# Chapter 1
# Machine Arithmetic and Related Matters

Songting Luo

Department of Mathematics
Iowa State University

MATH 561 Numerical Analysis

# Real Numbers

## binary number system

$$x \in \mathbf{R} \text{ iff } x = \pm(b_n 2^n + b_{n-1} 2^{n-1} + \cdots + b_0 + b_{-1} 2^{-1} + b_{-2} 2^{-2} + \cdots).$$

Here $n \geqslant 0$ is some integer, $b_i$ are either $0$ or $1$. In abbreviated form (familiar from the decimal number system):

$$x = \pm(b_n b_{n-1} \cdots b_0 . b_{-1} b_{-2} \cdots)_2.$$

## Remark

*Different bases correspond to different representations. E.g., decimal number system with base 10.*

# Machine Numbers: Way 1

(familiar from “scientific notation” in the decimal system)

## Floating-Point Numbers

- Denote $t$ the number of binary digits in fractional part, $s$ the number of binary digits in the exponent. Set of (real) floating-point numbers denoted as **R**$(t, s)$. Thus

$$x \in \mathbf{R}(t,s) \text{ iff } x = f \cdot 2^e,$$

  where $f$: mantissa of $x$; $e$: exponent of $x$,
  $f = \pm(.b_{-1}b_{-2}\cdots b_{-t})_2,\ e = \pm(c_{s-1}c_{s-2}\cdots c_0.)_2.$
- Largest: $(1-2^{-t})2^{2^s-1}$; Smallest: $2^{-2^s}$.
- $x$ is called normalized if $b_{-1} = 1$.
- “double” precision (64-bits, e.g., MATLAB arithmetic) uses $t = 53$ and $s = 10$. **NOTE**: **R**$(t, s)$ is a finite set !!!

# Machine Numbers: Way 2

## Fixed-Point Numbers

- When $e = 0$, i.e., fixed-point numbers are binary fractions, $x = f$ , hence $|f| < 1$.
- Requires extensive scaling and rescaling to lie in $(-1,\ 1)$. Complication.
- More binary digits $(s+t)$ for fraction $f$, more precision.

# Machine Numbers

## Rounding

- "exact" real number $x \in \mathbf{R}$, $x = \pm(\sum_{k=1}^{\infty} b_{-k} 2^{-k}) 2^{e}$.
- rounded number $x^* \in \mathbf{R}(t,s)$, $x^* = \pm(\sum_{k=1}^{t} b_{-k} 2^{-k}) 2^{e^*}$.
- Two ways of rounding:
  - Chopping:

$$x^* = chop(x),\ e^* = e,\ b^*_{-k} = b_{-k} \text{ for } k = 1, 2, \ldots, t.$$

  - Symmetric rounding: (familiar from rounding up and rounding down in decimal arithmetic)

$$x^* = rd(x),\ rd(x) = chop(x + \frac{1}{2} \cdot 2^{-t} \cdot 2^{e}).$$

# Rounding Error; Significant Digits

## Definition

If $x^*$ is an approximation to $x$, the absolute error is $|x - x^*|$, and the relative error is $\frac{|x - x^*|}{|x|}$, provided that $x \neq 0$.

## Definition

The number $x^*$ is said to approximate $x$ to $s$ significant digits (or figures) if $s$ is the largest nonnegative integer for which

$$\frac{|x - x^*|}{|x|} \leqslant 5 \times 10^{-s}.$$

# Rounding Error; Machine Precision

- $|x - chop(x)| = | \pm \sum_{k=t+1}^{\infty} b_{-k} 2^{-k}| 2^e \leqslant \sum_{k=t+1}^{\infty} 2^{-k} \cdot 2^e = 2^{-t} \cdot 2^e.$
- $|\frac{x-chop(x)}{x}| \leqslant \frac{2^{-t} \cdot 2^e}{|\pm \sum_{k=1}^{\infty} b_{-k} 2^{-k}| 2^e} \leqslant \frac{2^{-t} \cdot 2^e}{\frac{1}{2} \cdot 2^e} = 2 \cdot 2^{-t}.$
- $|\frac{x-rd(x)}{x}| \leqslant 2^{-t}.$
- machine precision (or unit roundoff):

$$eps = 2^{-t}.$$

- "double" precision: $t = 53$, $eps \approx 1.11 \times 10^{-16}$, "15–16" significant decimal digits.
- Equivallently,

$$rd(x) = x(1 + \epsilon), \ |\epsilon| \leqslant eps.$$

# Machine Arithmetic

## A Model of Machine Arithmetic; Floating-Point Arithmetic

- Machine addition, subtraction, multiplication, and division:

$$x \oplus y = fl(fl(x) + fl(y)), \;\; x \otimes y = fl(fl(x) \times fl(y))$$
$$x \ominus y = fl(fl(x) - fl(y)), \;\; x \oplus y = fl(fl(x) \div fl(y))$$

- In general

$$fl(x \circ y) = (x \circ y)(1 + \epsilon), \;\; |\epsilon| \leqslant eps.$$

- “Round input, perform exact arithmetic, round the result”

# Error Propagation; Cancellation Error

- Multiplication: $x(1+\epsilon_x)\cdot y(1+\epsilon_y) \approx x\cdot y(1+\epsilon_x+\epsilon_y)$, relative error $\epsilon_{x\cdot y} = \epsilon_x + \epsilon_y$.
- Division: $\frac{x(1+\epsilon_x)}{y(1+\epsilon_y)} \approx \frac{x}{y}(1+\epsilon_x-\epsilon_y)$, relative error $\epsilon_{x/y} = \epsilon_x - \epsilon_y$.
- Addition and subtraction: $x(1+\epsilon_x)+y(1+\epsilon_y) = (x+y)(1+\frac{x\epsilon_x+y\epsilon_y}{x+y})$, relative error $\epsilon_{x+y} = \frac{x}{x+y}\epsilon_x + \frac{y}{x+y}\epsilon_y$.
  - Cancellation error: if $|x+y|$ is arbitrary small compared to $|x|$ and $|y|$, large magnification of error.
  - Common problem: Subtraction of nearly equal numbers:

$$fl(x) = 0.d_1d_2\dots d_p\alpha_{p+1}\alpha_{p+2}\dots\alpha_k \times 10^n$$
$$fl(x) = 0.d_1d_2\dots d_p\beta_{p+1}\beta_{p+2}\dots\beta_k \times 10^n$$

gives fewer digits of significance: $fl(fl(x)-fl(y)) = 0.\sigma_{p+1}\sigma_{p+2}\dots\sigma_k \times 10^{n-p}$

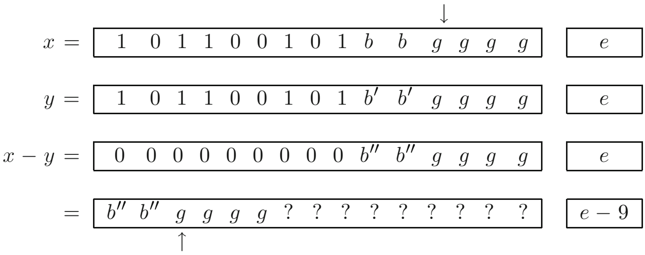
$x =$ 1 0 1 1 0 0 1 0 1 $b$ $b$ $g$ $g$ $g$ $g$ | $e$
$y =$ 1 0 1 1 0 0 1 0 1 $b'$ $b'$ $g$ $g$ $g$ $g$ | $e$
$x - y =$ 0 0 0 0 0 0 0 0 0 $b''$ $b''$ $g$ $g$ $g$ $g$ | $e$
$=$ $b''$ $b''$ $g$ $g$ $g$ $g$ ? ? ? ? ? ? ? ? ? | $e - 9$

**Fig. 1.3** The cancellation phenomenon

As before, the error in the result is a linear combination of the errors in the data, but now the coefficients are no longer ±1 but can assume values that are arbitrarily large. Note first, however, that when $x$ and $y$ have the same sign, then both coefficients are positive and bounded by 1, so that

$$|\varepsilon_{x+y}| \leq |\varepsilon_x| + |\varepsilon_y| \quad (x \cdot y > 0); \tag{1.19}$$

addition, in this case, is again a benign operation. It is only when $x$ and $y$ have opposite signs that the coefficients in (1.18) can be arbitrarily large, namely, when $|x + y|$ is arbitrarily small compared to $|x|$ and $|y|$. This happens when $x$ and $y$ are almost equal in absolute value, but opposite in sign. The large magnification of error then occurring in (1.18) is referred to as *cancellation error*. It is the only serious weakness – the Achilles heel, as it were – of numerical computation, and it should be avoided whenever possible. In particular, one should be prepared to encounter cancellation effects not only in single devastating amounts, but also repeatedly over a long period of time involving "small doses" of cancellation. Either way, the end result can be disastrous.

We illustrate the cancellation phenomenon schematically in Fig. 1.3, where $b$, $b'$, $b''$ stand for binary digits that are reliable, and the $g$ represent binary digits contaminated by error; these are often called "garbage" digits. Note in Fig. 1.3 that "garbage – garbage = garbage," but, more importantly, that the final normalization of the result moves the first garbage digit from the 12th position to the 3rd.

Cancellation is such a serious matter that we wish to give a number of elementary examples, not only of its occurrence, but also of how it might be avoided.

*Examples.* 1. An algebraic identity: $(a - b)^2 = a^2 - 2ab + b^2$. Although this is a valid identity in algebra, it is no longer valid in machine arithmetic. Thus, on a 2-decimal-digit computer, with $a = 1.8$, $b = 1.7$, we get, using symmetric rounding,

$$\text{fl}(a^2 - 2ab + b^2) = 3.2 - 6.2 + 2.9 = -0.10$$

instead of the true result 0.010, which we obtain also on our 2-digit computer if we use the left-hand side of the identity. The expanded form of the square thus produces a result which is off by one order of magnitude and on top has the wrong sign.

2. Quadratic equation: $x^2 - 56x + 1 = 0$. The usual formula for a quadratic gives, in 5-decimal arithmetic,

$$x_1 = 28 - \sqrt{783} = 28 - 27.982 = 0.018000,$$

$$x_2 = 28 + \sqrt{783} = 28 + 27.982 = 55.982.$$

   This should be contrasted with the exact roots 0.0178628… and 55.982137… . As can be seen, the smaller of the two is obtained to only two correct decimal digits, owing to cancellation. An easy way out, of course, is to compute $x_2$ first, which involves a benign addition, and then to compute $x_1 = 1/x_2$ by Vieta's formula, which again involves a benign operation – division. In this way we obtain both roots to full machine accuracy.
3. Compute $y = \sqrt{x+\delta} - \sqrt{x}$, where $x > 0$ and $|\delta|$ is very small. Clearly, the formula as written causes severe cancellation errors, since each square root has to be rounded. Writing instead

$$y = \frac{\delta}{\sqrt{x+\delta} + \sqrt{x}}$$

   completely removes the problem.
4. Compute $y = \cos(x + \delta) - \cos x$, where $|\delta|$ is very small. Here cancellation can be avoided by writing $y$ in the equivalent form

$$y = -2 \sin\frac{\delta}{2} \sin\left(x + \frac{\delta}{2}\right).$$

5. Compute $y = f(x+\delta) - f(x)$, where $|\delta|$ is very small and $f$ a given function. Special tricks, such as those used in the two preceding examples, can no longer be played, but if $f$ is sufficiently smooth in the neighborhood of $x$, we can use Taylor expansion:

$$y = f'(x)\delta + \frac{1}{2} f''(x)\delta^2 + \cdots .$$

   The terms in this series decrease rapidly when $|\delta|$ is small so that cancellation is no longer a problem.

Addition is an example of a potentially ill-conditioned function (of two variables). It naturally leads us to study the condition of more general functions.

## 1.3 The Condition of a Problem

A problem typically has an input and an output. The input consists of a set of data, say, the coefficients of some equation, and the output of another set of numbers uniquely determined by the input, say, all the roots of the equation in some prescribed order. If we collect the input in a vector $\boldsymbol{x} \in \mathbb{R}^m$ (assuming the data

# Condition Number of a Problem

Consider a problem as map $\mathbf{f}: \mathbf{R}^m \to \mathbf{R}^n$, $\mathbf{y} = (\mathbf{x})$. How sensitive is $\mathbf{f}$ to small perturbation of $\mathbf{x}$?

## Condition Numbers of f at x

- $m = 1, n = 1$. $\frac{\Delta y}{\Delta x}/\frac{y}{x} \approx \frac{xf'(x)}{f(x)}$; (relative) condition number of $f$ at $x$:

$$(cond\ f)(x) = |\frac{xf'(x)}{f(x)}|.$$

- In general. (relative) condition number of $\mathbf{f}$ at $\mathbf{x}$:

$$(cond\ \mathbf{f})(\mathbf{x}) = ||\Gamma(\mathbf{x})||,\ \Gamma(\mathbf{x}) = [\gamma_{\nu\mu}(\mathbf{x})],$$

  where $\gamma_{\nu\mu}(\mathbf{x}) = (cond_{\nu\mu}\ \mathbf{f})(\mathbf{x}) = |\frac{x_\mu \frac{\partial f_\nu}{\partial x_\mu}}{f_\nu(\mathbf{x})}|$.

- E.g., infinity norm: $(cond\ \mathbf{f})(\mathbf{x}) = \frac{||\mathbf{x}||_\infty ||\partial \mathbf{f}/\partial \mathbf{x}||_\infty}{||\mathbf{f}(\mathbf{x})||_\infty}$.

of vastly different magnitudes, then $\|\boldsymbol{x}\|_\infty$ is simply equal to the largest of these components, and all the others are ignored. For this reason, some caution is required when using (1.35).

To give an example, consider

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \dfrac{1}{x_1} + \dfrac{1}{x_2} \\ \dfrac{1}{x_1} - \dfrac{1}{x_2} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

The components of the condition matrix $\boldsymbol{\Gamma}(\boldsymbol{x})$ in (1.27) are then

$$\gamma_{11} = \left|\frac{x_2}{x_1 + x_2}\right|, \ \gamma_{12} = \left|\frac{x_1}{x_1 + x_2}\right|, \ \gamma_{21} = \left|\frac{x_2}{x_2 - x_1}\right|, \ \gamma_{22} = \left|\frac{x_1}{x_2 - x_1}\right|,$$

indicating ill-conditioning if either $x_1 \approx x_2$ or $x_1 \approx -x_2$ and $|x_1|$ (hence also $|x_2|$) is not small. The global condition number (1.35), on the other hand, since

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}) = -\frac{1}{x_1^2 x_2^2} \begin{bmatrix} x_2^2 & x_1^2 \\ x_2^2 & -x_1^2 \end{bmatrix},$$

becomes, when $L_1$ vector and matrix norms are used (cf. Ex. 33),

$$(\text{cond}\, \boldsymbol{f})(\boldsymbol{x}) = \frac{\|\boldsymbol{x}\|_1 \cdot \dfrac{2}{x_1^2 x_2^2} \max(x_1^2,\ x_2^2)}{\dfrac{1}{|x_1 x_2|}(|x_1 + x_2| + |x_1 - x_2|)} = 2\frac{|x_1| + |x_2|}{|x_1 x_2|} \frac{\max(x_1^2,\ x_2^2)}{|x_1 + x_2| + |x_1 - x_2|}.$$

Here $x_1 \approx x_2$ or $x_1 \approx -x_2$ yields $(\text{cond}\, \boldsymbol{f})(\boldsymbol{x}) \approx 2$, which is obviously misleading.

### 1.3.2 Examples

We illustrate the idea of numerical condition in a number of examples, some of which are of considerable interest in applications.

1. Compute $I_n = \displaystyle\int_0^1 \frac{t^n}{t+5}\mathrm{d}t$ for some fixed integer $n \geq 1$. As it stands, the example here deals with a map from the integers to reals and therefore does

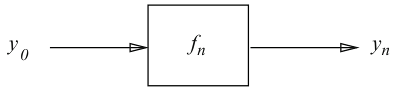

**Fig. 1.5** *Black box* for recursion (1.38)

not fit our concept of "problem" in (1.20). However, we propose to compute $I_n$ recursively by relating $I_k$ to $I_{k-1}$ and noting that

$$I_0 = \int_0^1 \frac{\mathrm{d}t}{t+5} = \ln(t+5)\Big|_0^1 = \ln\frac{6}{5}\,. \tag{1.36}$$

To find the recursion, observe that

$$\frac{t}{t+5} = 1 - \frac{5}{t+5}\,.$$

Thus, multiplying both sides by $t^{k-1}$ and integrating from 0 to 1 yields

$$I_k = -5I_{k-1} + \frac{1}{k}, \quad k = 1, 2, \ldots, n. \tag{1.37}$$

We see that $I_k$ is a solution of the (linear, inhomogeneous, first-order) difference equation

$$y_k = -5y_{k-1} + \frac{1}{k}, \quad k = 1, 2, 3, \ldots\,. \tag{1.38}$$

We now have what appears to be a practical scheme to compute $I_n$: start with $y_0 = I_0$ given by (1.36) and then apply in succession (1.38) for $k = 1, 2, \ldots, n$; then $y_n = I_n$. Recursion (1.38), for any starting value $y_0$, defines a function,

$$y_n = f_n(y_0). \tag{1.39}$$

We have the black box in Fig. 1.5 and thus a problem $f_n : \mathbb{R} \to \mathbb{R}$. (Here $n$ is a parameter.) We are interested in the condition of $f_n$ at the point $y_0 = I_0$ given by (1.36). Indeed, $I_0$ in (1.36) is not machine representable and must be rounded to $I_0^*$ before recursion (1.38) can be employed. Even if no further errors are introduced during the recursion, the final result will not be exactly $I_n$, but some approximation $I_n^* = f_n(I_0^*)$, and we have, at least approximately (actually exactly; see the remark after (1.46)),

$$\left|\frac{I_n^* - I_n}{I_n}\right| = (\mathrm{cond}\, f_n)(I_0)\left|\frac{I_0^* - I_0}{I_0}\right|. \tag{1.40}$$

To compute the condition number, note that $f_n$ is a linear function of $y_0$. Indeed, if $n = 1$, then

$$y_1 = f_1(y_0) = -5y_0 + 1.$$

If $n = 2$, then

$$y_2 = f_2(y_0) = -5y_1 + \frac{1}{2} = (-5)^2 y_0 - 5 + \frac{1}{2},$$

and so on. In general,

$$y_n = f_n(y_0) = (-5)^n y_0 + p_n,$$

where $p_n$ is some number (independent of $y_0$). There follows

$$(\text{cond}\, f_n)(y_0) = \left|\frac{y_0 f_n'(y_0)}{y_n}\right| = \left|\frac{y_0(-5)^n}{y_n}\right|. \tag{1.41}$$

Now, if $y_0 = I_0$, then $y_n = I_n$, and from the definition of $I_n$ as an integral it is clear that $I_n$ decreases monotonically in $n$ (and indeed converges monotonically to zero as $n \to \infty$). Therefore,

$$(\text{cond}\, f_n)(I_0) = \frac{I_0 \cdot 5^n}{I_n} > \frac{I_0 \cdot 5^n}{I_0} = 5^n. \tag{1.42}$$

We see that $f_n(y_0)$ is severely ill-conditioned at $y_0 = I_0$, the more so the larger $n$.

We could have anticipated this result by just looking at the recursion (1.38): we keep multiplying by (–5), which tends to make things bigger, whereas they should get smaller. Thus, there will be continuous cancellation occurring throughout the recursion.

How can we avoid this ill-conditioning? The clue comes from the remark just made: instead of multiplying by a large number, we would prefer dividing by a large number, especially if the results get bigger at the same time. This is accomplished by reversing recurrence (1.38), that is, by choosing an $\nu > n$ and computing

$$y_{k-1} = \frac{1}{5}\left(\frac{1}{k} - y_k\right), \quad k = \nu, \nu - 1, \ldots, n + 1. \tag{1.43}$$

The problem then, of course, is how to compute the starting value $y_\nu$. Before we deal with this, let us observe that we now have a new black box, as shown in Fig. 1.6.

As before, the function involved, $g_n$, is a linear function of $y_\nu$, and an argument similar to the one leading to (1.41) then gives

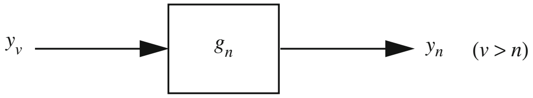

**Fig. 1.6** *Black box* for the recursion (1.43)

$$(\operatorname{cond} g_n)(y_\nu) = \left| \frac{y_\nu \left(-\frac{1}{5}\right)^{\nu-n}}{y_n} \right|, \quad \nu > n. \tag{1.44}$$

For $y_\nu = I_\nu$, we get, again by the monotonicity of $I_n$,

$$(\operatorname{cond} g_n)(I_\nu) < \left(\frac{1}{5}\right)^{\nu-n}, \quad \nu > n. \tag{1.45}$$

In analogy to (1.40), we now have

$$\left| \frac{I_n^* - I_n}{I_n} \right| = (\operatorname{cond} g_n)(I_\nu) \left| \frac{I_\nu^* - I_\nu}{I_\nu} \right| < \left(\frac{1}{5}\right)^{\nu-n} \left| \frac{I_\nu^* - I_\nu}{I_\nu} \right|, \tag{1.46}$$

where $I_\nu^*$ is some approximation of $I_\nu$. Actually, $I_\nu^*$ does not even have to be close to $I_\nu$ for (1.46) to hold, since the function $g_n$ is linear. Thus, we may take $I_\nu^* = 0$, committing a 100% error in the starting value, yet obtaining $I_n^*$ with a relative error

$$\left| \frac{I_n^* - I_n}{I_n} \right| < \left(\frac{1}{5}\right)^{\nu-n}, \quad \nu > n. \tag{1.47}$$

The bound on the right can be made arbitrarily small, say, $\leq \varepsilon$, if we choose $\nu$ large enough, for example,

$$\nu \geq n + \frac{\ln \frac{1}{\varepsilon}}{\ln 5}. \tag{1.48}$$

The final procedure, therefore, is: given the desired relative accuracy $\varepsilon$, choose $\nu$ to be the smallest integer satisfying (1.48) and then compute

$$\begin{aligned} I_\nu^* &= 0, \\ I_{k-1}^* &= \frac{1}{5}\left(\frac{1}{k} - I_k^*\right), \quad k = \nu, \nu-1, \ldots, n+1. \end{aligned} \tag{1.49}$$

This will produce a sufficiently accurate $I_n^* \approx I_n$, even in the presence of rounding errors committed in (1.49): they, too, will be consistently attenuated.

Similar ideas can be applied to the more important problem of computing solutions to second-order linear recurrence relations such as those satisfied by Bessel functions and many other special functions of mathematical physics.

The procedure of backward recurrence is then closely tied up with the theory of continued fractions.

2. *Algebraic equations*: these are equations involving a polynomial of given degree $n$,

$$p(x) = 0,\ \ p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \quad a_0 \neq 0. \tag{1.50}$$

Let $\xi$ be some fixed root of the equation, which we assume to be simple,

$$p(\xi) = 0,\ \ p'(\xi) \neq 0. \tag{1.51}$$

The problem then is to find $\xi$, given $p$. The data vector $\boldsymbol{a} = [a_0, a_1, \ldots, a_{n-1}]^{\mathrm{T}} \in \mathbb{R}^n$ consists of the coefficients of the polynomial $p$, and the result is $\xi$, a real or complex number. Thus, we have

$$\boldsymbol{\xi} :\ \ \mathbb{R}^n \to \mathbb{C},\ \ \xi = \xi(a_0, a_1, \ldots, a_{n-1}). \tag{1.52}$$

What is the condition of $\boldsymbol{\xi}$? We adopt the detailed approach of (1.27) and first define

$$\gamma_\nu = (\mathrm{cond}_\nu\, \boldsymbol{\xi})(\boldsymbol{a}) = \left| \frac{a_\nu \frac{\partial \xi}{\partial a_\nu}}{\xi} \right|, \quad \nu = 0, 1, \ldots, n-1. \tag{1.53}$$

Then we take a convenient norm, say, the $L_1$ norm $\|\boldsymbol{\gamma}\|_1 := \sum_{\nu=0}^{n-1} |\gamma_\nu|$ of the vector $\boldsymbol{\gamma} = [\gamma_0, \ldots, \gamma_{n-1}]^{\mathrm{T}}$, to define

$$(\mathrm{cond}\, \boldsymbol{\xi})(\boldsymbol{a}) = \sum_{\nu=0}^{n-1} (\mathrm{cond}_\nu\, \boldsymbol{\xi})(\boldsymbol{a}). \tag{1.54}$$

To determine the partial derivative of $\xi$ with respect to $a_\nu$, observe that we have the identity

$$[\xi(a_0, a_1, \ldots, a_n)]^n + a_{n-1}[\xi(\cdots)]^{n-1} + \cdots + a_\nu[\xi(\cdots)]^\nu + \cdots + a_0 \equiv 0.$$

Differentiating this with respect to $a_\nu$, we get

$$n[\xi(a_0, a_1, \ldots, a_n)]^{n-1} \frac{\partial \xi}{\partial a_\nu} + a_{n-1}(n-1)[\xi(\cdots)]^{n-2} \frac{\partial \xi}{\partial a_\nu} + \cdots$$

$$+ a_\nu \nu [\xi(\cdots)]^{\nu-1} \frac{\partial \xi}{\partial a_\nu} + \cdots + a_1 \frac{\partial \xi}{\partial a_\nu} + [\xi(\cdots)]^\nu \equiv 0,$$

where the last term comes from differentiating the first factor in the product $a_\nu \xi^\nu$. The last identity can be written as

$$p'(\xi)\frac{\partial\xi}{\partial a_\nu} + \xi^\nu = 0.$$

Since $p'(\xi) \neq 0$, we can solve for $\partial\xi/\partial a_\nu$ and insert the result in (1.53) and (1.54) to obtain

$$(\text{cond}\,\boldsymbol{\xi})(\boldsymbol{a}) = \frac{1}{|\xi p'(\xi)|}\sum_{\nu=0}^{n-1} |a_\nu|\,|\xi|^\nu. \tag{1.55}$$

We illustrate (1.55) by considering the polynomial $p$ of degree $n$ that has the zeros $1, 2, \ldots, n$,

$$p(x) = \prod_{\nu=1}^{n}(x-\nu) = x^n + a_{n-1}x^{n-1} + \cdots + a_0. \tag{1.56}$$

This is a famous example due to J. H. Wilkinson, who discovered the ill-conditioning of some of the zeros almost by accident. If we let $\xi_\mu = \mu$, $\mu = 1, 2, \ldots, n$, it can be shown that

$$\min_\mu \text{cond}\,\xi_\mu = \text{cond}\,\xi_1 \sim n^2 \quad \text{as} \quad n \to \infty,$$

$$\max_\mu \text{cond}\,\xi_\mu \sim \frac{1}{\left(2-\sqrt{2}\right)\pi n}\left(\frac{\sqrt{2}+1}{\sqrt{2}-1}\right)^n \quad \text{as} \quad n \to \infty.$$

The worst-conditioned root is $\xi_{\mu_0}$ with $\mu_0$ the integer closest to $n/\sqrt{2}$, when $n$ is large. Its condition number grows like $(5.828\ldots)^n$, thus exponentially fast in $n$. For example, when $n = 20$, then $\text{cond}\,\xi_{\mu_0} = 0.540 \times 10^{14}$.

The example teaches us that the roots of an algebraic equation written in the form (1.50) can be extremely sensitive to small changes in the coefficients $a_\nu$. It would, therefore, be ill-advised to express every polynomial in terms of powers, as in (1.56) and (1.50). This is particularly true for characteristic polynomials of matrices. It is much better here to work with the matrices themselves and try to reduce them (by similarity transformations) to a form that allows the eigenvalues – the roots of the characteristic equation – to be read off relatively easily.

3. *Systems of linear algebraic equations*: given a nonsingular square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, and a vector $\boldsymbol{b} \in \mathbb{R}^n$, the problem now discussed is solving the system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{1.57}$$

Here the data are the elements of $\boldsymbol{A}$ and $\boldsymbol{b}$, and the result the vector $\boldsymbol{x}$. The map in question is thus $\mathbb{R}^{n^2+n} \to \mathbb{R}^n$. To simplify matters, let us assume that $\boldsymbol{A}$ is a fixed matrix not subject to change, and only the vector $\boldsymbol{b}$ is undergoing perturbations. We then have a map $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ given by

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{b}) := \boldsymbol{A}^{-1}\boldsymbol{b}.$$

It is in fact a linear map. Therefore, $\partial \boldsymbol{f}/\partial \boldsymbol{b} = \boldsymbol{A}^{-1}$, and we get, using (1.35),

$$(\operatorname{cond} \boldsymbol{f})(\boldsymbol{b}) = \frac{\|\boldsymbol{b}\|\;\|\boldsymbol{A}^{-1}\|}{\|\boldsymbol{A}^{-1}\boldsymbol{b}\|}\,, \tag{1.58}$$

where we may take any vector norm in $\mathbb{R}^n$ and associated matrix norm (cf. (1.30)). We can write (1.58) alternatively in the form

$$(\operatorname{cond} \boldsymbol{f})(\boldsymbol{b}) = \frac{\|\boldsymbol{A}\boldsymbol{x}\|\;\|\boldsymbol{A}^{-1}\|}{\|\boldsymbol{x}\|} \quad (\text{where } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}),$$

and since there is a one-to-one correspondence between $\boldsymbol{x}$ and $\boldsymbol{b}$, we find for the worst condition number

$$\max_{\substack{\boldsymbol{b}\in\mathbb{R}^n\\ \boldsymbol{b}\neq\boldsymbol{0}}} (\operatorname{cond} \boldsymbol{f})(\boldsymbol{b}) = \max_{\substack{\boldsymbol{x}\in\mathbb{R}^n\\ \boldsymbol{x}\neq\boldsymbol{0}}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\cdot\|\boldsymbol{A}^{-1}\| = \|\boldsymbol{A}\|\cdot\|\boldsymbol{A}^{-1}\|\,,$$

by definition of the norm of $\boldsymbol{A}$. The number on the far right no longer depends on the particular system (i.e., on $\boldsymbol{b}$) and is called the *condition number* of the matrix $\boldsymbol{A}$. We denote it by

$$\operatorname{cond} \boldsymbol{A} := \|\boldsymbol{A}\|\cdot\|\boldsymbol{A}^{-1}\|\,. \tag{1.59}$$

It should be clearly understood, though, that it measures the condition of a linear system with coefficient matrix $\boldsymbol{A}$, and not the condition of other quantities that may depend on $\boldsymbol{A}$, such as eigenvalues.

Although we have considered only perturbations in the right-hand vector $\boldsymbol{b}$, it turns out that the condition number in (1.59) is also relevant when perturbations in the matrix $\boldsymbol{A}$ are allowed, provided they are sufficiently small (so small, for example, that $\|\Delta\boldsymbol{A}\|\cdot\|\boldsymbol{A}^{-1}\| < 1$).

We illustrate (1.59) by several examples.

(a) Hilbert[2] matrix:

[2] David Hilbert (1862–1943) was the most prominent member of the Göttingen school of mathematics. Hilbert's fundamental contributions to almost all parts of mathematics – algebra, number theory, geometry, integral equations, calculus of variations, and foundations – and in particular the 23 now famous problems he proposed in 1900 at the International Congress of Mathematicians in Paris gave a new impetus, and new directions, to 20th-century mathematics. Hilbert is also known for his work in mathematical physics, where among other things he formulated a variationl principle for Einstein's equations in the theory of relativity.

$$\boldsymbol{H}_n = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{1.60}$$

This is clearly a symmetric matrix, and it is also positive definite. Some numerical values for the condition number of $\boldsymbol{H}_n$, computed with the Euclidean norm,[3] are shown in Table 1.1. Their rapid growth is devastating.

**Table 1.1** The condition of Hilbert matrices

| $n$ | $\text{cond}_2 \boldsymbol{H}_n$ |
|---|---|
| 10 | $1.60 \times 10^{13}$ |
| 20 | $2.45 \times 10^{28}$ |
| 40 | $7.65 \times 10^{58}$ |

A system of order $n = 10$, for example, cannot be solved with any reliability in single precision on a 14-decimal computer. Double precision will be "exhausted" by the time we reach $n = 20$. The Hilbert matrix thus is a prototype of an ill-conditioned matrix. From a result of G. Szegő it can be seen that

$$\text{cond}_2 \boldsymbol{H}_n \sim \frac{\left(\sqrt{2}+1\right)^{4n+4}}{2^{15/4}\sqrt{\pi n}} \quad \text{as } n \to \infty.$$

(b) Vandermonde[4] matrices: these are matrices of the form

[3] We have $\text{cond}_2 \boldsymbol{H}_n = \lambda_{\max}(\boldsymbol{H}_n) \cdot \lambda_{\max}(\boldsymbol{H}_n^{-1})$, where $\lambda_{\max}(\boldsymbol{A})$ denotes the largest eigenvalue of the (symmetric, positive definite) matrix $\boldsymbol{A}$. The eigenvalues of $\boldsymbol{H}_n$ and $\boldsymbol{H}_n^{-1}$ are easily computed by the Matlab routine `eig`, provided that the inverse of $\boldsymbol{H}_n$ is computed directly from well-known formulae (not by inversion); see MA 9.

[4] Alexandre Théophile Vandermonde (1735–1796), a musician by training, but through acquaintance with Fontaine turned mathematician (temporarily), and even elected to the French Academy of Sciences, produced a total of four mathematical papers within 3 years (1770–1772). Though written by a novice to mathematics, they are not without interest. The first, e.g., made notable contributions to the then emerging theory of equations. By virtue of his fourth paper, he is regarded as the founder of the theory of determinants. What today is referred to as "Vandermonde determinant," however, does not appear anywhere in his writings. As a member of the Academy, he sat in a committee (together with Lagrange, among others) that was to define the unit of length – the meter. Later in his life, he became an ardent supporter of the French revolution.

$$V_n = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_1 & t_2 & \cdots & t_n \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ t_1^{n-1} & t_2^{n-1} & \cdots & t_n^{n-1} \end{bmatrix} \in \mathbb{R}^{n\times n}, \tag{1.61}$$

where $t_1, t_2, \ldots, t_n$ are parameters, here assumed real. The condition number of these matrices, in the $\infty$-norm, has been studied at length. Here are some sample results: if the parameters are equally spaced in [–1,1], that is,

$$t_\nu = 1 - \frac{2(\nu-1)}{n-1}, \quad \nu = 1, 2, \ldots, n,$$

then

$$\operatorname{cond}_\infty V_n \sim \frac{1}{\pi} \mathrm{e}^{-\pi/4} \mathrm{e}^{n\left(\frac{\pi}{4}+\frac{1}{2}\ln 2\right)}, \quad n \to \infty.$$

Numerical values are shown in Table 1.2. They are not growing quite as fast as those for the Hilbert matrix, but still exponentially fast. Worse than exponential growth is observed if one takes harmonic numbers as parameters,

**Table 1.2** The condition of Vandermonde matrices

| $n$ | $\operatorname{cond}_\infty V_n$ |
|---|---|
| 10 | $1.36 \times 10^4$ |
| 20 | $1.05 \times 10^9$ |
| 40 | $6.93 \times 10^{18}$ |
| 80 | $3.15 \times 10^{38}$ |

$$t_\nu = \frac{1}{\nu}, \quad \nu = 1, 2, \ldots, n.$$

Then indeed

$$\operatorname{cond}_\infty V_n > n^{n+1}.$$

Fortunately, there are not many matrices occurring naturally in applications that are *that* ill-conditioned, but moderately to severely ill-conditioned matrices are no rarity in real-life applications.

## 1.4 The Condition of an Algorithm

We again assume that we are dealing with a problem $\boldsymbol{f}$ given by

$$\boldsymbol{f}: \ \mathbb{R}^m \to \mathbb{R}^n, \ \ \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}). \tag{1.62}$$

# Condition Number of a Algorithm

Consider a algorithm $A$ solving the problem $\mathbf{f}$ as a map $\mathbf{f}_A: \ \mathbf{R}^m(t,s) \to \mathbf{R}^n(t,s), \ \mathbf{y}_A = \mathbf{f}_A(\mathbf{x})$. Assume that

$$\forall \mathbf{x} \in \mathbf{R}^m(t,s), \exists \mathbf{x}_A \in \mathbf{R}^m \ s.t. \ \mathbf{f}_A(\mathbf{x}) = \mathbf{f}(\mathbf{x}_A).$$

## Condition Number of $A$ at $\mathbf{x}$

$$(cond \ A)(\mathbf{x}) = \inf_{\mathbf{x}_A} \frac{||\mathbf{x}_A - \mathbf{x}||}{||\mathbf{x}||}/eps.$$

Along with the problem $\boldsymbol{f}$, we are also given an algorithm $A$ that "solves" the problem. That is, given a machine vector $\boldsymbol{x} \in \mathbb{R}^m(t,s)$, the algorithm $A$ produces a vector $\boldsymbol{y}_A$ (in machine arithmetic) that is supposed to approximate $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$. Thus, we have another map $\boldsymbol{f}_A$ describing how the problem $\boldsymbol{f}$ is solved by the algorithm $A$,

$$\boldsymbol{f}_A : \ \mathbb{R}^m(t,s) \to \mathbb{R}^n(t,s), \quad \boldsymbol{y}_A = \boldsymbol{f}_A(\boldsymbol{x}). \tag{1.63}$$

In order to be able to analyze $\boldsymbol{f}_A$ in these general terms, we must make a basic assumption, namely, that

$$\begin{gathered} \text{for every } \ \boldsymbol{x} \in \mathbb{R}^m(t,s), \ \ \text{there holds} \\ \boldsymbol{f}_A(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_A) \ \text{ for some } \ \boldsymbol{x}_A \in \mathbb{R}^m. \end{gathered} \tag{1.64}$$

That is, the computed solution corresponding to some input $\boldsymbol{x}$ is the exact solution for some different input $\boldsymbol{x}_A$ (not necessarily a machine vector and not necessarily uniquely determined) that we hope is close to $\boldsymbol{x}$. The closer we can find an $\boldsymbol{x}_A$ to $\boldsymbol{x}$, the more confidence we should place in the algorithm $A$. We therefore define the condition of $A$ in terms of the $\boldsymbol{x}_A$ closest to $\boldsymbol{x}$ (if there is more than one), by comparing its relative error with the machine precision eps:

$$(\operatorname{cond} A)(\boldsymbol{x}) = \inf_{\boldsymbol{x}_A} \frac{\|\boldsymbol{x}_A - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} / \text{eps}. \tag{1.65}$$

Here the infimum is over all $\boldsymbol{x}_A$ satisfying $\boldsymbol{y}_A = \boldsymbol{f}(\boldsymbol{x}_A)$. In practice, one can take any such $\boldsymbol{x}_A$ and then obtain an upper bound for the condition number:

$$(\operatorname{cond} A)(\boldsymbol{x}) \le \frac{\|\boldsymbol{x}_A - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} / \text{eps}. \tag{1.66}$$

The vector norm in (1.65), respectively, (1.66), can be chosen as seems convenient.

Here are some very elementary examples.

1. Suppose a library routine for the logarithm function $y = \ln x$, for any positive machine number $x$, produces a $y_A$ satisfying $y_A = [\ln x](1+\varepsilon)$, $|\varepsilon| \le 5\,\text{eps}$. What can we say about the condition of the underlying algorithm $A$? We clearly have

   $$y_A = \ln x_A, \ \ \text{where} \ \ x_A = x^{1+\varepsilon} \ \ \text{(uniquely)}.$$

   Consequently,

   $$\left|\frac{x_A - x}{x}\right| = \left|\frac{x^{1+\varepsilon} - x}{x}\right| = |x^\varepsilon - 1| = \left|\mathrm{e}^{\varepsilon \ln x} - 1\right| \approx |\varepsilon \ln x| \le 5\,|\ln x| \cdot \text{eps},$$

and, therefore, $(\operatorname{cond} A)(x) \le 5\,|\ln x|$. The algorithm $A$ is well conditioned, except in the immediate right-hand vicinity of $x = 0$ and for $x$ very large. (In the latter case, however, $x$ is likely to overflow before $A$ becomes seriously ill-conditioned.)

2. Consider the problem

$$\boldsymbol{f}: \ \mathbb{R}^n \to \mathbb{R}, \ \ y = x_1 x_2 \cdots x_n.$$

We solve the problem by the obvious algorithm

$$A: \quad \begin{aligned} p_1 &= x_1, \\ p_k &= \mathrm{fl}(x_k p_{k-1}), \ \ k = 2, 3, \dots, n, \\ y_A &= p_n. \end{aligned}$$

Note that $x_1$ is machine representable, since for the algorithm $A$ we assume $\boldsymbol{x} \in \mathbb{R}^n(t, s)$.

Now using the basic law of machine arithmetic (cf. (1.15)), we get

$$\begin{aligned} p_1 &= x_1, \\ p_k &= x_k p_{k-1}(1 + \varepsilon_k), \ \ k = 2, 3, \dots, n, \ \ |\varepsilon_k| \le \text{eps}, \end{aligned}$$

from which

$$p_n = x_1 x_2 \cdots x_n (1 + \varepsilon_2)(1 + \varepsilon_3) \cdots (1 + \varepsilon_n).$$

Therefore, we can take for example (there is no uniqueness),

$$\boldsymbol{x}_A = [x_1, x_2(1 + \varepsilon_2), \dots, x_n(1 + \varepsilon_n)]^{\mathrm{T}}.$$

This gives, using the $\infty$-norm,

$$\frac{\|\boldsymbol{x}_A - \boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty \text{eps}} = \frac{\|[0, x_2\varepsilon_2, \dots, x_n\varepsilon_n]^{\mathrm{T}}\|_\infty}{\|\boldsymbol{x}\|_\infty \text{eps}} \le \frac{\|\boldsymbol{x}\|_\infty \text{eps}}{\|\boldsymbol{x}\|_\infty \text{eps}} = 1,$$

and so, by (1.66), $(\operatorname{cond} A)(\boldsymbol{x}) \le 1$ for any $\boldsymbol{x} \in \mathbb{R}^n(t, s)$. Our algorithm, to nobody's surprise, is perfectly well conditioned.

# Computer Solution; Overall Error

$\mathbf{x}^* = fl(\mathbf{x}),\ \mathbf{y}_A^* = \mathbf{f}_A(\mathbf{x}^*)$.

- Total error:

$$\frac{||\mathbf{y}_A^* - \mathbf{y}||}{||\mathbf{y}||} \leqslant \frac{||\mathbf{y}_A^* - \mathbf{y}^*||}{||\mathbf{y}||} + \frac{||\mathbf{y}^* - \mathbf{y}||}{||\mathbf{y}||} \approx \frac{||\mathbf{y}_A^* - \mathbf{y}^*||}{||\mathbf{y}^*||} + \frac{||\mathbf{y}^* - \mathbf{y}||}{||\mathbf{y}||}$$

- 1st term of RHS

$$\frac{||\mathbf{y}_A^* - \mathbf{y}^*||}{||\mathbf{y}^*||} = \frac{||\mathbf{f}_A(\mathbf{x}^*) - \mathbf{f}(\mathbf{x}^*)||}{||\mathbf{f}(\mathbf{x}^*)||} = \frac{||\mathbf{f}(\mathbf{x}_A^*) - \mathbf{f}(\mathbf{x}^*)||}{||\mathbf{f}(\mathbf{x}^*)||}$$

$$\leqslant (cond\ \mathbf{f})(\mathbf{x}^*) \cdot \frac{||\mathbf{x}_A^* - \mathbf{x}^*||}{||\mathbf{x}^*||} = (cond\ \mathbf{f})(\mathbf{x}^*) \cdot (cond\ A)(\mathbf{x}^*) \cdot eps.$$

- 2nd term of RHS

$$\frac{||\mathbf{y}^* - \mathbf{y}||}{||\mathbf{y}||} = \frac{||\mathbf{f}(\mathbf{x}^*) - \mathbf{f}(\mathbf{x})||}{||\mathbf{f}(\mathbf{x})||} \leqslant (cond\ \mathbf{f})(\mathbf{x}) \cdot \frac{||\mathbf{x}^* - \mathbf{x}||}{||\mathbf{x}||} = (cond\ \mathbf{f})(\mathbf{x}) \cdot \epsilon.$$

- Thus

$$\frac{||\mathbf{y}_A^* - \mathbf{y}||}{||\mathbf{y}||} \leqslant (cond\ \mathbf{f})(\mathbf{x})\{\epsilon + (cond\ A)(\mathbf{x}^*) \cdot eps\}.$$

# Taylor Polynomials

## Theorem (Taylor's Theorem)

*Suppose* $f \in C^n[a,\ b]$, *that* $f^{(n+1)}$ *exists on* $[a,\ b]$, *and* $x_0 \in [a,\ b]$. *For every* $x_0 \in [a,\ b]$, *there exists a number* $\xi(x)$ *between* $x_0$ *and* $x$ *with*

$$f(x) = P_n(x) + R_n(x),$$

*where*

$$\begin{aligned} P_n(x) =& f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 \\ & + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n = \sum_{k=0} n \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k \end{aligned}$$

*and*

$$R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{n+1}.$$