the problem size n, and we are interested in the asymptotic behaviour for large n.

Definition 1.13 (Landau symbols)**.** For sequences $(a_n)_{n\in\mathbb{N}}$, $(b_n)_{n\in\mathbb{N}}$ of positive numbers we write $a_n = O(b_n)$ if there are $C > 0$ and $n_0 \in \mathbb{N}$ with $a_n \le C\, b_n$ for all $n \ge n_0$, and $a_n = o(b_n)$ if $a_n/b_n \to 0$ as $n \to \infty$.

For example, the scalar product of two vectors in $\mathbb{R}^n$ costs n multiplications and $n-1$ additions, that is, $2n-1 = O(n)$ flops; computed in the standard way, entry by entry, a matrix–vector product Ax with $A \in \mathbb{R}^{n\times n}$ costs $n(2n-1) = O(n^2)$ flops and a matrix–matrix product AB with $A, B \in \mathbb{R}^{n\times n}$ costs $n^2(2n-1) = 2n^3 - n^2 = O(n^3)$ flops (algorithms with fewer flops exist for the matrix product, but they are not used in this course). In Chapter 2 we will see that solving a linear system $Ax = b$ with $A \in \mathbb{R}^{n\times n}$ by Gaussian elimination costs $\frac{2}{3}n^3 + O(n^2)$ flops.

1.4 Vector and matrix norms

To measure errors of vectors and matrices, and to define condition numbers for matrix problems, we need norms.

1.4.1 Vector norms

Definition 1.14 (norm)**.** A **norm** on $\mathbb{R}^n$ is a map $\|\cdot\| \colon \mathbb{R}^n \to [0,\infty)$ with the following three properties:

(i) $\|x\| = 0$ if and only if $x = 0$ (definiteness);
(ii) $\|\alpha x\| = |\alpha|\,\|x\|$ for all $\alpha \in \mathbb{R}$, $x \in \mathbb{R}^n$ (homogeneity);
(iii) $\|x + y\| \le \|x\| + \|y\|$ for all $x, y \in \mathbb{R}^n$ (triangle inequality).

The most important examples are the **p-norms**

$$\|x\|_p = \Big(\sum_{j=1}^{n} |x_j|^p\Big)^{1/p} \quad (1 \le p < \infty), \qquad \|x\|_\infty = \max_{1\le j\le n} |x_j|,$$

in particular the Euclidean norm $\|x\|_2 = \sqrt{x^\top x}$, the 1-norm (the sum of the absolute values of the components) and the maximum norm $\|x\|_\infty$. That $\|\cdot\|_p$ satisfies the triangle inequality for general $p \in (1,\infty)$ is Minkowski's inequality, which we do not need in this generality; for the three norms we use, the norm properties are elementary.

Proposition 1.15. *$\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ are norms on $\mathbb{R}^n$.*

Proof. Definiteness and homogeneity are immediate from the definitions. For the triangle inequality we use $|x_j + y_j| \le |x_j| + |y_j|$: summing over j gives $\|x+y\|_1 \le \|x\|_1 + \|y\|_1$, and taking the maximum over j gives $\|x+y\|_\infty \le \max_j |x_j| + \max_j |y_j| = \|x\|_\infty + \|y\|_\infty$. For the 2-norm we recall the Cauchy–Schwarz inequality $|x^\top y| \le \|x\|_2 \|y\|_2$ from linear algebra; it gives

$$\|x+y\|_2^2 = \|x\|_2^2 + 2x^\top y + \|y\|_2^2 \le \|x\|_2^2 + 2\|x\|_2\|y\|_2 + \|y\|_2^2 = \big(\|x\|_2 + \|y\|_2\big)^2. \qquad \square$$

Theorem 1.16 (equivalence of the 1-, 2- and ∞-norm)**.** *For all $x \in \mathbb{R}^n$,*

$$\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\,\|x\|_\infty, \qquad \|x\|_\infty \le \|x\|_1 \le n\,\|x\|_\infty, \qquad \|x\|_2 \le \|x\|_1 \le \sqrt{n}\,\|x\|_2.$$

All six inequalities are sharp: equality holds in the left-hand inequalities for $x = e_1$ and in the right-hand inequalities for $x = (1,\dots,1)^\top$.

Proof. Let $x \in \mathbb{R}^n$ and let j^* be an index with $|x_{j^*}| = \|x\|_\infty$. Then

$$\|x\|_\infty^2 = |x_{j^*}|^2 \le \sum_{j=1}^n |x_j|^2 = \|x\|_2^2 \le n \max_j |x_j|^2 = n\|x\|_\infty^2,$$

which gives the first pair of inequalities after taking square roots. Similarly,

$$\|x\|_\infty = |x_{j^*}| \le \sum_{j=1}^n |x_j| = \|x\|_1 \le n \max_j |x_j| = n\|x\|_\infty,$$

which is the second pair. For the third pair, all terms in the expansion of $(\sum_j |x_j|)^2 = \sum_j |x_j|^2 + \sum_{j\ne k} |x_j||x_k|$ are nonnegative, hence $\|x\|_2^2 = \sum_j |x_j|^2 \le (\sum_j |x_j|)^2 = \|x\|_1^2$; and the Cauchy–Schwarz inequality applied to the vectors $(1,\dots,1)^\top$ and $(|x_1|,\dots,|x_n|)^\top$ gives

$$\|x\|_1 = \sum_{j=1}^n 1 \cdot |x_j| \le \sqrt{n}\,\Big(\sum_{j=1}^n |x_j|^2\Big)^{1/2} = \sqrt{n}\,\|x\|_2.$$

The equality cases are verified by inserting $x = e_1$ and $x = (1,\dots,1)^\top$. $\square$

Theorem 1.17 (equivalence of norms)**.** *Let $\|\cdot\|_a$ and $\|\cdot\|_b$ be norms on $\mathbb{R}^n$. Then there are constants $0 < c \le C$ such that*

$$c\,\|x\|_a \le \|x\|_b \le C\,\|x\|_a \qquad \text{for all } x \in \mathbb{R}^n.$$

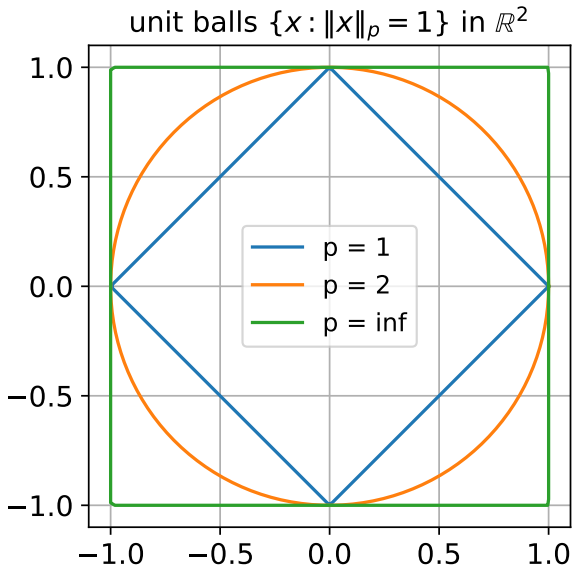
Proof *. It suffices to prove the statement for $\|\cdot\|_a = \|\cdot\|_\infty$ and an arbitrary norm $\|\cdot\|_b = \|\cdot\|$: if $c_1\|x\|_\infty \le \|x\|_a \le C_1\|x\|_\infty$ and $c_2\|x\|_\infty \le \|x\|_b \le C_2\|x\|_\infty$, then $(c_2/C_1)\|x\|_a \le \|x\|_b \le (C_2/c_1)\|x\|_a$.

Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Writing $x = \sum_j x_j e_j$ and using the triangle inequality and homogeneity, we obtain for all $x, y \in \mathbb{R}^n$

$$\big|\|x\| - \|y\|\big| \le \|x-y\| \le \sum_{j=1}^n |x_j - y_j|\,\|e_j\| \le M\|x-y\|_\infty, \qquad M := \sum_{j=1}^n \|e_j\|,$$

where the first inequality is the reverse triangle inequality, $\|x\| = \|(x-y)+y\| \le \|x-y\| + \|y\|$ and the same with x and y exchanged. Hence $\|\cdot\|$ is continuous with respect to $\|\cdot\|_\infty$, and the choice $y = 0$ gives $\|x\| \le M\|x\|_\infty$ for all x. The unit sphere $S = \{x \in \mathbb{R}^n : \|x\|_\infty = 1\}$ is closed and bounded, hence compact, so the continuous function $\|\cdot\|$ attains its minimum on S at some point $x_0 \in S$; since $x_0 \ne 0$, the minimum $c := \|x_0\|$ is positive. For $x \ne 0$ the vector $x/\|x\|_\infty$ lies in S, so $\|x\|/\|x\|_\infty = \|x/\|x\|_\infty\| \ge c$, that is, $c\|x\|_\infty \le \|x\|$; for $x = 0$ this is trivial. Together with $\|x\| \le M\|x\|_\infty$ this proves the claim with $C = M$. $\square$

Remark 1.18. By Theorem 1.17, whether a sequence of vectors converges does not depend on the norm used, and neither does the limit. The *constants* in error estimates do depend on the norm, however, and they may depend on the dimension n, as the factors $\sqrt{n}$ and n in Theorem 1.16 show. The figure below shows the unit balls $\{x \in \mathbb{R}^2 : \|x\|_p = 1\}$ for $p = 1, 2, \infty$.



1.4.2 Matrix norms

A matrix $A \in \mathbb{R}^{m\times n}$ can be regarded as a vector in $\mathbb{R}^{mn}$, which gives for instance the **Frobenius norm** $\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2\right)^{1/2}$. More useful in the analysis of algorithms are norms that measure how much a matrix can stretch a vector.

Definition 1.19 (induced matrix norm)**.** Let $\|\cdot\|$ denote a vector norm on $\mathbb{R}^n$ and a vector norm on $\mathbb{R}^m$. The **induced** (or **operator**) **norm** of $A \in \mathbb{R}^{m\times n}$ is

$$\|A\| = \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|.$$

The two expressions in the definition agree by homogeneity, $\|Ax\|/\|x\| = \|A(x/\|x\|)\|$, and the supremum over the unit sphere $\{x : \|x\| = 1\}$ is attained because the unit sphere is compact (it is closed and, by Theorem 1.17, bounded with respect to $\|\cdot\|_\infty$) and the map $x \mapsto \|Ax\|$ is continuous (indeed $|\|Ax\| - \|Ay\|| \le \|A(x-y)\| \le \sum_j |x_j - y_j| \|Ae_j\|$). Directly from the definition,

$$\|Ax\| \le \|A\|\,\|x\| \qquad \text{for all } x \in \mathbb{R}^n, \tag{1.4}$$

and $\|A\|$ is the smallest constant with this property.

Proposition 1.20 (properties of induced norms)**.** *Let $\|\cdot\|$ denote vector norms on $\mathbb{R}^n$, $\mathbb{R}^m$ and $\mathbb{R}^l$ and the induced matrix norms.*

(i) The induced norm is a norm on $\mathbb{R}^{m\times n}$.

(ii) $\|I\| = 1$ for the identity matrix $I \in \mathbb{R}^{n\times n}$.

*(iii) Induced norms are **submultiplicative**: $\|AB\| \le \|A\|\,\|B\|$ for all $A \in \mathbb{R}^{l\times m}$, $B \in \mathbb{R}^{m\times n}$.*

Proof. (i) If $\|A\| = 0$, then $\|Ax\| \le \|A\|\|x\| = 0$ for all x, so $Ax = 0$ for all x and $A = 0$; conversely $\|0\| = 0$. Homogeneity follows from $\|(\alpha A)x\| = |\alpha|\|Ax\|$. For the triangle inequality,

$\|(A+B)x\| \le \|Ax\| + \|Bx\| \le (\|A\| + \|B\|)\|x\|$ for all x, so the smallest constant $\|A+B\|$ in (1.4) satisfies $\|A+B\| \le \|A\| + \|B\|$. (ii) $\|Ix\| = \|x\|$ for all x. (iii) By (1.4), applied twice, $\|ABx\| \le \|A\|\,\|Bx\| \le \|A\|\,\|B\|\,\|x\|$ for all $x \in \mathbb{R}^n$, hence $\|AB\| \le \|A\|\|B\|$. □

Remark 1.21. The Frobenius norm is submultiplicative as well: by the Cauchy–Schwarz inequality, $|(AB)_{ij}|^2 = |\sum_k a_{ik}b_{kj}|^2 \le \big(\sum_k a_{ik}^2\big)\big(\sum_k b_{kj}^2\big)$, and summing over i and j gives $\|AB\|_F^2 \le \|A\|_F^2\|B\|_F^2$. It is not induced by any vector norm, because $\|I\|_F = \sqrt{n} \ne 1$ for $n \ge 2$.

Theorem 1.22 (explicit formulas for induced norms)**.** *For $A \in \mathbb{R}^{m\times n}$ the norms induced by the vector norms $\|\cdot\|_1$, $\|\cdot\|_\infty$ and $\|\cdot\|_2$ (on $\mathbb{R}^n$ and $\mathbb{R}^m$) are*

$$\|A\|_1 = \max_{k=1,\dots,n} \sum_{i=1}^{m} |a_{ik}| \quad (\textit{maximal absolute column sum}),$$

$$\|A\|_\infty = \max_{i=1,\dots,m} \sum_{k=1}^{n} |a_{ik}| \quad (\textit{maximal absolute row sum}),$$

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)},$$

where $\lambda_{\max}(A^\top A)$ denotes the largest eigenvalue of the symmetric positive semidefinite matrix $A^\top A$.

Proof. *∞-norm.* Let $\|x\|_\infty = 1$. For every row i,

$$|(Ax)_i| = \Big|\sum_{k=1}^{n} a_{ik}x_k\Big| \le \sum_{k=1}^{n} |a_{ik}|\,|x_k| \le \sum_{k=1}^{n} |a_{ik}|,$$

so $\|Ax\|_\infty \le \max_i \sum_k |a_{ik}|$ and hence $\|A\|_\infty \le \max_i \sum_k |a_{ik}|$. For the reverse inequality let i^* be a row with maximal absolute row sum and define x by $x_k = \operatorname{sign}(a_{i^*k})$ (with $\operatorname{sign}(0) = 1$). Then $\|x\|_\infty = 1$ and $(Ax)_{i^*} = \sum_k a_{i^*k}\operatorname{sign}(a_{i^*k}) = \sum_k |a_{i^*k}|$, so $\|Ax\|_\infty \ge \sum_k |a_{i^*k}|$, and equality holds.

1-norm. Let $\|x\|_1 = 1$. Then

$$\begin{aligned}\|Ax\|_1 = \sum_{i=1}^{m}\Big|\sum_{k=1}^{n} a_{ik}x_k\Big| &\le \sum_{i=1}^{m}\sum_{k=1}^{n} |a_{ik}||x_k| = \sum_{k=1}^{n} |x_k| \sum_{i=1}^{m} |a_{ik}| \\ &\le \Big(\max_k \sum_{i=1}^{m} |a_{ik}|\Big) \sum_{k=1}^{n} |x_k| = \max_k \sum_{i=1}^{m} |a_{ik}|.\end{aligned}$$

For $x = e_{k^*}$, the unit vector of a column k^* with maximal absolute column sum, we have $\|x\|_1 = 1$ and $\|Ax\|_1 = \sum_i |a_{ik^*}|$, so equality holds.

2-norm. The matrix $A^\top A \in \mathbb{R}^{n\times n}$ is symmetric, and $x^\top A^\top Ax = \|Ax\|_2^2 \ge 0$ for all x, so it is positive semidefinite. By the spectral theorem it has an orthonormal basis of eigenvectors $v_1, \dots, v_n$ with eigenvalues $\lambda_1 \ge \cdots \ge \lambda_n \ge 0$ (the eigenvalues are nonnegative because $\lambda_j = v_j^\top A^\top Av_j = \|Av_j\|_2^2$). For $x = \sum_j c_j v_j$ with $\|x\|_2^2 = \sum_j c_j^2 = 1$ we obtain

$$\|Ax\|_2^2 = x^\top A^\top Ax = \sum_{j=1}^{n} \lambda_j c_j^2 \le \lambda_1 \sum_{j=1}^{n} c_j^2 = \lambda_1,$$

with equality for $x = v_1$. Hence $\|A\|_2 = \sqrt{\lambda_1} = \sqrt{\lambda_{\max}(A^\top A)}$. □

The norms $\|A\|_1$ and $\|A\|_\infty$ cost $O(mn)$ flops and are used for cheap estimates; the computation of $\|A\|_2$ requires an eigenvalue computation (Chapter 5) or the singular value decomposition (Chapter 4).

For a square matrix $B \in \mathbb{R}^{n\times n}$ the quantity

$$\rho(B) = \max\{|\lambda| : \lambda \in \mathbb{C} \text{ is an eigenvalue of } B\}$$

is called the **spectral radius** of B. A matrix $Q \in \mathbb{R}^{n\times n}$ is called **orthogonal** if $Q^\top Q = I$ (see Chapter 3). The following properties of the 2-norm are used repeatedly.

Proposition 1.23 (properties of the 2-norm)**.** *Let $A \in \mathbb{R}^{m\times n}$.*

(i) $\|A^\top\|_2 = \|A\|_2$.

(ii) $\|QA\|_2 = \|A\|_2 = \|AP\|_2$ *for all orthogonal $Q \in \mathbb{R}^{m\times m}$ and $P \in \mathbb{R}^{n\times n}$.*

(iii) If $m = n$ and A is symmetric, then $\|A\|_2 = \rho(A)$. In particular, $\|A\|_2 = \sqrt{\rho(A^\top A)}$ for every $A \in \mathbb{R}^{m\times n}$.

*Proof *.* (i) By Theorem 1.22, $\|A\|_2^2 = \lambda_{\max}(A^\top A)$ and $\|A^\top\|_2^2 = \lambda_{\max}(AA^\top)$. If $A^\top Av = \lambda v$ with $v \neq 0$ and $\lambda \neq 0$, then $Av \neq 0$ and $AA^\top(Av) = A(A^\top Av) = \lambda\, Av$, so λ is an eigenvalue of $AA^\top$ as well; by symmetry of the argument, $A^\top A$ and $AA^\top$ have the same nonzero eigenvalues. If one of them has only the eigenvalue 0, it is the zero matrix (being symmetric), and then $A = 0$ (since $\|Ax\|_2^2 = x^\top A^\top Ax$, respectively $\|A^\top y\|_2^2 = y^\top AA^\top y$); in this case both norms vanish. Otherwise both largest eigenvalues are positive and coincide. (ii) For $y \in \mathbb{R}^m$ we have $\|Qy\|_2^2 = y^\top Q^\top Qy = \|y\|_2^2$, hence $\|QAx\|_2 = \|Ax\|_2$ for all x and $\|QA\|_2 = \|A\|_2$. Moreover $\|AP\|_2 = \|(AP)^\top\|_2 = \|P^\top A^\top\|_2 = \|A^\top\|_2 = \|A\|_2$ by (i) and the first part, since $P^\top$ is orthogonal as well ($PP^\top = P(P^\top P)P^{-1} = I$). (iii) Let $A = A^\top$ with orthonormal eigenbasis $v_1, \dots, v_n$ and eigenvalues $\mu_1, \dots, \mu_n \in \mathbb{R}$. Then $A^\top Av_j = A^2 v_j = \mu_j^2 v_j$, so the eigenvalues of $A^\top A$ are $\mu_1^2, \dots, \mu_n^2$ and $\|A\|_2 = \sqrt{\max_j \mu_j^2} = \max_j |\mu_j| = \rho(A)$. The last claim follows by applying this to the symmetric matrix $A^\top A$, whose eigenvalues are nonnegative: $\|A\|_2^2 = \lambda_{\max}(A^\top A) = \rho(A^\top A)$. □

Proposition 1.24. *For every induced norm $\|\cdot\|$ on $\mathbb{R}^{n\times n}$ and every $A \in \mathbb{R}^{n\times n}$ we have $\rho(A) \le \|A\|$.*

*Proof *.* Let $\lambda \in \mathbb{C}$ be an eigenvalue of A with eigenvector $v \in \mathbb{C}^n \setminus \{0\}$. If $\lambda = 0$ there is nothing to prove, so let $\lambda = |\lambda| e^{i\varphi} \neq 0$. Write $v = p + iq$ with $p, q \in \mathbb{R}^n$ and define the real vectors $w_\theta := \mathrm{Re}(e^{i\theta} v) = \cos\theta\, p - \sin\theta\, q$ for $\theta \in \mathbb{R}$. Since A is real, $Aw_\theta = \mathrm{Re}(e^{i\theta} Av) = \mathrm{Re}(e^{i\theta}\lambda v) = |\lambda|\, \mathrm{Re}(e^{i(\theta+\varphi)} v) = |\lambda|\, w_{\theta+\varphi}$. The function $\theta \mapsto \|w_\theta\|$ is continuous and 2π-periodic, so it attains its maximum M at some θ_0; moreover $M > 0$ because $w_0 = p$ and $w_{-\pi/2} = q$ do not both vanish. Put $w := w_{\theta_0 - \varphi}$. Then $Aw = |\lambda| w_{\theta_0}$, so $\|Aw\| = |\lambda| M > 0$, in particular $w \neq 0$, and $\|w\| \le M$. Hence

$$\|A\| \ge \frac{\|Aw\|}{\|w\|} \ge \frac{|\lambda| M}{M} = |\lambda|.$$

Since λ was an arbitrary eigenvalue, $\rho(A) \le \|A\|$. □

Example 1.25. For $A = \begin{pmatrix} 1 & 2 \\ -3 & 1 \end{pmatrix}$ the column sums of absolute values are 4 and 3 and the row sums are 3 and 4, so $\|A\|_1 = \|A\|_\infty = 4$ by Theorem 1.22. Furthermore $A^\top A = \begin{pmatrix} 10 & -1 \\ -1 & 5 \end{pmatrix}$ has the eigenvalues $(15 \pm \sqrt{29})/2$, so $\|A\|_2 = \sqrt{(15+\sqrt{29})/2} \approx 3.1926$, and $\|A\|_F = \sqrt{1+4+9+1} = \sqrt{15} \approx 3.8730$. The eigenvalues of A are $1 \pm i\sqrt{6}$, so $\rho(A) = \sqrt{7} \approx 2.6458$,

which is smaller than each of the three induced norms, in accordance with Proposition 1.24. The output below confirms these values.

```
||A||_1   = 4.0  (max column sum)
||A||_inf = 4.0  (max row sum)
||A||_2   = 3.192582403567252  = sqrt(lambda_max(A^T A)) = 3.192582403567252
||A||_F   = 3.872983346207417
spectral radius rho(A) = 2.6457513110645907  <= every induced norm
```

1.5 Summary

- Floating-point numbers form a finite, non-uniformly spaced set. Under rounding to nearest and in the absence of underflow and overflow, every basic operation is accurate to a relative error of at most $u = 2^{-53} \approx 1.1 \cdot 10^{-16}$ (standard model), but the subtraction of nearly equal numbers amplifies the relative errors already present in the operands by up to the factor $(|a| + |b|)/|a - b|$ (cancellation), even when the subtraction itself is exact, and can destroy all accuracy.
- The **condition number** measures the worst-case sensitivity of a *problem* to perturbations of its data; for an ill-conditioned problem no algorithm can guarantee an accurate result from uncertain data. **Stability** is a property of an *algorithm*: a backward stable algorithm returns the exact result for slightly perturbed data and therefore does not introduce an error amplification beyond the conditioning of the problem.
- The **cost** of an algorithm is measured in flops; the asymptotic order $O(n^p)$ determines which problem sizes are feasible, although constants and memory traffic also matter in practice.
- **Norms** measure the size of vectors and matrices; all norms on $\mathbb{R}^n$ are equivalent, but the constants may depend on n. Induced matrix norms satisfy $\|Ax\| \le \|A\|\,\|x\|$, $\|AB\| \le \|A\|\,\|B\|$ and $\rho(A) \le \|A\|$. The formulas $\|A\|_1 =$ maximal absolute column sum, $\|A\|_\infty =$ maximal absolute row sum and $\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)}$ are used throughout the course.

2 Linear systems: condition number, Gaussian elimination, LU and Cholesky

Learning goals. After this chapter you can

(i) bound the effect of data errors on the solution of $Ax = b$ by the condition number $\operatorname{cond}(A)$,

(ii) describe Gaussian elimination as the factorisation $A = LU$ and derive its cost of about $\frac{2}{3}n^3$ flops,

(iii) explain why pivoting may be necessary,

(iv) use the Cholesky factorisation for symmetric positive definite matrices,

(v) exploit band structure.

2.1 The problem

Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, find $x \in \mathbb{R}^n$ with $Ax = b$. Linear systems are the workhorse of scientific computing: discretised differential equations, least squares fits, Newton's method for nonlinear systems (Chapter 7) and implicit time stepping for differential equations (Chapter 10) all lead to linear systems, often with large n.

We recall from linear algebra that for $A \in \mathbb{R}^{n\times n}$ the following statements are equivalent:

(i) $Ax = b$ has a unique solution for every $b \in \mathbb{R}^n$;

(ii) $\ker A = \{0\}$;

(iii) $\operatorname{rank} A = n$, that is, the columns of A are linearly independent;

(iv) $\det A \neq 0$;

(v) 0 is not an eigenvalue of A.

A matrix with these properties is called *regular* (also: nonsingular or invertible).

2.2 Conditioning of linear systems

2.2.1 Perturbation of the right-hand side

Suppose the data b are perturbed to $b + \Delta b$ (by measurement errors or by rounding). The solution changes from x to $x + \Delta x$, where $A\Delta x = \Delta b$. We ask how large Δx is relative to x. Throughout this section $\|\cdot\|$ denotes a vector norm on $\mathbb{R}^n$ and the matrix norm it induces.

Definition 2.1 (condition number of a matrix)**.** For a regular matrix $A \in \mathbb{R}^{n\times n}$ and an induced matrix norm $\|\cdot\|$ the **condition number** of A is

$$\operatorname{cond}(A) = \|A\|\,\|A^{-1}\|.$$

We write $\operatorname{cond}_p(A)$ when the norm induced by the p-norm is used.

Theorem 2.2 (perturbation of the right-hand side)**.** *Let $A \in \mathbb{R}^{n\times n}$ be regular, let $Ax = b \neq 0$ and $A(x + \Delta x) = b + \Delta b$. Then*

$$\frac{\|\Delta x\|}{\|x\|} \leq \operatorname{cond}(A)\,\frac{\|\Delta b\|}{\|b\|}.$$

The bound is sharp: for every regular A there are $b \neq 0$ and $\Delta b \neq 0$ for which equality holds.

Proof. From $\Delta x = A^{-1}\Delta b$ and (1.4) we get $\|\Delta x\| \leq \|A^{-1}\|\,\|\Delta b\|$, and from $b = Ax$ we get $\|b\| \leq \|A\|\,\|x\|$, that is, $1/\|x\| \leq \|A\|/\|b\|$ (note that $x \neq 0$ because $b \neq 0$). Multiplying the two inequalities gives

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A^{-1}\|\,\|\Delta b\|\,\frac{\|A\|}{\|b\|} = \operatorname{cond}(A)\frac{\|\Delta b\|}{\|b\|}.$$

For the sharpness recall from Definition 1.19 that the maximum in the definition of an induced norm is attained. Choose $x \neq 0$ with $\|Ax\| = \|A\|\,\|x\|$ and $\Delta b \neq 0$ with $\|A^{-1}\Delta b\| = \|A^{-1}\|\,\|\Delta b\|$, and set $b := Ax$ and $\Delta x := A^{-1}\Delta b$. Then $b \neq 0$ since A is regular, and

$$\frac{\|\Delta x\|}{\|x\|} = \frac{\|A^{-1}\|\,\|\Delta b\|}{\|x\|} = \frac{\|A^{-1}\|\,\|\Delta b\|\,\|A\|}{\|b\|} = \operatorname{cond}(A)\frac{\|\Delta b\|}{\|b\|}. \qquad \square$$

Remark 2.3.

(i) For fixed A consider the problem $f(b) = A^{-1}b$ in the sense of Definition 1.6. Since f is linear, $\|f(b + \Delta b) - f(b)\| = \|A^{-1}\Delta b\|$, and the relative condition number at b is

$$\kappa_{\mathrm{rel}}(b) = \sup_{\Delta b\neq 0} \frac{\|A^{-1}\Delta b\|}{\|\Delta b\|}\cdot\frac{\|b\|}{\|A^{-1}b\|} = \|A^{-1}\|\,\frac{\|b\|}{\|A^{-1}b\|} \leq \|A^{-1}\|\,\|A\| = \operatorname{cond}(A),$$

with equality for the right-hand side $b = Ax$ constructed in the proof of Theorem 2.2. Thus $\operatorname{cond}(A)$ is the worst case over all right-hand sides of the relative condition number of the problem $b \mapsto A^{-1}b$.

(ii) By Proposition 1.20, $\operatorname{cond}(A) = \|A\|\,\|A^{-1}\| \geq \|AA^{-1}\| = \|I\| = 1$, and $\operatorname{cond}(\alpha A) = \operatorname{cond}(A)$ for all $\alpha \neq 0$. The condition number is therefore not related to the size of the determinant: $A = \alpha I$ has $\det A = \alpha^n$, which can be arbitrarily small or large, but $\operatorname{cond}(A) = 1$.

2.2.2 Perturbation of the matrix

Lemma 2.4 (perturbation lemma)**.** *Let $B \in \mathbb{R}^{n\times n}$ with $\|B\| < 1$ for an induced matrix norm. Then $I + B$ is regular and*

$$\|(I + B)^{-1}\| \leq \frac{1}{1 - \|B\|}.$$

Proof. If $(I + B)v = 0$, then $v = -Bv$ and $\|v\| \leq \|B\|\,\|v\|$, so $(1 - \|B\|)\|v\| \leq 0$ and $v = 0$; hence $\ker(I + B) = \{0\}$ and $I + B$ is regular. For $w \in \mathbb{R}^n$ let $v := (I + B)^{-1}w$. Then $v = w - Bv$, hence $\|v\| \leq \|w\| + \|B\|\,\|v\|$ and $\|v\| \leq \|w\|/(1 - \|B\|)$. Since w was arbitrary, $\|(I + B)^{-1}\| \leq 1/(1 - \|B\|)$. $\square$

Theorem 2.5 (perturbation of matrix and right-hand side)**.** *Let $A \in \mathbb{R}^{n\times n}$ be regular, let $\Delta A \in \mathbb{R}^{n\times n}$ satisfy $\|A^{-1}\|\,\|\Delta A\| < 1$, and let $Ax = b \neq 0$. Then $A + \Delta A$ is regular, and for every $\Delta b \in \mathbb{R}^n$ the solution $x + \Delta x$ of $(A + \Delta A)(x + \Delta x) = b + \Delta b$ satisfies*

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\operatorname{cond}(A)}{1 - \operatorname{cond}(A)\,\dfrac{\|\Delta A\|}{\|A\|}} \Big(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \Big).$$

Proof. Let $B := A^{-1}\Delta A$. Then $\|B\| \le \|A^{-1}\|\,\|\Delta A\| < 1$, so by Lemma 2.4 the matrix $I + B$ is regular with $\|(I + B)^{-1}\| \le 1/(1 - \|B\|)$, and $A + \Delta A = A(I + B)$ is regular as a product of regular matrices. Subtracting $Ax = b$ from $(A + \Delta A)(x + \Delta x) = b + \Delta b$ gives

$$(A + \Delta A)\Delta x = \Delta b - \Delta A\, x, \qquad \text{that is,} \qquad \Delta x = (I + B)^{-1} A^{-1} \big(\Delta b - \Delta A\, x\big).$$

Taking norms and using $\|B\| \le \|A^{-1}\|\|\Delta A\|$,

$$\|\Delta x\| \le \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\,\|\Delta A\|} \big(\|\Delta b\| + \|\Delta A\|\,\|x\|\big).$$

We divide by $\|x\|$ and use $1/\|x\| \le \|A\|/\|b\|$ (from $b = Ax$) in the first term:

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\,\|\Delta A\|} \Big(\|A\| \frac{\|\Delta b\|}{\|b\|} + \|\Delta A\| \Big) = \frac{\|A^{-1}\|\,\|A\|}{1 - \|A^{-1}\|\,\|A\|\,\frac{\|\Delta A\|}{\|A\|}} \Big(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \Big),$$

which is the claim since $\|A^{-1}\|\,\|A\| = \operatorname{cond}(A)$. □

If the relative errors $\|\Delta A\|/\|A\|$ and $\|\Delta b\|/\|b\|$ are at most ε with $\operatorname{cond}(A)\varepsilon \ll 1$, the theorem gives a relative error of at most about $2\operatorname{cond}(A)\varepsilon$ in x. In particular, since storing A and b in floating-point arithmetic already produces relative errors of the order u, no algorithm can guarantee a relative error smaller than about $\operatorname{cond}(A)u$ for the solution of $Ax = b$.

2.2.3 Properties and examples

Proposition 2.6 (properties of the condition number)**.**

(i) If A is symmetric and regular with eigenvalues $\lambda_1, \dots, \lambda_n$, then $\operatorname{cond}_2(A) = \dfrac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$.

(ii) If Q is orthogonal, then $\operatorname{cond}_2(Q) = 1$.

*(iii) (*Residual and error.*) Let A be regular, $Ax = b \neq 0$, and let $\tilde{x} \in \mathbb{R}^n$ be arbitrary with residual $r := b - A\tilde{x}$. Then*

$$\frac{1}{\operatorname{cond}(A)}\,\frac{\|r\|}{\|b\|} \le \frac{\|x - \tilde{x}\|}{\|x\|} \le \operatorname{cond}(A)\,\frac{\|r\|}{\|b\|}.$$

Proof. (i) By Proposition 1.23(iii), $\|A\|_2 = \rho(A) = \max_i |\lambda_i|$. The matrix A^{-1} is symmetric with eigenvalues $1/\lambda_i$ (from $Av_i = \lambda_i v_i$ we get $A^{-1}v_i = \lambda_i^{-1}v_i$), so $\|A^{-1}\|_2 = \max_i |\lambda_i|^{-1} = 1/\min_i |\lambda_i|$. (ii) $\|Qx\|_2 = \|x\|_2$ for all x gives $\|Q\|_2 = 1$, and $Q^{-1} = Q^{\top}$ is orthogonal as well, so $\|Q^{-1}\|_2 = 1$. (iii) We have $r = b - A\tilde{x} = A(x - \tilde{x})$, hence $x - \tilde{x} = A^{-1}r$ and $\|x - \tilde{x}\| \le \|A^{-1}\|\,\|r\|$; together with $1/\|x\| \le \|A\|/\|b\|$ this gives the upper bound. For the lower bound, $\|r\| = \|A(x - \tilde{x})\| \le \|A\|\,\|x - \tilde{x}\|$ and $\|x\| = \|A^{-1}b\| \le \|A^{-1}\|\,\|b\|$, so $\|r\|/\|b\| \le \|A\|\,\|A^{-1}\|\,\|x - \tilde{x}\|/\|x\|$. □

Part (iii) is important in practice: for a computed approximation $\tilde{x}$ the residual $r = b - A\tilde{x}$ can be computed from the data (in floating-point arithmetic up to rounding errors of the order $u\,(\|A\|\,\|\tilde{x}\| + \|b\|)$), whereas the error $x - \tilde{x}$ cannot be evaluated without knowing the exact solution. A small relative residual guarantees a small relative error only if $\operatorname{cond}(A)$ is small. Part (ii) is the reason why orthogonal transformations are used wherever possible (Chapter 3). For every regular matrix A, $\operatorname{cond}_2(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$ is the ratio of the largest and the smallest singular value of A (Chapter 4).

Example 2.7* (Hilbert matrix)**.** The Hilbert matrix $H_n = \big(1/(i+j-1)\big)_{i,j=1}^{n}$ has the entries $\int_0^1 t^{i-1}t^{j-1}\,\mathrm{d}t = 1/(i+j-1)$; it is the matrix of the normal equations (Chapter 3) for the least squares approximation of a function in $L^2(0,1)$ by polynomials in the monomial basis $1, t, \ldots, t^{n-1}$. It is symmetric positive definite, because $x^\top H_n x = \int_0^1 \big(\sum_{i=1}^n x_i t^{i-1}\big)^2\,\mathrm{d}t > 0$ for $x \neq 0$ (a nonzero polynomial has only finitely many roots), but extremely ill-conditioned: $\operatorname{cond}_2(H_n)$ grows exponentially in n, like $(1+\sqrt{2})^{4n}/\sqrt{n}$ with $(1+\sqrt{2})^4 \approx 34$ up to a constant factor (J. Todd, 1954; see N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd edition, Section 28.1); for instance $\operatorname{cond}_2(H_{10}) \approx 1.6 \cdot 10^{13}$, $\operatorname{cond}_2(H_{12}) \approx 1.7 \cdot 10^{16}$ and $\operatorname{cond}_2(H_{14}) \approx 1.9 \cdot 10^{19}$. The output below lists the condition number computed in double precision and the relative error of the solution of $H_n x = H_n \mathbf{1}$ computed by Gaussian elimination with partial pivoting (`numpy.linalg.solve`); the exact solution of the mathematical system is $x = \mathbf{1} = (1, \ldots, 1)^\top$. In the experiment the entries of H_n and the right-hand side $H_n\mathbf{1}$ are rounded to doubles, so the measured error contains the effect of these data perturbations—which is what Theorem 2.5 bounds—together with the rounding errors of the elimination. The error grows roughly like $\operatorname{cond}_2(H_n)\,u$, in accordance with the bound of Theorem 2.5; for $n \geq 12$ the hypothesis of that theorem fails for perturbations of size u, because $\operatorname{cond}_2(H_n)\,u \gtrsim 1$, and indeed the relative error is 13 % for $n = 12$ and exceeds 100 % for $n = 14$, so the computed solution no longer carries information. (The exact numbers depend on the hardware and the LAPACK version. For $n = 14$ the condition number computed in double precision, $3.2 \cdot 10^{17}$, is itself far too small, since the smallest singular value lies below the rounding level of the matrix entries; the value $1.9 \cdot 10^{19}$ above was computed in 80-digit arithmetic.)

```
 n   cond_2(H_n)     rel. error of numpy's solve for H_n x = H_n * ones
 4     1.55e+04       3.37e-14
 6     1.50e+07       4.08e-11
 8     1.53e+10       3.71e-08
10     1.60e+13       2.27e-05
12     1.64e+16       1.29e-01
14     3.22e+17       3.02e+00
```

2.3 Triangular systems

A matrix $U = (u_{ij}) \in \mathbb{R}^{n\times n}$ is called **upper triangular** if $u_{ij} = 0$ for $i > j$, and $L = (l_{ij})$ is called **lower triangular** if $l_{ij} = 0$ for $i < j$; a lower (upper) triangular matrix with all diagonal entries equal to 1 is called *unit* lower (upper) triangular.

Lemma 2.8 (triangular matrices)**.** *Let $L, M \in \mathbb{R}^{n\times n}$ be lower triangular.*

(i) $\det L = \prod_{i=1}^n l_{ii}$*; in particular L is regular if and only if all diagonal entries are nonzero.*

(ii) LM is lower triangular with diagonal entries $l_{ii}m_{ii}$.

(iii) If L is regular, then L^{-1} is lower triangular with diagonal entries $1/l_{ii}$.

In particular, products and inverses of unit lower triangular matrices are unit lower triangular. The same statements hold for upper triangular matrices (apply the lemma to the transposed matrices).

Proof *. (i) Expanding $\det L$ along the first row, in which only l_{11} can be nonzero, gives $\det L = l_{11} \det L'$ with the lower triangular matrix L' obtained by deleting the first row and column; the claim follows by induction on n. (ii) We have $(LM)_{ij} = \sum_{k=1}^{n} l_{ik} m_{kj}$. A term can be nonzero only if $k \le i$ and $j \le k$, hence only if $j \le i$. For $i < j$ all terms vanish, and for $i = j$ only the term $k = i$ remains, which gives $(LM)_{ii} = l_{ii} m_{ii}$. (iii) Let x be the j-th column of L^{-1}, that is, $Lx = e_j$. The i-th row of this system reads $\sum_{k \le i} l_{ik} x_k = \delta_{ij}$. For $i < j$ we obtain by induction on i that $x_i = \frac{1}{l_{ii}}\big(0 - \sum_{k<i} l_{ik} x_k\big) = 0$, and for $i = j$ that $x_j = \frac{1}{l_{jj}}\big(1 - \sum_{k<j} l_{jk} x_k\big) = \frac{1}{l_{jj}}$. Thus the entries of L^{-1} above the diagonal vanish and the diagonal entries are $1/l_{jj}$. □

Triangular systems are solved by **substitution**: the i-th equation of $Ux = b$ reads $\sum_{j \ge i} u_{ij} x_j = b_i$ and contains only the unknowns $x_i, \dots, x_n$, so the unknowns can be computed in the order $x_n, x_{n-1}, \dots, x_1$; similarly for $Ly = b$ in the order $y_1, \dots, y_n$.

Algorithm 2.9 (backward and forward substitution)**.** Let U be upper triangular and L lower triangular, both regular.

$$Ux = b: \qquad x_i = \frac{1}{u_{ii}}\Big(b_i - \sum_{j=i+1}^{n} u_{ij} x_j\Big), \qquad i = n, n-1, \dots, 1 \qquad \text{(backward substitution)},$$

$$Ly = b: \qquad y_i = \frac{1}{l_{ii}}\Big(b_i - \sum_{j=1}^{i-1} l_{ij} y_j\Big), \qquad i = 1, 2, \dots, n \qquad \text{(forward substitution)}.$$

Computing x_i requires $n - i$ multiplications, $n - i$ subtractions and one division, so the total cost of a substitution is $\sum_{i=1}^{n} \big(2(n-i)+1\big) = n(n-1) + n = n^2$ flops.

2.4 Gaussian elimination and the LU decomposition

2.4.1 The algorithm

The idea of Gaussian elimination is to transform $Ax = b$ into an equivalent triangular system by subtracting multiples of one equation from the others.

Algorithm 2.10 (Gaussian elimination without pivoting)**.** Set $A^{(1)} := A$ and $b^{(1)} := b$. For $k = 1, \dots, n-1$: if the **pivot** $a_{kk}^{(k)}$ vanishes, stop; otherwise compute the **multipliers**

$$l_{ik} := \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \qquad i = k+1, \dots, n,$$

and subtract l_{ik} times row k from row i for $i = k+1, \dots, n$:

$$a_{ij}^{(k+1)} := a_{ij}^{(k)} - l_{ik}\, a_{kj}^{(k)} \quad (j = 1, \dots, n), \qquad b_i^{(k+1)} := b_i^{(k)} - l_{ik}\, b_k^{(k)}, \tag{2.1}$$

while the rows $1, \dots, k$ remain unchanged. After $n-1$ steps, $U := A^{(n)}$ is upper triangular. If $u_{nn} \neq 0$, the system $Ux = b^{(n)}$ is solved by backward substitution; if $u_{nn} = 0$, then A is singular ($\det A = \det U = 0$ by Lemma 2.11 and Lemma 2.8(i)) and $Ax = b$ has no unique solution.

In step k the entries in column k below the diagonal become zero, $a_{ik}^{(k+1)} = a_{ik}^{(k)} - l_{ik} a_{kk}^{(k)} = 0$ for $i > k$, and the entries in the columns $j < k$, which are already zero in the rows $i > k$, remain zero; apart from the entries below the pivot in column k, which become zero, the update (2.1) therefore only changes the trailing block $i, j > k$. Each step consists of elementary row operations, which do not change the solution set, so $Ux = b^{(n)}$ has the same solution as $Ax = b$.

Lemma 2.11 (elimination steps as matrix multiplications)**.** *Suppose that Algorithm 2.10 does not break down, that is, the pivots $a_{kk}^{(k)}$, $k = 1, \dots, n-1$, are nonzero, and let*

$$\ell_k := (0, \dots, 0, l_{k+1,k}, \dots, l_{nk})^\top \in \mathbb{R}^n, \qquad E_k := I - \ell_k e_k^\top, \qquad k = 1, \dots, n-1.$$

(i) $A^{(k+1)} = E_k A^{(k)}$ *and* $b^{(k+1)} = E_k b^{(k)}$ *for* $k = 1, \dots, n-1$*; hence* $U = E_{n-1} \cdots E_1 A$.
(ii) $E_k^{-1} = I + \ell_k e_k^\top$.
(iii) $E_1^{-1} E_2^{-1} \cdots E_{n-1}^{-1} = I + \sum_{k=1}^{n-1} \ell_k e_k^\top =: L$, *where*

$$L = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{pmatrix}$$

is the unit lower triangular matrix whose entry (i,k) below the diagonal is the multiplier l_{ik} (the matrices E_k are called elimination matrices*).*

Consequently $A = LU$.

Proof *. (i) The matrix $\ell_k e_k^\top$ has the rows $l_{ik}\, e_k^\top$ for $i > k$ and zero rows for $i \le k$, so the i-th row of $\ell_k e_k^\top A^{(k)}$ is l_{ik} times the k-th row of $A^{(k)}$ for $i > k$ and zero for $i \le k$. Hence $E_k A^{(k)} = A^{(k)} - \ell_k e_k^\top A^{(k)}$ is exactly the result of the row operations (2.1); the same argument applies to $b^{(k)}$. (ii) Since the k-th component of ℓ_k is zero, $e_k^\top \ell_k = 0$, and therefore $(I - \ell_k e_k^\top)(I + \ell_k e_k^\top) = I - \ell_k (e_k^\top \ell_k) e_k^\top = I$. (iii) We show by induction on j that $E_1^{-1} \cdots E_j^{-1} = I + \sum_{k \le j} \ell_k e_k^\top$. For $j = 1$ this is (ii). If it holds for $j-1$, then

$$\Big(I + \sum_{k<j} \ell_k e_k^\top\Big)\big(I + \ell_j e_j^\top\big) = I + \sum_{k<j} \ell_k e_k^\top + \ell_j e_j^\top + \sum_{k<j} \ell_k \,(e_k^\top \ell_j)\, e_j^\top = I + \sum_{k \le j} \ell_k e_k^\top,$$

because the k-th component of ℓ_j vanishes for $k \le j$, so $e_k^\top \ell_j = 0$ for $k < j$. The entry (i,k) of $\ell_k e_k^\top$ is l_{ik} for $i > k$ and zero otherwise, which gives the stated form of L. Finally, $A = (E_{n-1} \cdots E_1)^{-1} U = E_1^{-1} \cdots E_{n-1}^{-1} U = LU$ by (i). □

Definition 2.12 (LU decomposition)**.** A factorisation $A = LU$ with a unit lower triangular matrix L and an upper triangular matrix U is called an **LU decomposition** of A.

Theorem 2.13 (existence and uniqueness of the LU decomposition)**.** *A regular matrix $A \in \mathbb{R}^{n \times n}$ has an LU decomposition if and only if all leading principal submatrices $A_k := (a_{ij})_{i,j=1}^{k}$, $k = 1, \dots, n-1$, are regular. In this case the LU decomposition is unique, U is regular, and L and U are the matrices produced by Algorithm 2.10, which does not break down.*

Proof *. For a matrix $M \in \mathbb{R}^{n \times n}$ we write M_k for its leading principal $k \times k$ submatrix. We first note that for a lower triangular M and arbitrary $B \in \mathbb{R}^{n \times n}$

$$(MB)_k = M_k B_k, \tag{2.2}$$

because $(MB)_{ij} = \sum_{l \le i} m_{il} b_{lj}$ involves for $i, j \le k$ only entries of M_k and B_k.

Necessity. Let $A = LU$ be an LU decomposition. By (2.2) and the analogous identity for upper triangular factors, $A_k = L_k U_k$ for all k, hence $\det A_k = \det L_k \det U_k = u_{11} \cdots u_{kk}$ by Lemma 2.8(i). Since $\det A = u_{11} \cdots u_{nn} \neq 0$, all u_{kk} are nonzero, so $\det A_k \neq 0$ for all k; in particular U is regular.

Sufficiency. Let $A_1, \dots, A_{n-1}$ be regular and set $\det A_0 := 1$. We show by induction on k that the first $k-1$ steps of Algorithm 2.10 can be carried out and that the k-th pivot satisfies

$$a_{kk}^{(k)} = \frac{\det A_k}{\det A_{k-1}}, \qquad k = 1, \dots, n. \tag{2.3}$$

For $k = 1$ this is the statement $a_{11} = \det A_1$. Let $k \geq 2$ and assume that (2.3) holds for $1, \dots, k-1$; then the pivots $a_{11}^{(1)}, \dots, a_{k-1,k-1}^{(k-1)}$ are nonzero and $A^{(k)} = E_{k-1} \cdots E_1 A$ is well defined by Lemma 2.11(i). The matrix $M := E_{k-1} \cdots E_1$ is unit lower triangular (Lemma 2.8), so by (2.2) the leading block of $A^{(k)}$ is $(A^{(k)})_k = M_k A_k$ with $\det M_k = 1$, hence $\det(A^{(k)})_k = \det A_k$. On the other hand, after the steps $1, \dots, k-1$ the entries of $A^{(k)}$ below the diagonal in the columns $1, \dots, k-1$ are zero, so $(A^{(k)})_k$ is upper triangular, and its diagonal entries are $a_{jj}^{(k)} = a_{jj}^{(j)}$ for $j \leq k$ (row j is not changed after step $j-1$). By Lemma 2.8(i),

$$\det A_k = \det(A^{(k)})_k = a_{11}^{(1)} \cdots a_{k-1,k-1}^{(k-1)} \, a_{kk}^{(k)} = \det A_{k-1} \, a_{kk}^{(k)},$$

where the last equality follows by multiplying the identities (2.3) for $1, \dots, k-1$, whose right-hand sides telescope to $\det A_{k-1} / \det A_0 = \det A_{k-1}$. This proves (2.3) for k. Since $A_1, \dots, A_{n-1}$ and $A_n = A$ are regular, all pivots are nonzero, the algorithm does not break down, and $A = LU$ by Lemma 2.11.

Uniqueness. Let $A = \tilde{L}\tilde{U} = \hat{L}\hat{U}$ be two LU decompositions. By the necessity part, $\tilde{U}$ and $\hat{U}$ are regular, so $\hat{L}^{-1}\tilde{L} = \hat{U}\tilde{U}^{-1}$. By Lemma 2.8 the left-hand side is unit lower triangular and the right-hand side is upper triangular, so both are equal to the identity, that is, $\tilde{L} = \hat{L}$ and $\tilde{U} = \hat{U}$. In particular, the unique LU decomposition is the one produced by Algorithm 2.10. □

2.4.2 Solving $Ax = b$ with the LU decomposition

Once $A = LU$ is known, the system $Ax = b$ is equivalent to $L(Ux) = b$ and is solved by

$$\text{(1) } Ly = b \text{ (forward substitution)}, \qquad \text{(2) } Ux = y \text{ (backward substitution)}.$$

Cost. Step k of the elimination requires $n-k$ divisions for the multipliers and $(n-k)^2$ multiplications and subtractions each for the update of the trailing block, in total

$$\sum_{k=1}^{n-1} \big(2(n-k)^2 + (n-k)\big) = 2\sum_{m=1}^{n-1} m^2 + \sum_{m=1}^{n-1} m = \frac{(n-1)n(2n-1)}{3} + \frac{n(n-1)}{2} = \frac{2}{3}n^3 + O(n^2)$$

flops. The two substitutions cost $2n^2 - n$ flops (the forward substitution with L needs no divisions, since L has a unit diagonal), that is, about $2n^2$ flops. Consequently:

- The decomposition ($\frac{2}{3}n^3$ flops) is the expensive part and is computed *once*.
- Each additional right-hand side—for instance the columns of A^{-1}, or a sequence of time steps with the same matrix (Chapter 10)—costs only about $2n^2$ flops.
- Computing A^{-1} explicitly (an LU decomposition plus n pairs of substitutions, about $2n^3$ flops when the zeros in the right-hand sides e_j are exploited) and then forming $A^{-1}b$ costs about three times as much as solving $Ax = b$ directly, and the result is in general less accurate: unlike the solution by substitution, the vector $A^{-1}b$ formed with the computed inverse need not have a small residual (Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd edition, Section 14.1). A matrix is therefore not inverted in order to solve a linear system; one factorises it and substitutes.

- $\det A = \prod_{k=1}^{n} u_{kk}$ (times ± 1 when rows are exchanged, see Section 2.4.3); this is how determinants are computed in practice. Since the product easily overflows or underflows, one accumulates the sign and $\log|\det A| = \sum_k \log|u_{kk}|$ separately (`numpy.linalg.slogdet`).

Example 2.14 (LU decomposition by hand)**.** Let $A = \begin{pmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{pmatrix}$ and $b = (5, -2, 9)^\top$. Step 1: the pivot is $a_{11} = 2$, the multipliers are $l_{21} = 4/2 = 2$ and $l_{31} = -2/2 = -1$, and subtracting 2 times row 1 from row 2 and (-1) times row 1 from row 3 gives

$$A^{(2)} = \begin{pmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 8 & 3 \end{pmatrix}.$$

Step 2: the pivot is $a^{(2)}_{22} = -8$, the multiplier is $l_{32} = 8/(-8) = -1$, and subtracting (-1) times row 2 from row 3 gives

$$U = A^{(3)} = \begin{pmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{pmatrix}, \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix}.$$

One checks directly that $LU = A$. Forward substitution for $Ly = b$ gives $y_1 = 5$, $y_2 = -2 - 2\cdot 5 = -12$, $y_3 = 9 - (-1)\cdot 5 - (-1)\cdot(-12) = 2$; backward substitution for $Ux = y$ gives $x_3 = 2$, $x_2 = (-12 + 2\cdot 2)/(-8) = 1$, $x_1 = (5 - 1 - 2)/2 = 1$, so $x = (1, 1, 2)^\top$. The output below reproduces this computation.

```
L =
 [[ 1.  0.  0.]
 [ 2.  1.  0.]
 [-1. -1.  1.]]
U =
 [[ 2.  1.  1.]
 [ 0. -8. -2.]
 [ 0.  0.  1.]]
L@U - A =
 [[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
x = [1. 1. 2.]   residual: [0. 0. 0.]
```

2.4.3 Pivoting

Gaussian elimination without pivoting fails for the regular matrix $A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, because $a_{11} = 0$ (the leading principal submatrix A_1 is singular, in accordance with Theorem 2.13). It is also *unstable* when a pivot is small compared to the entries below it.

Example 2.15 (small pivot)**.** Let $A = \begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}$ with $0 < \varepsilon \ll 1$. Elimination without pivoting gives the multiplier $l_{21} = 1/\varepsilon$ and $u_{22} = 1 - 1/\varepsilon$. In floating-point arithmetic with $\varepsilon = 10^{-17}$ the value $1 - 10^{17}$ is rounded to -10^{17}, so the computed factors are

$$\tilde{L} = \begin{pmatrix} 1 & 0 \\ 10^{17} & 1 \end{pmatrix}, \qquad \tilde{U} = \begin{pmatrix} 10^{-17} & 1 \\ 0 & -10^{17} \end{pmatrix}, \qquad \tilde{L}\tilde{U} = \begin{pmatrix} \varepsilon & 1 \\ 1 & 0 \end{pmatrix}.$$

The entry $a_{22} = 1$ has been lost completely, although the matrix is well-conditioned: $\operatorname{cond}_2(A) \approx 2.6$ for small ε. The large multiplier amplified the rounding error committed in u_{22}.

The remedy is to exchange rows so that the pivot is the entry of largest modulus in its column: this is **partial pivoting**. In step k one chooses $p \geq k$ with $|a_{pk}^{(k)}| = \max_{i \geq k} |a_{ik}^{(k)}|$, exchanges the rows k and p (of the matrix and of the right-hand side), and then eliminates as before. All multipliers then satisfy $|l_{ik}| \leq 1$. A row exchange is the multiplication from the left with a *permutation matrix*, that is, a matrix obtained from I by permuting its rows; permutation matrices are orthogonal, and products of permutation matrices are permutation matrices.

Theorem 2.16 (LU decomposition with partial pivoting)**.** *For every regular matrix $A \in \mathbb{R}^{n \times n}$ there are a permutation matrix P, a unit lower triangular matrix L with $|l_{ij}| \leq 1$ for all i, j, and a regular upper triangular matrix U such that*

$$PA = LU.$$

The system $Ax = b$ is then solved via $LUx = Pb$ by two substitutions.

Proof *. We argue by induction on n. For $n = 1$ take $P = L = (1)$ and $U = A$. Let $n \geq 2$ and assume the statement for regular matrices of size $n - 1$. Since A is regular, its first column is not zero; choose p with $|a_{p1}| = \max_i |a_{i1}| > 0$ and let P_1 be the permutation matrix that exchanges the rows 1 and p. Write

$$P_1 A = \begin{pmatrix} \alpha & w^{\top} \\ v & B \end{pmatrix}, \qquad \alpha = a_{p1} \neq 0,\ v \in \mathbb{R}^{n-1},\ w \in \mathbb{R}^{n-1},\ B \in \mathbb{R}^{(n-1)\times(n-1)},$$

where $|v_i| \leq |\alpha|$ for all i by the choice of p. Then

$$P_1 A = \begin{pmatrix} 1 & 0 \\ v/\alpha & I \end{pmatrix} \begin{pmatrix} \alpha & w^{\top} \\ 0 & S \end{pmatrix}, \qquad S := B - \frac{v w^{\top}}{\alpha} \in \mathbb{R}^{(n-1)\times(n-1)}, \tag{2.4}$$

as one verifies by multiplying out. Taking determinants, $\pm \det A = \det(P_1 A) = \alpha \det S$, so S is regular. By the induction hypothesis there are a permutation matrix P', a unit lower triangular L' with entries of modulus at most 1 and a regular upper triangular U' with $P'S = L'U'$. Define

$$P := \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix} P_1, \qquad L := \begin{pmatrix} 1 & 0 \\ P'v/\alpha & L' \end{pmatrix}, \qquad U := \begin{pmatrix} \alpha & w^{\top} \\ 0 & U' \end{pmatrix}.$$

P is a permutation matrix, L is unit lower triangular with all entries of modulus at most 1 (the entries of $P'v/\alpha$ are the numbers v_i/α in a different order), and U is upper triangular and regular ($\det U = \alpha \det U' \neq 0$). Finally, using (2.4),

$$PA = \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix} \begin{pmatrix} \alpha & w^{\top} \\ v & B \end{pmatrix} = \begin{pmatrix} \alpha & w^{\top} \\ P'v & P'B \end{pmatrix} \quad \text{and} \quad LU = \begin{pmatrix} \alpha & w^{\top} \\ P'v & \dfrac{P'vw^{\top}}{\alpha} + L'U' \end{pmatrix},$$

and these agree because $L'U' = P'S = P'B - P'vw^{\top}/\alpha$. This completes the induction. The elimination with partial pivoting described above produces exactly this factorisation. □

Remark 2.17* (growth factor and backward stability). Let A be regular and let $a_{ij}^{(k)}$ denote the entries produced by Gaussian elimination with partial pivoting in exact arithmetic. The *growth factor* is

$$\rho_n := \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}.$$

Since $|l_{ik}| \leq 1$, the update (2.1) gives $|a_{ij}^{(k+1)}| \leq |a_{ij}^{(k)}| + |a_{kj}^{(k)}| \leq 2 \max_{i,j} |a_{ij}^{(k)}|$, so the largest modulus at most doubles in every step and $\rho_n \leq 2^{n-1}$. This bound is attained, for instance for

the matrix with $a_{ii} = 1$, $a_{ij} = -1$ for $i > j$, $a_{in} = 1$ for all i, and $a_{ij} = 0$ otherwise, when ties in the pivot search are resolved in favour of the current row. The rounding error analysis of Gaussian elimination (J. H. Wilkinson; see N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd edition, Theorems 9.3 and 9.4) shows that the computed factors $\tilde{L}, \tilde{U}$ satisfy

$$\tilde{L}\tilde{U} = PA + \Delta A, \qquad |\Delta A| \le \gamma_n |\tilde{L}|\,|\tilde{U}| \ \text{ entrywise}, \qquad \gamma_n := \frac{nu}{1 - nu} \quad (nu < 1),$$

and hence $\|\Delta A\|_\infty \le n^2 \gamma_n \tilde{\rho}_n \|A\|_\infty$, where $\tilde{\rho}_n$ is the growth factor formed with the computed entries (it differs from ρ_n by terms of higher order in u); a similar bound with γ_{3n} in place of γ_n holds for the backward error of the computed solution of $Ax = b$. Thus the backward error is of the order $n^3 \rho_n u \|A\|_\infty$ in the worst case. Because of the factor $\rho_n \le 2^{n-1}$, Gaussian elimination with partial pivoting is not backward stable in the strict sense of Definition 1.10: the constant in the worst-case bound grows exponentially with n and is not of moderate size. For the vast majority of matrices arising in applications, however, the growth factor is moderate and the worst-case bound is pessimistic by a large factor; matrices with exponential growth are rare, although not purely artificial (S. J. Wright, 1993, gives examples from discretised boundary value problems; see L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, Lecture 22, for an explanation of why large growth is so rare). For these reasons Gaussian elimination with partial pivoting is the standard method for dense linear systems.

2.5 Symmetric positive definite matrices and the Cholesky decomposition

Definition 2.18 (symmetric positive definite matrix)**.** A matrix $A \in \mathbb{R}^{n \times n}$ is called **symmetric positive definite (SPD)** if $A = A^\top$ and $x^\top A x > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$. It is called symmetric positive *semi*definite if $A = A^\top$ and $x^\top A x \ge 0$ for all x.

SPD matrices are ubiquitous: the matrices $A^\top A$ of the normal equations for a matrix A with full column rank (Chapter 3), Hessian matrices at non-degenerate minima (at a general minimum the Hessian is only positive semidefinite), discretised elliptic differential operators (Example 2.25), and covariance matrices of random vectors whose centred components are linearly independent in L^2 (in general covariance matrices are only positive semidefinite).

Proposition 2.19 (properties of SPD matrices)**.** *Let $A \in \mathbb{R}^{n \times n}$ be SPD. Then*

(i) A is regular;

(ii) all leading principal submatrices A_k, $k = 1, \dots, n$, are SPD;

(iii) all eigenvalues of A are positive, and $\det A > 0$*;*

(iv) $a_{ii} > 0$ for all i, and $\max_{i,j} |a_{ij}| = \max_i a_{ii}$*.*

Proof. (i) If $Ax = 0$, then $x^\top A x = 0$, hence $x = 0$. (ii) A_k is symmetric. For $y \in \mathbb{R}^k \setminus \{0\}$ let $x := (y, 0)^\top \in \mathbb{R}^n$; then $y^\top A_k y = x^\top A x > 0$. (iii) The eigenvalues of the symmetric matrix A are real, and $Av = \lambda v$ with $v \neq 0$ gives $\lambda \|v\|_2^2 = v^\top A v > 0$. The determinant is the product of the eigenvalues, hence positive. (iv) $a_{ii} = e_i^\top A e_i > 0$. For $i \neq j$ we have $0 < (e_i \pm e_j)^\top A (e_i \pm e_j) = a_{ii} + a_{jj} \pm 2a_{ij}$, hence $|a_{ij}| < \tfrac{1}{2}(a_{ii} + a_{jj}) \le \max_i a_{ii}$. □

By (ii) and Theorem 2.13, an SPD matrix has an LU decomposition, and Gaussian elimination without pivoting does not break down. The symmetry allows us to halve the work.

Theorem 2.20 (Cholesky decomposition)**.** *A matrix $A \in \mathbb{R}^{n\times n}$ is SPD if and only if there is a lower triangular matrix L with positive diagonal entries such that*

$$A = LL^\top.$$

In this case L is unique.

*Proof *.* "⇐": $A = LL^\top$ is symmetric, and $L^\top$ is regular by Lemma 2.8(i) because its diagonal entries are positive. For $x \neq 0$ we therefore have $L^\top x \neq 0$ and $x^\top LL^\top x = \|L^\top x\|_2^2 > 0$.

"⇒": We argue by induction on n. For $n = 1$ the matrix is $A = (a_{11})$ with $a_{11} > 0$, and $L = (\sqrt{a_{11}})$ is the unique lower triangular matrix with positive diagonal and $LL^\top = A$. Let $n \geq 2$ and write

$$A = \begin{pmatrix} A_{n-1} & a \\ a^\top & \alpha \end{pmatrix}, \qquad a \in \mathbb{R}^{n-1},\ \alpha \in \mathbb{R}.$$

By Proposition 2.19(ii), A_{n-1} is SPD, so by the induction hypothesis there is a unique lower triangular L_{n-1} with positive diagonal and $A_{n-1} = L_{n-1}L_{n-1}^\top$. We look for L of the form

$$L = \begin{pmatrix} L_{n-1} & 0 \\ \ell^\top & \lambda \end{pmatrix}, \qquad \ell \in \mathbb{R}^{n-1},\ \lambda > 0.$$

Multiplying out, $LL^\top = \begin{pmatrix} L_{n-1}L_{n-1}^\top & L_{n-1}\ell \\ \ell^\top L_{n-1}^\top & \ell^\top\ell + \lambda^2 \end{pmatrix}$, so $A = LL^\top$ holds if and only if $L_{n-1}\ell = a$ and $\ell^\top\ell + \lambda^2 = \alpha$. The first equation has the unique solution $\ell = L_{n-1}^{-1}a$ (forward substitution). For the second we need $\alpha - \ell^\top\ell > 0$. With $z := A_{n-1}^{-1}a$ we have $\ell^\top\ell = a^\top L_{n-1}^{-\top}L_{n-1}^{-1}a = a^\top(L_{n-1}L_{n-1}^\top)^{-1}a = a^\top A_{n-1}^{-1}a = a^\top z$, and for the vector $x := (-z, 1)^\top \in \mathbb{R}^n \setminus \{0\}$,

$$0 < x^\top Ax = z^\top A_{n-1}z - 2a^\top z + \alpha = a^\top z - 2a^\top z + \alpha = \alpha - a^\top z = \alpha - \ell^\top\ell.$$

Hence $\lambda := \sqrt{\alpha - \ell^\top\ell} > 0$ is well defined and is the unique positive number with $\ell^\top\ell + \lambda^2 = \alpha$. This proves existence; uniqueness follows from the same argument, since L_{n-1}, ℓ and λ were determined uniquely. □

Algorithm 2.21 (Cholesky decomposition, column by column)**.** Comparing the entries of $A = LL^\top$, that is, $a_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik}l_{jk}$, column by column gives for $j = 1, \dots, n$

$$l_{jj} = \Big(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2\Big)^{1/2}, \qquad l_{ij} = \frac{1}{l_{jj}}\Big(a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}\Big) \quad (i = j+1, \dots, n).$$

Remark 2.22 (properties of the Cholesky decomposition)*.*

(i) *Cost and storage.* The computation of column j requires about $(n-j)j$ multiplications and as many additions, so the total cost is $\sum_{j=1}^{n} 2(n-j)j + O(n^2) = \frac{1}{3}n^3 + O(n^2)$ flops, half the cost of the LU decomposition, and n square roots. Only the lower triangle of A has to be stored and accessed.

(ii) *No pivoting, boundedness.* From $a_{ii} = \sum_{k\leq i} l_{ik}^2$ we get $|l_{ik}| \leq \sqrt{a_{ii}}$ for all $k \leq i$: the factor L cannot have large entries, so no pivoting is needed. The algorithm is backward stable: the computed factor $\tilde{L}$ satisfies $\tilde{L}\tilde{L}^\top = A + \Delta A$ with $|\Delta A| \leq \gamma_{n+1}|\tilde{L}|\,|\tilde{L}^\top|$ entrywise (Higham, *Accuracy and Stability of Numerical Algorithms*, Theorem 10.3).

(iii) *Test for positive definiteness.* In exact arithmetic, Algorithm 2.21 encounters a radicand $a_{jj} - \sum_{k<j} l_{jk}^2 \leq 0$ if and only if A is not positive definite. Indeed, if A is SPD, the radicands are the numbers $\lambda^2 > 0$ of the proof of Theorem 2.20 (applied to the leading

submatrices); conversely, if all radicands are positive, the algorithm produces a lower triangular L with positive diagonal and $LL^{\top} = A$, so A is SPD by Theorem 2.20. For a general dense symmetric matrix, attempting the Cholesky decomposition is therefore the standard efficient test for positive definiteness. In floating-point arithmetic the algorithm may stop although A is positive definite, but only if A is nearly singular: a symmetric perturbation with $\|\Delta A\|_2 < \lambda_{\min}(A)$ preserves positive definiteness, because $x^{\top}(A+\Delta A)x \geq \big(\lambda_{\min}(A) - \|\Delta A\|_2\big)\|x\|_2^2 > 0$ for $x \neq 0$, so that relative perturbations of size $\eta = \|\Delta A\|_2/\|A\|_2$ are harmless as long as $\eta\,\mathrm{cond}_2(A) < 1$; the rounding error analysis shows that the algorithm runs to completion whenever $20\,n^{3/2}\,\mathrm{cond}_2(A)\,u \leq 1$ (J. H. Wilkinson, 1968; Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd edition, Theorem 10.7).

Example 2.23 (Cholesky decomposition by hand)**.** For $A = \begin{pmatrix} 4 & 2 & -2 \\ 2 & 10 & 2 \\ -2 & 2 & 5 \end{pmatrix}$ Algorithm 2.21 gives

$$l_{11} = \sqrt{4} = 2, \quad l_{21} = 2/2 = 1, \quad l_{31} = -2/2 = -1,$$
$$l_{22} = \sqrt{10 - 1^2} = 3, \quad l_{32} = \big(2 - (-1)\cdot 1\big)/3 = 1, \qquad l_{33} = \sqrt{5 - (-1)^2 - 1^2} = \sqrt{3},$$

so $L = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 3 & 0 \\ -1 & 1 & \sqrt{3} \end{pmatrix}$, and all radicands are positive; hence A is SPD. For the symmetric matrix $\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ we get $l_{11} = 1$, $l_{21} = 2$ and the radicand $1 - 2^2 = -3 < 0$ in the second step: the matrix is not positive definite (its eigenvalues are 3 and -1). The output below reproduces both computations; the residual $LL^{\top} - A$ vanishes to the displayed precision.

```
L =
 [[ 2.      0.      0.    ]
 [ 1.      3.      0.    ]
 [-1.      1.      1.7321]]
L L^T - A =
 [[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
indefinite matrix: nonpositive radicand in column 1: not positive definite
```

2.6 Exploiting structure: banded and sparse matrices*

Matrices arising from discretised differential equations are **sparse**—most of their entries are zero—and often **banded**: A has *bandwidth* w (more precisely, half-bandwidth w) if $a_{ij} = 0$ for $|i-j| > w$, so that at most $2w+1$ diagonals of A are nonzero; such a matrix is stored in $O(nw)$ memory.

Proposition 2.24 (Gaussian elimination preserves the band)**.** *Let $A \in \mathbb{R}^{n\times n}$ have bandwidth w and suppose that Gaussian elimination without pivoting does not break down. Then every $A^{(k)}$ has bandwidth w, $l_{ik} = 0$ for $i - k > w$ and $u_{kj} = 0$ for $j - k > w$. The decomposition costs at most $2nw^2 + O(nw)$ flops and each substitution at most $2nw + O(n)$ flops.*

Proof. We show by induction on k that $A^{(k)}$ has bandwidth w; for $k = 1$ this is the assumption. If $A^{(k)}$ has bandwidth w, then $l_{ik} = a_{ik}^{(k)}/a_{kk}^{(k)} = 0$ for $i - k > w$ and $a_{kj}^{(k)} = 0$ for $j - k > w$. In the update $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik}a_{kj}^{(k)}$ the subtracted term is nonzero only if $k < i \leq k + w$ and

$k < j \le k + w$, which implies $|i - j| < w$; all entries with $|i - j| > w$ therefore remain zero, and $A^{(k+1)}$ has bandwidth w. The statements about L and U follow, since $U = A^{(n)}$ and the l_{ik} are the multipliers. In step k at most w multipliers and a block of at most $w \times w$ entries are updated, which costs at most $2w^2 + w$ flops; summing over k gives the cost of the decomposition, and in a substitution each of the n rows involves at most w off-diagonal entries. □

For a banded matrix the LU decomposition therefore costs $O(nw^2)$ instead of $\frac{2}{3}n^3$ flops, and the substitutions cost $O(nw)$. For tridiagonal matrices ($w = 1$) this is the **Thomas algorithm**, with $O(n)$ cost in total. With partial pivoting the band is not preserved exactly, but the fill-in remains controlled: U has at most $2w + 1$ nonzero diagonals and every column of L has at most $w + 1$ nonzero entries (G. H. Golub and C. F. Van Loan, *Matrix Computations*, Section 4.3), so the cost remains $O(nw^2)$; for SPD matrices (Cholesky) and for strictly diagonally dominant matrices (Chapter 5) no pivoting is needed at all. For general sparse matrices, direct solvers reorder the unknowns so as to limit the *fill-in*, that is, the creation of nonzero entries in L and U at positions where A has zeros; when the fill-in makes a direct solver too expensive, one resorts to the iterative methods of Chapter 6.

Example 2.25 (one-dimensional Poisson problem)**.** The boundary value problem $-u''(t) = f(t)$ on $(0, 1)$ with $u(0) = u(1) = 0$ is discretised on the grid $t_i = ih$, $i = 0, \dots, n + 1$, $h = 1/(n + 1)$, by replacing $-u''(t_i)$ with the central difference quotient $\big(-u_{i-1} + 2u_i - u_{i+1}\big)/h^2$, where $u_i \approx u(t_i)$ and $u_0 = u_{n+1} = 0$. This gives the tridiagonal system

$$A_h u = F, \qquad A_h = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n\times n}, \qquad u = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}, \quad F = \begin{pmatrix} f(t_1) \\ \vdots \\ f(t_n) \end{pmatrix}.$$

The eigenvalues and eigenvectors of A_h are known explicitly: for $k = 1, \dots, n$ let $v^{(k)} := \big(\sin(k\pi jh)\big)_{j=1}^{n}$. Using $\sin(\alpha - \beta) + \sin(\alpha + \beta) = 2\sin\alpha\cos\beta$ and $1 - \cos\beta = 2\sin^2(\beta/2)$ we obtain for $j = 1, \dots, n$

$$\begin{aligned} \big(A_h v^{(k)}\big)_j &= \frac{1}{h^2}\big(-\sin(k\pi(j-1)h) + 2\sin(k\pi jh) - \sin(k\pi(j+1)h)\big) \\ &= \frac{2 - 2\cos(k\pi h)}{h^2}\sin(k\pi jh) = \frac{4}{h^2}\sin^2\frac{k\pi h}{2}\,\sin(k\pi jh), \end{aligned}$$

where for $j = 1$ and $j = n$ the missing neighbours are accounted for by $\sin(0) = 0$ and $\sin(k\pi(n+1)h) = \sin(k\pi) = 0$. Hence

$$\lambda_k = \frac{4}{h^2}\sin^2\frac{k\pi h}{2}, \qquad k = 1, \dots, n,$$

are the eigenvalues of A_h with eigenvectors $v^{(k)}$; they are positive, so A_h is SPD by Proposition 2.19(iii) and the spectral theorem (a symmetric matrix with positive eigenvalues is positive definite). By Proposition 2.6(i) and $\sin\frac{n\pi h}{2} = \sin\big(\frac{\pi}{2} - \frac{\pi h}{2}\big) = \cos\frac{\pi h}{2}$,

$$\operatorname{cond}_2(A_h) = \frac{\lambda_n}{\lambda_1} = \frac{\cos^2(\pi h/2)}{\sin^2(\pi h/2)} = \cot^2\frac{\pi h}{2} \approx \frac{4}{\pi^2 h^2} = O(n^2)\,:$$

finer grids give worse conditioning. This matrix will accompany us in Chapters 5, 6 and 10.

2.7 Summary

- $\text{cond}(A) = \|A\| \, \|A^{-1}\| \geq 1$ bounds the amplification of relative data errors in $Ax = b$; a small residual implies a small error only if $\text{cond}(A)$ is small.
- Gaussian elimination with partial pivoting computes $PA = LU$ in $\frac{2}{3}n^3 + O(n^2)$ flops; each right-hand side then costs about $2n^2$ flops. Partial pivoting keeps $|l_{ij}| \leq 1$; the backward error contains the growth factor $\rho_n \leq 2^{n-1}$, which is moderate for most matrices in practice.
- SPD matrices admit the Cholesky decomposition $A = LL^\top$ ($\frac{1}{3}n^3$ flops, no pivoting); the algorithm is also a test for positive definiteness.
- Banded and sparse structure reduces the cost dramatically (for bandwidth w: $O(nw^2)$); direct solvers exploit it, and when the fill-in becomes too large, iterative methods (Chapter 6) are the alternative.

3 Least squares problems and the QR decomposition

Learning goals. After this chapter you can

(i) formulate data fitting as a linear least squares problem and characterise its solutions by the normal equations,

(ii) explain the geometry of least squares (orthogonal projection),

(iii) explain why solving the normal equations squares the condition number,

(iv) construct a QR decomposition by Householder reflections, use it to solve least squares problems with full column rank, and explain why this method is numerically reliable (Gram–Schmidt orthogonalisation as the alternative construction is optional, Section 3.4).

The conditioning of the least squares problem itself, Gram–Schmidt orthogonalisation, Givens rotations and the polynomial regression experiment are not lectured; they are included for self-study and marked with an asterisk (*).

3.1 Overdetermined systems and data fitting

Suppose a quantity y is measured at times $t_1, \dots, t_m$ and is assumed to satisfy $y(t) = x_1\phi_1(t) + \cdots + x_n\phi_n(t)$ for known functions $\phi_1, \dots, \phi_n$ (for instance $1, t, t^2$) and unknown parameters $x_1, \dots, x_n$. Each measurement gives one linear equation,

$$\sum_{j=1}^{n} \phi_j(t_i)\, x_j = y_i, \quad i = 1, \dots, m, \qquad \text{that is,} \qquad Ax = b \quad \text{with} \quad A = \big(\phi_j(t_i)\big)_{i,j} \in \mathbb{R}^{m\times n},$$

where $b = (y_1, \dots, y_m)^\top \in \mathbb{R}^m$ and the unknown vector is $x = (x_1, \dots, x_n)^\top$. Typically $m \gg n$ (many measurements, few parameters). Because of measurement errors, and because the model is only an approximation of reality, the system has no solution in general—$b \notin \operatorname{Im} A$—and one looks instead for the parameter vector x that fits the data best.

Definition 3.1 (linear least squares problem)**.** Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. The **(linear) least squares problem** is to find $x^* \in \mathbb{R}^n$ with

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

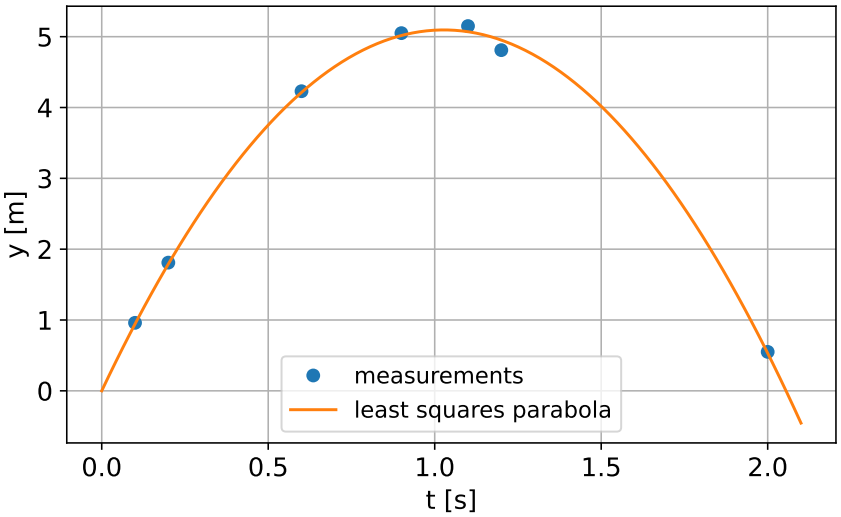
Every such x^* is called a *least squares solution* of $Ax = b$, and $r := b - Ax^*$ is the **residual**.

Remark 3.2. The Euclidean norm is used for certain reasons: it leads to a *linear* characterisation of the solutions (Theorem 3.4) and it has a clear geometric meaning (orthogonal projection).

Example 3.3 (projectile)**.** A body launched at time 0 from the origin with vertical velocity v_y under the constant gravitational acceleration g (air resistance neglected) is at height $y(t) =$

$v_y t - \frac{1}{2} g t^2$. From measured heights y_i at times t_i we want to determine v_y and g. This is a *linear* least squares problem—linear in the unknown parameters, although y depends quadratically on t—with $\phi_1(t) = t$, $\phi_2(t) = -t^2/2$ and unknowns $x = (v_y, g)^\top$. For the seven measurements $t = (0.1, 0.2, 0.6, 0.9, 1.1, 1.2, 2.0)$ (in seconds) and $y = (0.96, 1.81, 4.23, 5.05, 5.15, 4.81, 0.55)$ (in metres), the least squares solution and the norm of the residual are given below; the fitted parabola is shown in the figure.

```
v_y = 9.918 m/s,  g = 9.652 m/s^2,  ||r||_2 = 0.170
```



3.2 The normal equations and the geometry of least squares

We recall from linear algebra: for $A \in \mathbb{R}^{m\times n}$ the *image* (column space) is $\operatorname{Im} A = \{Ax : x \in \mathbb{R}^n\} \subset \mathbb{R}^m$, the *kernel* is $\ker A = \{x \in \mathbb{R}^n : Ax = 0\}$, and $\operatorname{rank} A = \dim \operatorname{Im} A = \operatorname{rank} A^\top$, $\dim \ker A = n - \operatorname{rank} A$. Moreover $(\operatorname{Im} A)^\perp = \ker A^\top$, because $y \perp Ax$ for all x is equivalent to $x^\top(A^\top y) = 0$ for all x, that is, to $A^\top y = 0$. Hence $\mathbb{R}^m$ is the orthogonal direct sum

$$\mathbb{R}^m = \operatorname{Im} A \oplus \ker A^\top, \tag{3.1}$$

that is, every $b \in \mathbb{R}^m$ can be written uniquely as $b = p + q$ with $p \in \operatorname{Im} A$, $q \in \ker A^\top$, and $p \perp q$.

Theorem 3.4 (normal equations)**.** *Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. A vector $x^* \in \mathbb{R}^n$ is a least squares solution of $Ax = b$ if and only if it satisfies the* ***normal equations***

$$A^\top A\, x^* = A^\top b.$$

A least squares solution always exists. It is unique if and only if $\operatorname{rank} A = n$ *(full column rank,* $\ker A = \{0\}$*), and in this case $A^\top A$ is SPD and $x^* = (A^\top A)^{-1} A^\top b$.*

Proof. Decompose $b = p + q$ according to (3.1). For every $x \in \mathbb{R}^n$ the vector $Ax - p$ lies in $\operatorname{Im} A$ and is therefore orthogonal to q, so by Pythagoras' theorem

$$\|Ax - b\|_2^2 = \|(Ax - p) - q\|_2^2 = \|Ax - p\|_2^2 + \|q\|_2^2 \;\geq\; \|q\|_2^2,$$

with equality if and only if $Ax = p$. Since $p \in \operatorname{Im} A$, there is x with $Ax = p$; this proves existence, and it shows that x^* is a least squares solution if and only if $Ax^* = p$. Now $Ax^* = p$

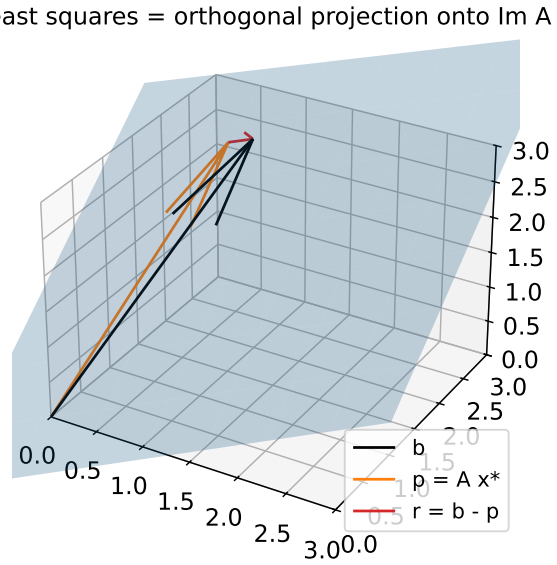
holds if and only if $b - Ax^* = q \in \ker A^\top$ (by uniqueness of the decomposition), that is, if and only if $A^\top(b - Ax^*) = 0$, which are the normal equations.

Two least squares solutions $x^*, \hat{x}$ satisfy $Ax^* = A\hat{x} = p$, so they differ by an element of $\ker A$; conversely, if x^* is a solution, so is $x^* + z$ for every $z \in \ker A$. Hence the solution is unique if and only if $\ker A = \{0\}$. Finally, if $\ker A = \{0\}$, then $A^\top A$ is symmetric and $x^\top A^\top A x = \|Ax\|_2^2 > 0$ for $x \neq 0$, so $A^\top A$ is SPD, in particular regular, and the normal equations give $x^* = (A^\top A)^{-1} A^\top b$. □

Geometrically, $Ax^* = p$ is the **orthogonal projection** of b onto $\operatorname{Im} A$, and the residual $r = b - Ax^* = q$ is orthogonal to all columns of A, that is, $A^\top r = 0$ (the residual is *normal* to $\operatorname{Im} A$, whence the name). Note that the projection p and the residual r are always unique, even when the least squares solution x^* is not. If A has full column rank, the projection is given by the matrix $P := A(A^\top A)^{-1} A^\top$, which satisfies $P^2 = P$ and $P^\top = P$ (as one checks by multiplying out), and $Ax^* = Pb$. The formulas $x^* = (A^\top A)^{-1} A^\top b$ and $P = A(A^\top A)^{-1} A^\top$ describe the solution and the projection; in computations the inverse is not formed, as for square systems: the normal equations are solved by a Cholesky decomposition (Section 3.2.1) or, better, the problem is solved via a QR decomposition, which also gives $P = Q_1 Q_1^\top$ (Section 3.3).

Example 3.5 (orthogonal projection in $\mathbb{R}^3$)**.** Let $A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $b = (1, 2, 3)^\top$. The columns of A span a plane in $\mathbb{R}^3$. The normal equations $A^\top A x = A^\top b$ read $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} x = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$ and have the solution $x^* = (\frac{1}{3}, \frac{7}{3})^\top$. The projection is $p = Ax^* = (\frac{1}{3}, \frac{8}{3}, \frac{7}{3})^\top$ and the residual is $r = b - p = \frac{2}{3}(1, -1, 1)^\top$, which is orthogonal to both columns of A: $A^\top r = 0$; the minimal value of $\|Ax - b\|_2$ is $\|r\|_2 = 2/\sqrt{3}$. The figure shows b, p and r.

```
x* = [0.3333 2.3333]   projection p = [0.3333 2.6667 2.3333]   residual r = [ 0.6667 -0.6667  0.6667]
A^T r = [-0. -0.]   (orthogonality of the residual)
```



3.2.1 Solving the normal equations, and why one should be careful

If A has full column rank, $A^\top A$ is SPD, so the normal equations can be solved by the Cholesky decomposition of Section 2.5. Forming $A^\top A$ costs about mn^2 flops (by symmetry only half of

the n^2 entries, each an inner product of length m, are needed) and the Cholesky decomposition $\frac{1}{3}n^3$ flops; this is cheap. The price is accuracy.

Definition 3.6 (condition number of a rectangular matrix)**.** Let $A \in \mathbb{R}^{m\times n}$ have full column rank n. Since $A^\top A$ is SPD, its eigenvalues are positive, and we set

$$\sigma_{\max}(A) := \sqrt{\lambda_{\max}(A^\top A)}, \qquad \sigma_{\min}(A) := \sqrt{\lambda_{\min}(A^\top A)}, \qquad \operatorname{cond}_2(A) := \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}.$$

By Theorem 1.22, $\sigma_{\max}(A) = \|A\|_2$. Moreover, the matrix $A^+ := (A^\top A)^{-1}A^\top$, which maps b to the least squares solution, satisfies $\|A^+\|_2 = 1/\sigma_{\min}(A)$, hence $\operatorname{cond}_2(A) = \|A\|_2\,\|A^+\|_2$, in analogy with Definition 2.1. For a square regular matrix, $A^+ = A^{-1}$ and the definition coincides with the earlier one.

Proof of* $\|A^+\|_2 = 1/\sigma_{\min}(A)$. We have $A^+(A^+)^\top = (A^\top A)^{-1}A^\top A(A^\top A)^{-1} = (A^\top A)^{-1}$, so by Proposition 1.23(i) and Theorem 1.22 (applied to $(A^+)^\top$)

$$\|A^+\|_2 = \|(A^+)^\top\|_2 = \sqrt{\lambda_{\max}\big(A^+(A^+)^\top\big)} = \sqrt{\lambda_{\max}\big((A^\top A)^{-1}\big)} = \frac{1}{\sqrt{\lambda_{\min}(A^\top A)}} = \frac{1}{\sigma_{\min}(A)}.$$

□

Proposition 3.7. *If $A \in \mathbb{R}^{m\times n}$ has full column rank, then* $\operatorname{cond}_2(A^\top A) = \operatorname{cond}_2(A)^2$.

Proof. $A^\top A$ is SPD with eigenvalues in $[\sigma_{\min}(A)^2, \sigma_{\max}(A)^2]$, the extreme values being attained. By Proposition 2.6(i), $\operatorname{cond}_2(A^\top A) = \lambda_{\max}(A^\top A)/\lambda_{\min}(A^\top A) = \sigma_{\max}(A)^2/\sigma_{\min}(A)^2 = \operatorname{cond}_2(A)^2$. □

Remark 3.8* (conditioning of the least squares problem itself). Let $A \in \mathbb{R}^{m\times n}$ have full column rank, let $b \in \mathbb{R}^m$ with least squares solution $x^* \neq 0$, and let $\theta \in [0, \pi/2)$ be the angle between b and $\operatorname{Im} A$, that is, $\cos\theta = \|Ax^*\|_2/\|b\|_2$; put $\eta := \|A\|_2\|x^*\|_2/\|Ax^*\|_2 \in [1, \operatorname{cond}_2(A)]$. Then the relative condition numbers (Definition 1.6, with the 2-norms on $\mathbb{R}^m$, $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$) of the map $b \mapsto x^*$ and of the map $A \mapsto x^*$ satisfy

$$\kappa_{b\to x^*} = \frac{\operatorname{cond}_2(A)}{\eta\cos\theta} \le \frac{\operatorname{cond}_2(A)}{\cos\theta},$$

$$\kappa_{A\to x^*} \le \operatorname{cond}_2(A) + \frac{\operatorname{cond}_2(A)^2\tan\theta}{\eta} \le \operatorname{cond}_2(A) + \operatorname{cond}_2(A)^2\tan\theta,$$

see L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, Theorem 18.1. The first formula is an equality, the second one only an upper bound: for $A = (1,0)^\top$ and $b = (1,1)^\top$ one computes $\kappa_{A\to x^*} = \sqrt{2}$, whereas the bound gives 2. Hence for perturbations of b the problem is conditioned like $\operatorname{cond}_2(A)$, up to the factor $1/\cos\theta$. For perturbations of A the bound contains the additional term $\operatorname{cond}_2(A)^2\tan\theta/\eta$, whose ratio to the first term is $\operatorname{cond}_2(A)\tan\theta/\eta$; it is negligible only if $\tan\theta \ll \eta/\operatorname{cond}_2(A)$. Whether a residual counts as small must therefore be judged on the scale $1/\operatorname{cond}_2(A)$: for an ill-conditioned matrix a residual that looks small can still make the term with $\operatorname{cond}_2(A)^2$ dominant.

Sketch of proof.* If f is differentiable at x, its relative condition number there is $\|Df(x)\|\,\|x\|/\|f(x)\|$, with the operator norm of the derivative (this is the multivariate form of Lemma 1.7). The map $b \mapsto x^* = A^+b$ is linear, so $\kappa_{b\to x^*} = \|A^+\|_2\|b\|_2/\|x^*\|_2 = \|b\|_2/\big(\sigma_{\min}(A)\|x^*\|_2\big)$, and $\|b\|_2 = \|Ax^*\|_2/\cos\theta$ turns this into $\operatorname{cond}_2(A)/(\eta\cos\theta)$. The map $A \mapsto x^* = (A^\top A)^{-1}A^\top b$ is differentiable on the open set of matrices with full column

rank; differentiating the normal equations $A^\top A x^* = A^\top b$ in the direction $E \in \mathbb{R}^{m\times n}$ gives $(E^\top A + A^\top E)x^* + A^\top A\, Dx^*[E] = E^\top b$, that is,

$$Dx^*[E] = (A^\top A)^{-1}E^\top r - A^+ E x^*, \qquad r = b - Ax^*.$$

Hence $\|Dx^*[E]\|_2 \le \|E\|_2\|r\|_2/\sigma_{\min}(A)^2 + \|E\|_2\|x^*\|_2/\sigma_{\min}(A)$, and since $\|r\|_2 = \|Ax^*\|_2 \tan\theta$, multiplying by $\|A\|_2/(\|E\|_2\|x^*\|_2)$ gives the bound for $\kappa_{A\to x^*}$.

Example 3.9 (Läuchli)**.** Let $A = \begin{pmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \end{pmatrix}$ with $0 < \varepsilon \ll 1$. Then $A^\top A = \begin{pmatrix} 1+\varepsilon^2 & 1 \\ 1 & 1+\varepsilon^2 \end{pmatrix}$ has the eigenvalues $2+\varepsilon^2$ and ε^2, so $\operatorname{cond}_2(A) = \sqrt{(2+\varepsilon^2)/\varepsilon^2} \approx \sqrt{2}/\varepsilon$. For $\varepsilon = 10^{-8}$ we have $\operatorname{cond}_2(A) \approx 1.4 \cdot 10^8$, which is far from $1/u$, but $\mathrm{fl}(1+\varepsilon^2) = 1$ because $\varepsilon^2 = 10^{-16} < u$, so the computed matrix $A^\top A$ is the singular matrix $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. With the right-hand side $b = (2, \varepsilon, \varepsilon)^\top$ the least squares solution is $x^* = (1,1)^\top$, which even solves $Ax = b$ exactly. The Cholesky decomposition of the computed normal equations fails, whereas the QR approach of the next section returns x^* to working precision. (The output reports a finite value of about $6 \cdot 10^{16}$ for the condition number of the computed matrix $A^\top A$, although this matrix is exactly singular: its smallest singular value is computed as a number of the order u instead of 0.)

```
cond_2(A) = 141421356.23730955    cond_2(A^T A) computed = 5.961777047638983e+16
normal equations + Cholesky failed: 2-th leading minor of the array is not positive definite
via QR                     : [1. 1.]
```

3.3 Orthogonal matrices and the QR decomposition

The way out is to avoid forming $A^\top A$ and to work with orthogonal transformations, which do not change the Euclidean norm and hence neither the least squares problem nor its conditioning.

Definition 3.10 (orthogonal matrix)**.** A matrix $Q \in \mathbb{R}^{m\times m}$ is called **orthogonal** if $Q^\top Q = I$. A matrix $Q_1 \in \mathbb{R}^{m\times n}$ with $n \le m$ is said to have *orthonormal columns* if $Q_1^\top Q_1 = I \in \mathbb{R}^{n\times n}$.

Lemma 3.11 (properties of orthogonal matrices)**.** *Let $Q \in \mathbb{R}^{m\times m}$ be orthogonal.*

(i) Q is regular with $Q^{-1} = Q^\top$, and $Q^\top$ is orthogonal; the columns of Q form an orthonormal basis of $\mathbb{R}^m$. Conversely, every matrix whose columns form an orthonormal basis of $\mathbb{R}^m$ is orthogonal.

(ii) $\|Qx\|_2 = \|x\|_2$ for all $x \in \mathbb{R}^m$, $\|QA\|_2 = \|A\|_2$ for all $A \in \mathbb{R}^{m\times n}$, and $\operatorname{cond}_2(Q) = 1$.

(iii) Products of orthogonal matrices are orthogonal.

(iv) If $Q_1 \in \mathbb{R}^{m\times n}$ has orthonormal columns, then $\|Q_1 y\|_2 = \|y\|_2$ for all $y \in \mathbb{R}^n$.

Proof. (i) $Q^\top Q = I$ says that $Q^\top$ is a left inverse of the square matrix Q; hence Q is regular and $Q^{-1} = Q^\top$, so also $QQ^\top = I$, which is the orthogonality of $Q^\top$. The (i,j) entry of $Q^\top Q = I$ is the inner product of the columns i and j of Q, so the columns are orthonormal, and m orthonormal vectors form a basis of $\mathbb{R}^m$; the converse is the same computation read backwards. (ii) $\|Qx\|_2^2 = x^\top Q^\top Q x = x^\top x = \|x\|_2^2$; hence $\|QAx\|_2 = \|Ax\|_2$ for all x and $\|QA\|_2 = \|A\|_2$; finally $\|Q\|_2 = 1$ and $\|Q^{-1}\|_2 = \|Q^\top\|_2 = 1$ by (i). (iii) $(Q_1Q_2)^\top(Q_1Q_2) = Q_2^\top Q_1^\top Q_1 Q_2 = Q_2^\top Q_2 = I$. (iv) $\|Q_1 y\|_2^2 = y^\top Q_1^\top Q_1 y = \|y\|_2^2$. □

Theorem 3.12 (QR decomposition)**.** *Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. There exist an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ and an upper triangular matrix $R_1 \in \mathbb{R}^{n\times n}$ such that*

$$A = Q\begin{pmatrix} R_1 \\ 0 \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}\begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1,$$

where $Q_1 \in \mathbb{R}^{m\times n}$ consists of the first n columns of Q (and therefore has orthonormal columns) and $Q_2 \in \mathbb{R}^{m\times(m-n)}$ of the remaining ones. If $\operatorname{rank} A = n$, then R_1 is regular, and Q_1 and R_1 are uniquely determined by the additional requirement that the diagonal entries of R_1 be positive.

The factorisation $A = Q_1R_1$ is called the **reduced** (or economy-size) QR decomposition. If $\operatorname{rank} A = n$, the columns of Q_1 form an orthonormal basis of $\operatorname{Im} A$ (as R_1 is regular, $\operatorname{Im} A = \operatorname{Im} Q_1$) and those of Q_2 one of $(\operatorname{Im} A)^\perp = \ker A^\top$, see (3.1); the uniqueness statement concerns Q_1 and R_1 only, since any orthonormal basis of $\ker A^\top$ may serve as Q_2. Existence is proved constructively by Householder reflections in Section 3.5 (Proposition 3.23); for $\operatorname{rank} A = n$ it also follows from Gram–Schmidt orthogonalisation (Section 3.4). We prove the remaining statements here.

Proof of the regularity statement. Let $\operatorname{rank} A = n$ and $A = Q_1R_1$ with Q_1 having orthonormal columns and R_1 upper triangular. Since $n = \operatorname{rank}(Q_1R_1) \leq \operatorname{rank} R_1 \leq n$, the matrix R_1 is regular. □

Proof of the uniqueness statement.* Let $\operatorname{rank} A = n$, and let $A = Q_1R_1 = \hat{Q}_1\hat{R}_1$ be two such factorisations with positive diagonals (both R_1 and $\hat{R}_1$ are regular by the first part). Then $M := \hat{R}_1R_1^{-1}$ is upper triangular with positive diagonal entries $\hat{r}_{ii}/r_{ii}$ (Lemma 2.8), and $Q_1R_1 = \hat{Q}_1\hat{R}_1$ gives $Q_1 = \hat{Q}_1M$. Hence

$$I = Q_1^\top Q_1 = M^\top \hat{Q}_1^\top \hat{Q}_1 M = M^\top M,$$

so $M^{-1} = M^\top$ is lower triangular; since M^{-1} is also upper triangular (Lemma 2.8(iii)), M^{-1} and therefore M are diagonal. A diagonal matrix with $M^\top M = I$ has diagonal entries ± 1, and since they are positive, $M = I$. Thus $\hat{R}_1 = R_1$ and $\hat{Q}_1 = Q_1$. □

Theorem 3.13 (least squares via QR)**.** *Let $A \in \mathbb{R}^{m\times n}$ with $\operatorname{rank} A = n$ and $A = QR = Q_1R_1$ as in Theorem 3.12. Then the least squares solution of $Ax = b$ is the solution of the triangular system*

$$R_1x^* = Q_1^\top b, \qquad \text{and} \qquad \min_{x\in\mathbb{R}^n} \|Ax - b\|_2 = \|Q_2^\top b\|_2.$$

Proof. Since $Q^\top$ is orthogonal (Lemma 3.11(i)), it preserves the Euclidean norm, so for every $x \in \mathbb{R}^n$

$$\|Ax - b\|_2^2 = \|Q^\top(Ax - b)\|_2^2 = \left\| \begin{pmatrix} R_1x \\ 0 \end{pmatrix} - \begin{pmatrix} Q_1^\top b \\ Q_2^\top b \end{pmatrix} \right\|_2^2 = \|R_1x - Q_1^\top b\|_2^2 + \|Q_2^\top b\|_2^2.$$

The second term does not depend on x, and the first term is nonnegative and vanishes exactly for $x = R_1^{-1}Q_1^\top b$ (R_1 is regular by Theorem 3.12). Hence this x is the unique minimiser, with minimal value $\|Q_2^\top b\|_2$. □

Remark 3.14 (no loss of conditioning). With $A = Q_1R_1$ and $\operatorname{rank} A = n$ we have $\operatorname{cond}_2(R_1) = \operatorname{cond}_2(A)$. Indeed, $A^\top A = R_1^\top Q_1^\top Q_1 R_1 = R_1^\top R_1$, so A and R_1 have the same singular values by Definition 3.6 (the eigenvalues of $A^\top A = R_1^\top R_1$), hence $\sigma_{\max}(A) = \sigma_{\max}(R_1)$, $\sigma_{\min}(A) = \sigma_{\min}(R_1)$ and $\operatorname{cond}_2(R_1) = \operatorname{cond}_2(A)$. In contrast to the normal equations, the triangular system of Theorem 3.13 is therefore conditioned exactly like the matrix A itself (this is a statement

about the matrix of the system; the conditioning of the least squares problem as a whole also depends on the residual, see Remark 3.8). The QR approach is the standard method for full-rank least squares problems; `numpy.linalg.lstsq` uses the singular value decomposition (Chapter 4), which also handles rank-deficient problems.

For the matrix and the right-hand side of Example 3.5, Gram–Schmidt orthogonalisation of the two columns (Section 3.4; one checks directly that $Q_1^\top Q_1 = I$ and $A = Q_1 R_1$) gives the reduced QR decomposition

$$A = Q_1 R_1, \qquad Q_1 = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{6} \\ 1/\sqrt{2} & 1/\sqrt{6} \\ 0 & 2/\sqrt{6} \end{pmatrix}, \qquad R_1 = \begin{pmatrix} \sqrt{2} & 1/\sqrt{2} \\ 0 & \sqrt{3/2} \end{pmatrix},$$

and $Q_2 = \frac{1}{\sqrt{3}}(1,-1,1)^\top$ completes Q_1 to an orthogonal matrix Q. Then $Q_1^\top b = (3/\sqrt{2},\, 7/\sqrt{6})^\top$, and backward substitution for $R_1 x = Q_1^\top b$ gives $x_2 = 7/3$ and $x_1 = \big(3/\sqrt{2} - 7/(3\sqrt{2})\big)/\sqrt{2} = 1/3$, the solution found from the normal equations in Example 3.5; moreover $\|Q_2^\top b\|_2 = 2/\sqrt{3} = \|r\|_2$ is the minimal residual. In the proof of Theorem 3.13 only the first term $\|R_1 x - Q_1^\top b\|_2$ depends on x, and it has been made zero; the second term is the residual that no choice of x can remove. In general, Theorem 3.13 gives $Ax^* = Q_1 R_1 R_1^{-1} Q_1^\top b = Q_1 Q_1^\top b$ and $r = b - Q_1 Q_1^\top b = Q_2 Q_2^\top b$ (because $Q_1 Q_1^\top + Q_2 Q_2^\top = QQ^\top = I$): the projection matrix P of Section 3.2 is $P = Q_1 Q_1^\top$, and $Q_2 Q_2^\top$ is the orthogonal projection onto $\ker A^\top$. Here $Q_1 Q_1^\top b = (\frac{1}{3}, \frac{8}{3}, \frac{7}{3})^\top = p$ and $Q_2 Q_2^\top b = \frac{2}{3}(1,-1,1)^\top = r$, as in Example 3.5.

3.4 Gram–Schmidt orthogonalisation*

The Gram–Schmidt process of linear algebra orthogonalises the columns of A one after the other; read column by column, it produces a reduced QR decomposition of a matrix with full column rank.

Algorithm 3.15* (classical Gram–Schmidt orthogonalisation)**.** Let $a_1, \dots, a_n \in \mathbb{R}^m$ be linearly independent. For $k = 1, \dots, n$ compute

$$\tilde{q}_k := a_k - \sum_{j=1}^{k-1} (q_j^\top a_k)\, q_j, \qquad q_k := \frac{\tilde{q}_k}{\|\tilde{q}_k\|_2}, \qquad r_{jk} := q_j^\top a_k \ (j < k), \qquad r_{kk} := \|\tilde{q}_k\|_2.$$

Lemma 3.16* (Gram–Schmidt)**.** *In Algorithm 3.15 all $\tilde{q}_k$ are nonzero, so the algorithm is well defined. The vectors $q_1, \dots, q_n$ are orthonormal,* $\operatorname{span}(q_1, \dots, q_k) = \operatorname{span}(a_1, \dots, a_k)$ *for $k = 1, \dots, n$, and with $Q_1 := (q_1 \ \cdots \ q_n)$ and $R_1 := (r_{jk})$ (with $r_{jk} := 0$ for $j > k$) we have $A = Q_1 R_1$, where R_1 is upper triangular with positive diagonal. In particular, Theorem 3.12 holds for matrices of full column rank (completing $q_1, \dots, q_n$ to an orthonormal basis $q_1, \dots, q_m$ of $\mathbb{R}^m$ gives the orthogonal matrix $Q = (q_1 \ \cdots \ q_m)$).*

Proof. We show by induction on k that $\tilde{q}_k \neq 0$, that $q_1, \dots, q_k$ are orthonormal and that $\operatorname{span}(q_1, \dots, q_k) = \operatorname{span}(a_1, \dots, a_k)$. For $k = 1$: $\tilde{q}_1 = a_1 \neq 0$ by linear independence, $q_1 = a_1/\|a_1\|_2$ is a unit vector, and the spans agree. Let $k \geq 2$ and assume the statements for $k-1$. Then $\tilde{q}_k \in a_k + \operatorname{span}(q_1, \dots, q_{k-1}) = a_k + \operatorname{span}(a_1, \dots, a_{k-1})$; if $\tilde{q}_k$ were zero, a_k would lie in $\operatorname{span}(a_1, \dots, a_{k-1})$, contradicting linear independence. For $i < k$, using the orthonormality of $q_1, \dots, q_{k-1}$,

$$q_i^\top \tilde{q}_k = q_i^\top a_k - \sum_{j=1}^{k-1} (q_j^\top a_k)\, q_i^\top q_j = q_i^\top a_k - q_i^\top a_k = 0,$$

so $q_k = \tilde{q}_k/\|\tilde{q}_k\|_2$ is a unit vector orthogonal to $q_1, \dots, q_{k-1}$. Finally $q_k \in \operatorname{span}(a_1, \dots, a_k)$ by construction, and $a_k = \tilde{q}_k + \sum_{j<k}(q_j^\top a_k) q_j \in \operatorname{span}(q_1, \dots, q_k)$; since both spans have dimension k, they coincide. Rewriting the definition of $\tilde{q}_k$ as

$$a_k = \sum_{j=1}^{k-1} r_{jk}\, q_j + r_{kk}\, q_k, \qquad k = 1, \dots, n,$$

and reading this column by column gives $A = Q_1 R_1$ with $r_{kk} = \|\tilde{q}_k\|_2 > 0$. □

Remark 3.17* (numerical behaviour). In floating-point arithmetic the classical Gram–Schmidt process (CGS, Algorithm 3.15) suffers from a *loss of orthogonality* when the columns of A are nearly linearly dependent: the computed vectors $\hat{q}_k$ are not orthogonal to the earlier ones. The loss is governed by the conditioning of A. For CGS one has $\|\hat{Q}_1^\top \hat{Q}_1 - I\|_2 \le c\,u\,\operatorname{cond}_2(A)^2$ as long as $u\operatorname{cond}_2(A)^2$ is small (L. Giraud, J. Langou, M. Rozložník, 2005), and this bound is attained in examples; for the *modified* Gram–Schmidt process (MGS) described below (Algorithm 3.18) one has $\|\hat{Q}_1^\top \hat{Q}_1 - I\|_2 \le c\,u\,\operatorname{cond}_2(A)$ as long as $u\operatorname{cond}_2(A)$ is small (Å. Björck, 1967), where c denotes constants depending only on m and n. Householder's method of Section 3.5 produces a factor Q that is orthogonal to working precision regardless of the conditioning of A and is the method of choice. The experiment below illustrates the three behaviours for a 30×10 matrix with $\operatorname{cond}_2(A) = 10^8$; it measures the loss of orthogonality in the Frobenius norm, which differs from the 2-norm by a factor of at most $\sqrt{n}$ ($\|M\|_2 \le \|M\|_F \le \sqrt{n}\,\|M\|_2$ for $M \in \mathbb{R}^{n\times n}$).

Algorithm 3.18* (modified Gram–Schmidt orthogonalisation)**.** MGS subtracts the projections one after the other, each time from the already updated vector: for $k = 1, \dots, n$,

$$\tilde{q}_k \leftarrow a_k; \qquad \text{for } j = 1, \dots, k-1: \quad r_{jk} := q_j^\top \tilde{q}_k, \quad \tilde{q}_k \leftarrow \tilde{q}_k - r_{jk}\, q_j;$$
$$r_{kk} := \|\tilde{q}_k\|_2, \qquad q_k := \tilde{q}_k / r_{kk}.$$

In exact arithmetic MGS produces the same Q_1 and R_1 as CGS (in floating-point arithmetic the different order of the operations leads to the different behaviour described in Remark 3.17): by induction, $q_j^\top \tilde{q}_k = q_j^\top a_k$ for the partially updated vector, because the subtracted terms $r_{ik} q_i$, $i < j$, are orthogonal to q_j. The cost of both variants is $2mn^2 + O(mn)$ flops.

```
classical GS  : ||Q^T Q - I||_F = 5.21e-01,   ||A - QR||_F/||A||_F = 1.01e-16
modified GS   : ||Q^T Q - I||_F = 7.07e-09,   ||A - QR||_F/||A||_F = 8.81e-17
Householder   : ||Q^T Q - I||_F = 1.20e-15,   ||A - QR||_F/||A||_F = 5.66e-16
(eps * cond = 2.2e-08,  eps * cond^2 = 2.2e+00,  eps = 2u = 2.2e-16)
```

All three methods reproduce $A = QR$ to working precision. The computed Householder factor is orthogonal to working precision, and the loss of orthogonality of MGS is of the order $u\operatorname{cond}_2(A) \approx 10^{-8}$, as the bound of Remark 3.17 predicts. For CGS the quantity $u\operatorname{cond}_2(A)^2$ is of order 1 here, so the CGS bound no longer applies, and the orthogonality is lost completely: $\|\hat{Q}_1^\top \hat{Q}_1 - I\|$ is of order 1. (The values shown are those of one run; in such a sensitive experiment the first two numbers vary from platform to platform, their orders of magnitude do not.)

3.5 Householder reflections

A QR decomposition of a matrix with full column rank can be obtained by orthogonalising its columns (Gram–Schmidt orthogonalisation, the optional Section 3.4). Here we instead *triangularise* A by a sequence of orthogonal transformations, in the same spirit as Gaussian elimination but with reflections in place of the elimination matrices E_k of Lemma 2.11. In exact arithmetic both constructions give a QR decomposition; in floating-point arithmetic, however, Gram–Schmidt orthogonalisation loses the orthogonality of the computed Q_1 when

A is ill-conditioned (Remark 3.17), whereas the method of this section produces a factor Q that is orthogonal to working precision whatever the conditioning of A (Remark 3.24). It is the method used in practice.

Definition 3.19 (Householder reflection)**.** For $v \in \mathbb{R}^m \setminus \{0\}$ the **Householder reflection** is the matrix $H_v := I - 2\dfrac{vv^\top}{v^\top v} \in \mathbb{R}^{m\times m}$.

Lemma 3.20 (properties of Householder reflections)**.** *Let $v \in \mathbb{R}^m \setminus \{0\}$.*

(i) H_v is symmetric and orthogonal, and $H_v^2 = I$.

(ii) $H_v v = -v$, and $H_v w = w$ for all $w \perp v$; thus H_v is the reflection in the hyperplane $v^\perp$.

(iii) For $x \in \mathbb{R}^m$ the product $H_v x = x - 2\dfrac{v^\top x}{v^\top v} v$ can be computed with $O(m)$ flops; in computations the matrix H_v is not formed.

Proof. (i) $H_v^\top = H_v$ is clear. Moreover $H_v^2 = I - 4\frac{vv^\top}{v^\top v} + 4\frac{v(v^\top v)v^\top}{(v^\top v)^2} = I$, hence $H_v^\top H_v = H_v^2 = I$. (ii) $H_v v = v - 2v\frac{v^\top v}{v^\top v} = -v$, and for $w \perp v$ the term $v^\top w$ vanishes, so $H_v w = w$. (iii) is immediate from the definition. □

Lemma 3.21 (choice of the reflection)**.** *Let $x \in \mathbb{R}^m \setminus \{0\}$, let $\operatorname{sign}(x_1) \in \{\pm 1\}$ denote the sign of x_1 with the convention $\operatorname{sign}(0) = 1$, and set $\alpha := -\operatorname{sign}(x_1)\|x\|_2$ and $v := x - \alpha e_1$. Then $v \neq 0$ and $H_v x = \alpha e_1 = -\operatorname{sign}(x_1)\|x\|_2\, e_1$.*

Proof. The first component of v is $v_1 = x_1 - \alpha = x_1 + \operatorname{sign}(x_1)\|x\|_2$, so $|v_1| = |x_1| + \|x\|_2 > 0$ and $v \neq 0$. Since $|\alpha| = \|x\|_2$,

$$v^\top v = \|x\|_2^2 - 2\alpha x_1 + \alpha^2 = 2\big(\|x\|_2^2 - \alpha x_1\big), \qquad v^\top x = \|x\|_2^2 - \alpha x_1,$$

hence $2\,v^\top x / v^\top v = 1$ and $H_v x = x - v = \alpha e_1$ by Lemma 3.20(iii). □

The sign of α is chosen so that x_1 and $-\alpha = \operatorname{sign}(x_1)\|x\|_2$ have the same sign: the first component $v_1 = x_1 + \operatorname{sign}(x_1)\|x\|_2$ of v is then computed *without cancellation* (compare Example 1.5). For the same reason the denominator $v^\top v = 2\big(\|x\|_2^2 + |x_1|\,\|x\|_2\big)$ is a sum of nonnegative terms (the first one positive), computed without cancellation as well. (A library implementation additionally scales x to avoid overflow and underflow in $\|x\|_2$; the illustrative code below does not.) For instance, $x = (3,4)^\top$ gives $\|x\|_2 = 5$, $\alpha = -5$, $v = (8,4)^\top$ and

$$H_v = I - \frac{2}{80}\begin{pmatrix} 64 & 32 \\ 32 & 16 \end{pmatrix} = \frac{1}{5}\begin{pmatrix} -3 & -4 \\ -4 & 3 \end{pmatrix}, \qquad H_v x = \begin{pmatrix} -5 \\ 0 \end{pmatrix} = \alpha e_1.$$

Algorithm 3.22 (Householder QR decomposition)**.** Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, and set $A^{(1)} := A$. For $k = 1, \dots, \min(n, m-1)$ let $x^{(k)} := \big(a_{kk}^{(k)}, a_{k+1,k}^{(k)}, \dots, a_{mk}^{(k)}\big)^\top \in \mathbb{R}^{m-k+1}$ be the part of column k of $A^{(k)}$ on and below the diagonal. If $x^{(k)} = 0$, set $H_k := I$ and $A^{(k+1)} := A^{(k)}$. Otherwise choose $v_k \in \mathbb{R}^{m-k+1}$ for $x^{(k)}$ according to Lemma 3.21 and set

$$H_k := \begin{pmatrix} I_{k-1} & 0 \\ 0 & H_{v_k} \end{pmatrix} \in \mathbb{R}^{m\times m}, \qquad A^{(k+1)} := H_k A^{(k)},$$

which amounts to applying H_{v_k} to the rows $k, \dots, m$ of the columns $k, \dots, n$.

Proposition 3.23. *Let $p := \min(n, m-1)$. In Algorithm 3.22 every H_k is symmetric and orthogonal, the first $k-1$ rows of $A^{(k)}$ are not changed in step k, and $A^{(k+1)}$ has zeros below the diagonal in the columns $1, \dots, k$. Consequently*

$$H_p \cdots H_1 A = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}, \qquad Q := H_1 \cdots H_p,$$

with $R_1 \in \mathbb{R}^{n \times n}$ upper triangular and Q orthogonal, so $A = Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$. In particular, Theorem 3.12 holds for every $A \in \mathbb{R}^{m \times n}$ with $m \geq n$.

Proof. H_k is symmetric and orthogonal because I_{k-1} and H_{v_k} are (Lemma 3.20(i)), or because $H_k = I$. Since H_k acts as the identity on the first $k-1$ rows, these are not changed. We show by induction on k that $A^{(k+1)}$ has zeros below the diagonal in the columns $1, \dots, k$. For the columns $j < k$ this holds by the induction hypothesis for the rows $1, \dots, k-1$ (unchanged) and because the entries in the rows $k, \dots, m$ of these columns are zero and H_{v_k} maps the zero vector to itself. For column k, the rows $k, \dots, m$ of $A^{(k+1)}$ contain $H_{v_k} x^{(k)} = \alpha_k e_1$ by Lemma 3.21 (or $x^{(k)} = 0$ if $H_k = I$), which has zeros below its first component. After p steps the matrix $H_p \cdots H_1 A$ has zeros below the diagonal in the columns $1, \dots, p$; if $p = n$ this means that it has the form $\begin{pmatrix} R_1 \\ 0 \end{pmatrix}$, and if $p = m - 1 < n$ (so $m = n$) the last column needs no reflection, since its only entry on or below the diagonal is $a_{mm}^{(m)}$. The matrix $Q = H_1 \cdots H_p$ is orthogonal as a product of orthogonal matrices (Lemma 3.11(iii)), and $Q^\top = H_p \cdots H_1$ since the H_k are symmetric; hence $Q^\top A = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$, that is, $A = Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$. □

Note the order of the factors: $Q = H_1 \cdots H_p$, but $Q^\top = H_p \cdots H_1$, so applying H_1 first and H_p last to a vector y produces $H_p \cdots H_1 y = Q^\top y$. For a least squares problem with $\operatorname{rank} A = n$ there is no need to form Q: the reflections $H_1, \dots, H_p$ are applied to b as well (in this order), which gives $Q^\top b$, and then the regular triangular system $R_1 x = (Q^\top b)_{1:n}$ is solved; the norm of the remaining components $(Q^\top b)_{n+1:m} = Q_2^\top b$ is the minimal residual (Theorem 3.13). The full column rank is needed here: if R_1 is singular, the system $R_1 x = (Q^\top b)_{1:n}$ need not be solvable, and $\|Q_2^\top b\|_2$ is then not the minimal residual. The case $x^{(k)} = 0$ had to be included in the algorithm because no rank assumption is made; it occurs only for rank-deficient A.

Remark 3.24 (cost and stability). The Householder QR decomposition costs $2mn^2 - \frac{2}{3}n^3 + O(mn)$ flops for R_1 (for $m = n$ about $\frac{4}{3}n^3$, twice the cost of Gaussian elimination); applying the reflections to b costs only $O(mn)$ more. Forming Q_1 explicitly costs about as much again, and the full Q about $4(m^2 n - mn^2 + \frac{1}{3}n^3)$ flops; neither is needed for solving least squares problems. The method is backward stable: the computed $\hat{R}_1$ is the exact triangular factor of a slightly perturbed matrix, $A + \Delta A = \tilde{Q} \begin{pmatrix} \hat{R}_1 \\ 0 \end{pmatrix}$ with some exactly orthogonal matrix $\tilde{Q}$ (a theoretical factor, not the computed $\hat{Q}$) and $\|\Delta A\|_F \leq c\, mn\, u\, \|A\|_F + O(u^2)$ for a constant c of moderate size (Higham, *Accuracy and Stability of Numerical Algorithms*, Theorem 19.4). If Q is nevertheless formed from the reflections, the computed $\hat{Q}$ differs from an exactly orthogonal matrix by at most $c'u$ in the 2-norm, with c' depending only on m and n (Higham, loc. cit., Section 19.3), so that $\|\hat{Q}^\top \hat{Q} - I\|_2 \leq 2c'u + O(u^2)$ independently of $\operatorname{cond}_2(A)$; this is the precise meaning of "orthogonal to working precision" above. This is the method behind `numpy.linalg.qr` (LAPACK routine `geqrf`).

Example 3.25 (Householder QR by hand)**.** Let $A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & -\sqrt{2} & \sqrt{2}/2 \\ 0 & \sqrt{2} & 5/\sqrt{2} \end{pmatrix}$ and $b = (3\sqrt{2}, -1, 7)^\top$.

Step 1. $x^{(1)} = (1,0,0)^\top$, $\|x^{(1)}\|_2 = 1$, $\alpha_1 = -\operatorname{sign}(1)\cdot 1 = -1$, $v_1 = x^{(1)} - \alpha_1 e_1 = (2,0,0)^\top$, so $H_1 = I - 2e_1e_1^\top = \operatorname{diag}(-1,1,1)$ and

$$A^{(2)} = H_1 A = \begin{pmatrix} -1 & -1 & -1 \\ 0 & -\sqrt{2} & \sqrt{2}/2 \\ 0 & \sqrt{2} & 5/\sqrt{2} \end{pmatrix}, \qquad H_1 b = (-3\sqrt{2}, -1, 7)^\top.$$

Step 2. $x^{(2)} = (-\sqrt{2}, \sqrt{2})^\top$, $\|x^{(2)}\|_2 = 2$, $\alpha_2 = -\operatorname{sign}(-\sqrt{2})\cdot 2 = 2$, $v_2 = x^{(2)} - 2e_1 = (-\sqrt{2}-2, \sqrt{2})^\top$ with $v_2^\top v_2 = 8 + 4\sqrt{2}$. By Lemma 3.21, $H_{v_2}x^{(2)} = (2,0)^\top$. For the remaining column, $y = (\sqrt{2}/2, 5/\sqrt{2})^\top$ gives $v_2^\top y = 4 - \sqrt{2}$ and $2\,v_2^\top y / v_2^\top v_2 = (5 - 3\sqrt{2})/2$, hence

$$H_{v_2} y = y - \frac{5-3\sqrt{2}}{2}\, v_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix},$$

and for the right-hand side, $z = (-1,7)^\top$ gives $v_2^\top z = 2 + 8\sqrt{2}$, $2\,v_2^\top z / v_2^\top v_2 = (7\sqrt{2} - 6)/2$ and $H_{v_2} z = (4\sqrt{2}, 3\sqrt{2})^\top$. (As a check: written out, $H_{v_2} = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$, and the two products can be read off directly.) Therefore

$$R = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix}, \qquad Q^\top b = \begin{pmatrix} -3\sqrt{2} \\ 4\sqrt{2} \\ 3\sqrt{2} \end{pmatrix},$$

and backward substitution for $Rx = Q^\top b$ gives $x_3 = \sqrt{2}$, $x_2 = (4\sqrt{2} - 2\sqrt{2})/2 = \sqrt{2}$ and, from $-x_1 - x_2 - x_3 = -3\sqrt{2}$, $x_1 = 3\sqrt{2} - \sqrt{2} - \sqrt{2} = \sqrt{2}$, that is, $x = (\sqrt{2}, \sqrt{2}, \sqrt{2})^\top$ (indeed $A(\sqrt{2},\sqrt{2},\sqrt{2})^\top = (3\sqrt{2},\, -2+1,\, 2+5)^\top = b$). Since the system is square and regular, the residual is zero (there is no block Q_2); for an overdetermined system the last $m-n$ components of $Q^\top b$ would remain and give the minimal residual. Note also that $r_{11} = -1 < 0$: Householder's sign choice does not produce the positive diagonal of Theorem 3.12; multiplying the first row of R and the first column of Q by -1 does. The output below reproduces R and x and reports the Frobenius norms of $A - QR$ and of $Q^\top Q - I$ for the computed factors.

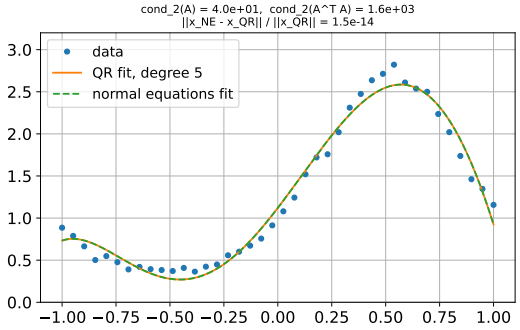
```
R =
 [[-1. -1. -1.]
 [ 0.  2.  2.]
 [ 0.  0.  3.]]
||A - QR||_F = 5.438959822042073e-16   ||Q^T Q - I||_F = 1.434936932798653e-17
solution of Ax = b: [1.4142 1.4142 1.4142]   residual: 4.440892098500626e-16
```

Remark 3.26* (Givens rotations). Plane rotations in the coordinates (i,j), called *Givens rotations*, zero one entry at a time and are an alternative to reflections. They are useful when only a few entries have to be eliminated, for instance for updating a QR decomposition when a row is appended to A, and for some sparse or structured matrices (such as upper Hessenberg matrices, Chapter 5).

3.6 Application: polynomial regression and the choice of basis*

Fitting a polynomial $p(t) = \sum_{j=0}^{d} x_j t^j$ to data (t_i, y_i), $i = 1,\dots,m$, in the least squares sense leads to the **Vandermonde matrix** $A = (t_i^j)_{i,j} \in \mathbb{R}^{m\times(d+1)}$. It has full column rank as soon as $d+1$ of the nodes are distinct, because a nonzero polynomial of degree at most d has at most d roots. Its condition number grows exponentially with the degree d (for the square case see W. Gautschi, 1975); for the 40 equidistant nodes in $[-1,1]$ of the experiment below it grows roughly by a factor 2.5 per degree. The experiment fits polynomials of increasing degree to 40

noisy samples of a smooth function on $[-1, 1]$, once via Householder QR with column pivoting (LAPACK routine `gelsy`) and once via the normal equations with Cholesky decomposition, and reports $\mathrm{cond}_2(A)$, $\mathrm{cond}_2(A^\top A)$ and the relative difference between the two coefficient vectors.



The figure shows the fits of degree 5, where the two coefficient vectors still agree to about 14 digits. For increasing degree the relative difference between the coefficients obtained from the normal equations and from QR grows rapidly, at a rate consistent with the squared condition number of the normal equations (Proposition 3.7 and Theorem 2.5); the table lists the values of one run (to the precision shown, the condition numbers are reproducible; the differences and the degree of the first Cholesky failure vary somewhat with the platform). A difference between two computed solutions is not by itself an error; without a reference solution it does not tell which of the two is wrong. For the experiment, the exact least squares solution of the stored floating-point data (computed in rational arithmetic) shows that the QR coefficients have a relative error below 10^{-8} for all degrees of the table (about 10^{-9} at degrees 21 to 23, in line with $u\,\mathrm{cond}_2(A)$); the differences listed are therefore, to the digits shown, the errors of the coefficients obtained from the normal equations.

degree d	5	12	18	20	21	22	23
$\mathrm{cond}_2(A)$	$4 \cdot 10^1$	$2 \cdot 10^4$	$4 \cdot 10^6$	$3 \cdot 10^7$	$7 \cdot 10^7$	$2 \cdot 10^8$	$5 \cdot 10^8$
$u\,\mathrm{cond}_2(A)^2$	$2 \cdot 10^{-13}$	$3 \cdot 10^{-8}$	$2 \cdot 10^{-3}$	$8 \cdot 10^{-2}$	0.6	4	30
$\|x_{\mathrm{NE}} - x_{\mathrm{QR}}\|_2 / \|x_{\mathrm{QR}}\|_2$	$2 \cdot 10^{-14}$	$3 \cdot 10^{-9}$	$1 \cdot 10^{-4}$	$4 \cdot 10^{-3}$	$8 \cdot 10^{-2}$	> 1	Cholesky fails

From about degree 23 on, the Cholesky decomposition of the computed matrix $A^\top A$ breaks down (a computed pivot is no longer positive), while the QR fit remains a sensible curve. The breakdown is only the visible symptom: once $u\,\mathrm{cond}_2(A)^2 > 1$, the rounding errors made in forming $A^\top A$, of size about $u\|A^\top A\|_2$ (up to a factor depending on m), exceed its smallest eigenvalue $\lambda_{\min}(A^\top A) = \sigma_{\min}(A)^2$, so the computed matrix need not be positive definite at all. In the table this is the case from degree 22 on, where the Cholesky decomposition can still run through—and then returns coefficients with relative error > 1. The accuracy of the computed coefficients must not be confused with the quality of the fit: the fit itself is meaningful only for moderate d; for larger d the polynomial follows the noise in the data (*overfitting*, see the Stochastics part of the course), however accurately its coefficients are computed. The ill-conditioning of A can be reduced by replacing the monomials by an orthogonal polynomial basis—Chebyshev polynomials (Chapter 8) or Legendre polynomials (Chapter 9): their sampled columns are not exactly orthogonal, but for the 40 nodes of the experiment the condition number stays below 10^2 up to degree 24 for the Chebyshev basis and up to degree 23 for the Legendre basis (at degree 23: 45 and 67 instead of $5 \cdot 10^8$ for the monomials). In exact arithmetic this changes only the representation of the fitted polynomial, not the polynomial itself (the least squares fit in the space of polynomials of degree at most d is unique, as A has full column rank), and it does not prevent overfitting.

3.7 Summary

- Least squares: minimise $\|Ax - b\|_2$. The solutions are characterised by the normal equations $A^\top Ax = A^\top b$; the residual is orthogonal to $\operatorname{Im} A$; the solution is unique if and only if $\ker A = \{0\}$.
- If $\operatorname{rank} A = n$, solving the normal equations squares the condition number: $\operatorname{cond}_2(A^\top A) = \operatorname{cond}_2(A)^2$.
- If $\operatorname{rank} A = n$, the QR decomposition $A = Q_1 R_1$ reduces the least squares problem to the regular triangular system $R_1 x = Q_1^\top b$ with $\operatorname{cond}_2(R_1) = \operatorname{cond}_2(A)$. Householder reflections $H_v = I - 2\,vv^\top/(v^\top v)$ with $v = x - \alpha e_1$, $\alpha = -\operatorname{sign}(x_1)\|x\|_2$, map x to αe_1 without cancellation; applied column by column they compute the decomposition (R_1 and the reflection vectors) in a backward stable way in $2mn^2 - \frac{2}{3}n^3$ flops, and $Q^\top b$ is obtained by applying them to b, without forming Q. Neither method changes the conditioning of the least squares problem itself (Remark 3.8).