Floating-point arithmetic and error analysis (AFAE)

Large scale rounding error analysis, conditioning

Stef Graillat

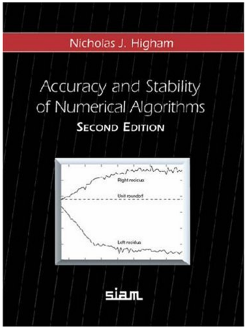
LIP6/PEQUAN - Sorbonne University

Lecture Master 2 SFPN - MAIN5

Pioneers of rounding error analysis

Alan Turing (1912-1954)   James Wilkinson (1919-1986)

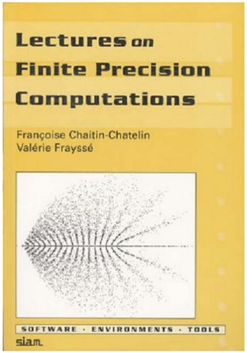
Bibliography


Nicholas J. Higham

Accuracy and Stability of Numerical Algorithms

Second Edition

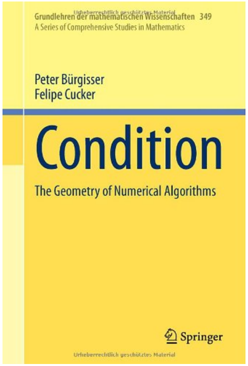
siam


Lectures on Finite Precision Computations

Françoise Chaitin-Chatelin
Valérie Frayssé

SOFTWARE · ENVIRONMENTS · TOOLS

siam


Grundlehren der mathematischen Wissenschaften 349

A Series of Comprehensive Studies in Mathematics

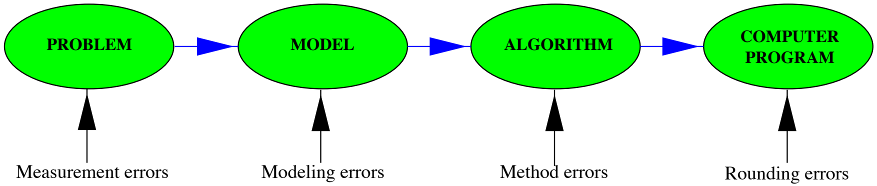
Peter Bürgisser
Felipe Cucker

Condition

The Geometry of Numerical Algorithms

Springer

Different sources of error



In this course, we will only focus on rounding errors.

In general, to obtain good results, it is needed to take into account all these steps.

But numerical algorithms should be designed by taking into account rounding errors at the beginning.

Objectives of rounding error analysis

Goal

To appreciate the quality of the approximation, i.e. to estimate or to bound the accuracy of the approximation

This analysis must also make it possible to identify the different sources of error and to measure their impact on the accuracy of the result.

This is a prerequisite for improving the quality of the approximation.

For this purpose, it is essential to distinguish the two following aspects:

- the difficulty of solving the problem,
- the reliability of the algorithm solving the problem.

Well-posed problems, ill-posed problems

We consider the following mathematical problem P:

$$(P) : \text{find } x \text{ such that } F(x) = y \text{ for a given } y$$

We say that the problem (P) is well-posed if the solution $x = F^{-1}(y)$ exists, is unique and depends continuously of the data y (which is equivalent to the fact that F^{-1} is continuous in a neighborhood of y)

If it is not the case, the problem is said to be ill-posed (ex: solve $Ax = b$ with A singular)

In this lecture, we will only deal with well-posed problems!

Absolute error, relative error

$\widehat{x} \in \mathbb{R}$ an approximation of a nonzero real number $x \in \mathbb{R}$ and $\Delta x = |x - \widehat{x}|$

2 ways to measure the accuracy of $\widehat{x}$:

absolute error:

$$E_a(\widehat{x}) = |x - \widehat{x}| = |\Delta x|,$$

relative error:

$$E_r(\widehat{x}) = \frac{|x - \widehat{x}|}{|x|} = \frac{|\Delta x|}{|x|}.$$

Si $x \in \mathbb{R}^n$

absolute error:

$$E_a(\widehat{x}) = \|x - \widehat{x}\| = \|\Delta x\|$$

relative error:

normwise: $\|\Delta x\|_g = \|\Delta x\| / \|x\|$

componentwise: $\|\Delta x\|_c = \max_i |\Delta x_i| / |x_i| \qquad x = (x_i), x_i \neq 0$

Conditioning of a problem

If problem (P) is well-posed, then $x = G(y)$ where $G = F^{-1}$
If error Δy on the data y, the computed value is $\widehat{x} = G(y + \Delta y)$.
If Δy is sufficiently small, then we have

$$\widehat{x} - x \approx G'(y) \cdot \Delta y,$$

which can be written (if $x \neq 0$ and $y \neq 0$) as

$$(\widehat{x} - x)/x \approx (yG'(y)/G(y)) \cdot (\Delta y/y).$$

This leads to

$$\frac{|\widehat{x} - x|}{|x|} = K(G, y) \cdot \frac{|\Delta y|}{|y|} + \mathcal{O}(|\Delta y|^2)$$

where

$$K(G, y) = \left| \frac{yG'(y)}{G(y)} \right|.$$

$K(G, y)$ is the the condition number of the evaluation of G en y.

Conditioning of polynomial evaluation

Let us consider the evaluation of the polynomial

$$p(z) = \sum_{i=0}^{n} a_i z^i,$$

$a_i, z \in \mathbb{R}$, when perturbing only the data $y = z$.

If $z \neq 0$, the previous relation gives use

$$K(p, z) = |zp'(z)/p(z)|$$

One can notice that the condition number goes to infinity where z converges to a root of p.

Thus the accuracy of the evaluation of a polynomial in the neighborhood of one of its roots can be arbitrarily bad.

Conditioning of polynomial evaluation (1/2)

Let us consider the problem of polynomial evaluation considering perturbation on the coefficient of the polynomial.

Let $p(z) = \sum_{i=0}^{n} a_i z^i$, with $a_i, z \in \mathbb{R}$ and $a_i \neq 0$, and $\Delta y = (\Delta a_0, \ldots, \Delta a_n)^T = \Delta a$ a perturbation of the data $y = (a_0, \ldots, a_n)^T = a$.

The result $x = p(y)$ is perturbed into $\widehat{x} = p(y + \Delta y) = \sum_{i=0}^{n} (a_i + \Delta a_i) z^i$.

So the perturbation is $\Delta x = \widehat{x} - x = \sum_{i=0}^{n} \Delta a_i z^i$.

We endow $\mathcal{D} = \mathbb{R}^{n+1}$ with the relative norm $\|\Delta a\|_{\mathcal{D}} = \max_{i=0:n} |\Delta a_i|/|a_i|$ and $\mathcal{R} = \mathbb{R}$ with the relative norm $\|\Delta x\|_{\mathcal{R}} = |\Delta x|/|x|$.

Conditioning of polynomial evaluation (2/2)

As $|\Delta x| = |\sum_{i=0}^n \Delta a_i z^i| \leq \|\Delta a\|_{\mathcal{D}} \sum_{i=0}^n |a_i||z|^i$, we have

$$\frac{\|\Delta x\|_{\mathcal{R}}}{\|\Delta y\|_{\mathcal{D}}} \leq \frac{\sum_{i=0}^n |a_i||z|^i}{|\sum_{i=0}^n a_i z^i|}.$$

The bound is attained for the perturbation Δa such that $\Delta a_i / a_i = \mathrm{sign}(a_i z^i) \|\Delta a\|_{\mathcal{D}}$.

The condition number for the evaluation of p at z is

$$K(p(z), a) = \frac{\sum_{i=0}^n |a_i z^i|}{|p(z)|}.$$

We can notice that the condition number is particularly large as we evaluate the polynomial near one of its root.

Conditioning for linear systems (1/3)

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \qquad b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

The solution of the linear system $Ax = b$ is $x = (1, 1, 1, 1)^T$.
If we perturb the system

$$A + \Delta A = \begin{pmatrix} 10 & 7 & 8.1 & 7.2 \\ 7.08 & 5.04 & 6 & 5 \\ 8 & 5.98 & 9.89 & 9 \\ 6.99 & 4.99 & 9 & 9.98 \end{pmatrix}$$

the new solution of $(A + \Delta A)x = b$ is $x = (-81, 137, -34, 22)^T$.

Conditioning for linear systems (2/3)

If x is the solution of $Ax = b$ and $\widehat{x} = x + \Delta x$ the solution of $Ax = b + \Delta b$, then

$$A\Delta x = \Delta b \quad \text{and so} \quad \Delta x = A^{-1}\Delta b$$

$$\|\Delta x\| = \|A^{-1}\Delta b\| \le \|A^{-1}\|\|\Delta b\|$$

We also have

$$\begin{aligned} \|Ax\| &= \|b\| \\ \|A\|\|x\| &\ge \|b\| \\ \frac{\|A\|}{\|b\|} &\ge \frac{1}{\|x\|} \end{aligned}$$

and so

$$\frac{\|\Delta x\|}{\|x\|} \le \|A\|\|A^{-1}\|\frac{\|\Delta b\|}{\|b\|}$$

Conditioning for linear systems (3/3)

Let us note

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

If $x + \Delta x$ is a solution of $(A + \Delta A)x = b$, then

$$\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A)\frac{\|\Delta A\|}{\|A\|}$$

and if $x + \Delta x$ is a solution of $(A + \Delta A)x = b + \Delta b$ then

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}}\left[\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right]$$

Some remarks on the condition number

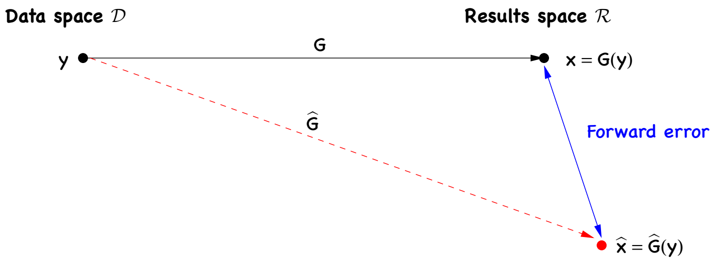
The choice of a norm modifies the condition number as well as the choice of the perturbations

In general, the condition number measures the inverse of the distance to singularity

The condition number only depends on the problem and not on the algorithm used to solve this problem

We only consider infinitesimale perturbations

Error analysis

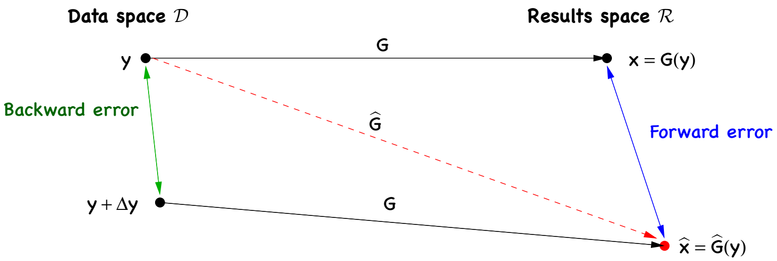


Forward error analysis

Backward error analysis

Identify $\widehat{x}$ as the solution of a perturbed problem: $\widehat{x} = G(y + \Delta y)$.

Error analysis



Forward error analysis

Backward error analysis

Identify $\widehat{x}$ as the solution of a perturbed problem: $\widehat{x} = G(y + \Delta y)$.

Forward error analysis

Forward analysis considers the execution of the algorithm $\widehat{G}$ for the input y.

By formalizing the propagation of errors generated by the evaluation of each intermediate variable, the forward analysis provides an upper bound on the gap between the exact solution x and the computed solution $\widehat{x}$.

This gap is called forward error and corresponds to the accuracy of $\widehat{x}$.

Hence, forward analysis answers the question "To which accuracy is the problem solved?".

Tracking the propagation of these intermediate errors becomes very quickly a complicated exercise and leads to expressions that are difficult to exploit.

Backward error analysis

The backward analysis circumvents the difficulty to drive and exploit forward analysis for the computation $\widehat{x} = \widehat{G}(y)$. It proceeds in two stages.

- The approximation $\widehat{x}$ is identified with the exact evaluation of G but at a perturbed data $(y + \Delta y)$. The error Δy on the data y is the backward error associated with $\widehat{x}$; it satisfies $\widehat{x} = G(y + \Delta y)$.
  The effect of errors on x is therefore interpreted as a perturbation Δy of the data y for the calculation of G. In this first step, we answer the question "Which problem was actually solved?".
- Since the backward error Δy is estimated or bounded, the analysis of the error on x is thus reduced to a sensitivity analysis (conditioning). The second step is therefore to obtain an estimate or a (first-order) bound on the accuracy of the approximation $\widehat{x}$.

Advantages of backward error analysis

How to measure the difficulty of solving the problem ?
Condition number measures the sensitivity of the solution to perturbation in the data

Condition number : $\mathrm{cond}(P, y) := \lim_{\varepsilon \to 0} \sup_{\|\Delta y\| \leq \varepsilon} \left\{ \frac{\|\Delta x\|_{\mathcal{R}}}{\|\Delta y\|_{\mathcal{D}}} \right\}$

How to appreciate the reliability of the algorithm?
Backward error measures the distance between the problem we solved and the initial problem.

Backward error : $\eta(\widehat{x}) =$

How to estimate the accuracy of the computed solution?
At first order, the rule of thumb:

forward error $\lesssim$ condition number $\times$ backward error.

Advantages of backward error analysis

How to measure the difficulty of solving the problem ?
Condition number measures the sensitivity of the solution to perturbation in the data

Condition number : $\mathrm{cond}(P, y) := \lim_{\varepsilon \to 0} \sup_{\|\Delta y\| \leq \varepsilon} \left\{ \frac{\|\Delta x\|_{\mathcal{R}}}{\|\Delta y\|_{\mathcal{D}}} \right\}$

How to appreciate the reliability of the algorithm?
Backward error measures the distance between the problem we solved and the initial problem.

Backward error : $\eta(\widehat{x}) = \min_{\Delta y \in \mathcal{D}} \{\|\Delta y\|_{\mathcal{D}} : \widehat{x} = G(y + \Delta y)\}$

How to estimate the accuracy of the computed solution?
At first order, the rule of thumb:

forward error $\lesssim$ condition number $\times$ backward error.

Advantages of backward error analysis

How to measure the difficulty of solving the problem ?
Condition number measures the sensitivity of the solution to perturbation in the data

Condition number : $\mathrm{cond}(P, y) := \lim_{\varepsilon \to 0} \sup_{\|\Delta y\| \leq \varepsilon} \left\{ \frac{\|\Delta x\|_{\mathcal{R}}}{\|\Delta y\|_{\mathcal{D}}} \right\}$

How to appreciate the reliability of the algorithm?
Backward error measures the distance between the problem we solved and the initial problem.

Backward error : $\eta(\widehat{x}) = \min_{\Delta y \in \mathcal{D}} \{ \|\Delta y\|_{\mathcal{D}} : \widehat{x} = G(y + \Delta y) \}$

How to estimate the accuracy of the computed solution?
At first order, the rule of thumb:

forward error $\lesssim$ condition number $\times$ backward error.

Stability of algorithms and backward error

The stability of an algorithm describes the influence of finite precision on his execution. The backward analysis makes it possible to define the most commonly used notion of stability: the backward-stability of an algorithm.

The backward error associated to $\widehat{x} = \widehat{G}(y)$ is the scalar $\eta(\widehat{x})$ such that

$$\eta(\widehat{x}) = \min_{\Delta y \in \mathcal{D}} \{ \|\Delta y\|_{\mathcal{D}} : \widehat{x} = G(y + \Delta y) \}$$

Definition 1

An algorithm is backward-stable for solving the problem (P) if the computed solution $\widehat{x}$ has a small backward error $\eta(\widehat{x})$.

Standard model for floating-point arithmetic

Using floating-point numbers on computers requires to approach each real number by a floating-point number.

This is equivalent to define an application $\mathrm{fl} : \mathbb{R} \to \mathbb{F}$, $x \mapsto \mathrm{fl}(x)$ that maps a real number to a floating-point number.

This application defines a rounding from $\mathbb{R}$ to $\mathbb{F}$ if it satisfies the following properties:

$\mathrm{fl}(x) = x$ for all $x \in \mathbb{F}$;

$\mathrm{fl}(x) \leq \mathrm{fl}(y)$ for all $x, y \in \mathbb{R}$ such that $x \leq y$.

There are several ways to define a rounding $\mathrm{fl}(\cdot)$. One can choose a rounding to nearest, i.e., $\mathrm{fl}(x) = \mathrm{argmin}_{y \in \mathbb{F}} |x - y|$ or choose a directed rounding toward 0 or toward $+\infty$ or $-\infty$.

Standard model for floating-point arithmetic (1/2)

If we approximate $x \in \mathbb{R}$ by $\mathrm{fl}(x) \in \mathbb{F}$, it is important to be able to quantify the rounding error.

This is the object of the following theorem. It states that the relative error is less than the rounding unit $\mathbf{u}$ that depends precision ε and of rounding mode.

We have $\mathbf{u} = \varepsilon/2$ for rounding to nearest and $\mathbf{u} = \varepsilon$ for directed rounding.

Theorem 1

Let $x \in \mathbb{R}$ then

$$\mathrm{fl}(x) = x(1+\delta), \quad |\delta| \le \mathbf{u}.$$

Standard model for floating-point arithmetic (2/2)

Standard model

$$\mathrm{fl}(x \circ y) = (x \circ y)(1+\delta), \quad |\delta| \le \mathbf{u}, \quad \circ \in \{+, -, \cdot, /\}.$$

$$\mathrm{fl}(x \circ y) = \frac{x \circ y}{1+\delta'}, \quad |\delta'| \le \mathbf{u}.$$

With underflow

The previous standard model ignores the possibility of underflow or overflow.

To take into account possibles underflows, one can modify the standard model as follows:

$$\mathrm{fl}(x \circ y) = (x \circ y)(1+\delta) + \eta, \quad |\delta| \le \mathbf{u}, \quad |\eta| \le \underline{\mathrm{u}}.$$

One always has $\delta\eta = 0$. If there is an underflow then $\delta = 0$ else $\eta = 0$.

In double precision (`binary64`) with rounding to nearest, we have

$$\mathbf{u} = 2^{-53} \qquad \text{et} \qquad \underline{\mathrm{u}} = 2^{-1074}$$

Backward stability in finite precision

An algorithm that computes $\widehat{x}$ in finite precision ($\mathbf{u}$ unit rounding) is backward-stable if its backward error $\eta(\widehat{x})$ is of the order of $\mathbf{u}$.

A **backward-stable** algorithm calculates the exact solution of a perturbed problem and this perturbation is sufficiently small that the perturbed problem and the exact problem are hardly discernible with the precision used.

A **backward-stable** algorithm does not introduce therefore more error in the data than that the one introduced on the same data by the available precision.

Such a data error is intrinsic to the precision used. Thus, a backward-stable algorithm allows optimal use of the computer precision. Of course, this does not guarantee that the accuracy of the solution is of the order of $\mathbf{u}$.

Accuracy of the solution

The accuracy of the calculated solution generally follows the following bound:

$$\text{forward error} \lesssim \text{condition number} \times \text{backward error}.$$

The accuracy $\|\Delta x\|$ of a solution computed by an backward-stable algorithm thus satisfies the relation:

$$\|\Delta \mathbf{x}\| \lesssim K\mathbf{u}.$$

A problem is **ill-conditioned** with relative accuracy $\mathbf{u}$ if it admits a conditioning number that is of an order of magnitude greater than $1/\mathbf{u}$, that is to say, if $K \times \mathbf{u} \geq 1$.