Numerical Analysis
Theory

Christian Rossi

Academic Year 2023-2024

Abstract

The topics of the course are:

- Floating-point arithmetic: different sources of the computational error; absolute vs relative errors; the floating point representation of real numbers; the round-off unit; the machine epsilon; floating-point operations; over- and under-flow; numerical cancellation.

- Numerical approximation of nonlinear equations: the bisection and the Newton methods; the fixed-point iteration; convergence analysis (global and local results); order of convergence; stopping criteria and corresponding reliability; generalization to the system of nonlinear equations (hints).

- Numerical approximation of systems of linear equations: direct methods (Gaussian elimination method; LU and Cholesky factorizations; pivoting; sparse systems: Thomas algorithm for tridiagonal systems); iterative methods (the stationary and the dynamic Richardson scheme; Jacobi, Gauss-Seidel, gradient, conjugate gradient methods (hints); choice of the preconditioner; stopping criteria and corresponding reliability); accuracy and stability of the approximation; the condition number of a matrix; over- and under-determined systems: the singular value decomposition (hints).

- Numerical approximation of functions and data: Polynomial interpolation (Lagrange form); piecewise interpolation; cubic interpolating splines; least-squares approximation of clouds of data.

- Numerical approximation of derivatives: finite difference schemes of the first and second order; the undetermined coefficient method.

- Numerical approximation of definite integrals: simple and composite formulas; midpoint, trapezoidal, Cavalieri-Simpson quadrature rules; Gaussian formulas; degree of exactness and order of accuracy of a quadrature rule.

- Numerical approximation of ODEs: the Cauchy problem; one-step methods (forward and backward Euler and Crank-Nicolson schemes); consistency, stability, and convergence (hints).

Contents

Chapter 1

Introduction

1.1 Numerical analysis and errors

Numerical analysis is the field of mathematics dealing with methods to find the solutions of certain mathematical problems with an electronic calculator. It is the intersection between math and computer science.

On the other hand, scientific computing also deals with the model formalization and so it needs also engineering knowledge.

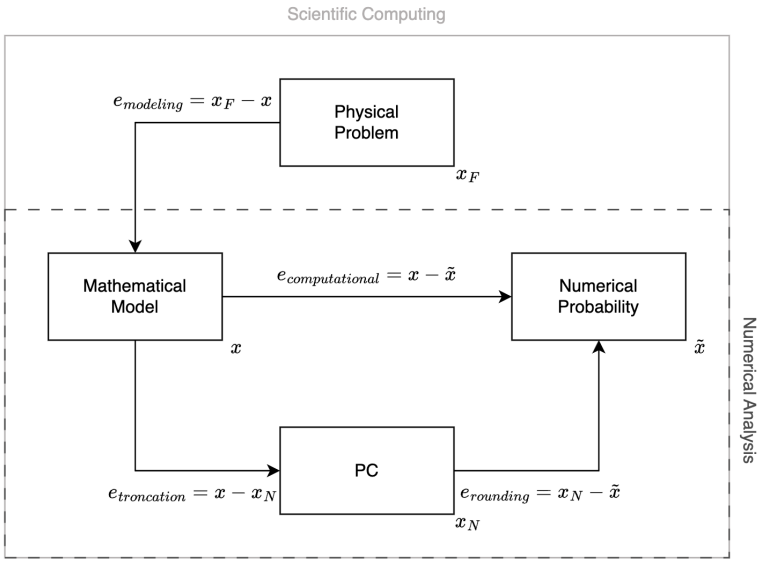


Figure 1.1: Difference between numerical analysis and scientific computing

As it is possible to see from the diagram every step of the computation have to deal with errors. The possible types of errors are:

- Absolute: $|x - \tilde{x}|$
- Relative: $\dfrac{|x - \tilde{x}|}{|x|}$, where $x \neq 0$

The relative error is more precise because it compares the error with the measure quantity.

Example : Let us consider $x = 100$ and $\tilde{x} = 100.1$. The errors in this case are:

$$e_{abs} = |x - \tilde{x}| = |100 - 100.1| = 0.1$$

$$e_{rel} = \frac{|x - \tilde{x}|}{|x|} = \frac{|100 - 100.1|}{|100|} = 0.001$$

Let us consider $x = 0.2$ and $\tilde{x} = 0.1$. The errors in this case are:

$$e_{abs} = |x - \tilde{x}| = |0.2 - 0.1| = 0.1$$

$$e_{rel} = \frac{|x - \tilde{x}|}{|x|} = \frac{|0.2 - 0.1|}{|0.2|} = 0.5$$

The result are that the measures have the same absolute error (10%), but the relative error is much grater in the second example (50% vs 0.1%). This result proves that the relative error is the most precise.

1.2 Floating point

A calculator can only handle a finite quantity of numbers and compute a finite number of operations. For those reason the set of the real numbers $\mathbb{R}$ is indeed represented by a finite set of machine numbers $\mathbb{F} = \{-\tilde{a}_{min}, \ldots, \tilde{a}_{max}\}$ called floating points numbers. The function used to find the corresponding value in $\mathbb{F}$ to a number in $\mathbb{R}$ is $fl(x)$ that does an operation called truncation and rounding.

The set $\mathbb{F} = \mathbb{F}(\beta, t, L, U)$ is characterized by four parameters β, t, L and U such that every real number $fl(x) \in \mathbb{F}$ can be written as:

$$fl(x) = (-1)^s m \beta^{e-t} = (-1)^s (a_1 a_2 \ldots a_t)_\beta \beta^{e-t}$$

where:

- $\beta \geq 2$ is the basics, an integer that determines the numeric system.
- $m = (a_1 a_2 \ldots a_t)_\beta$ is the mantissa, $(0 < m < \beta^t - 1)$ where t is the number of digits such that $0 < a_1 \leq \beta - 1$ and $0 \leq a_i \leq \beta - 1$ for $i = 2, \ldots, t$.

- $e = \mathbb{Z}$ is the exponent such that $L < e < U$, with $L < 0$ and $U > 0$.
- $s = \{0, 1\}$ is the sign.

In the definition of the numbers in the mantissa set we have to set the constraint $a_1 \neq 0$ to ensure the uniqueness of the representation.

The set of floating points has the following characteristic values:

- Machine epsilon, that is the distance between one and the smallest floating point number greater than one, and it is equal to:

$$\epsilon_M = \beta^{1-t}$$

- Round-off error, that is the relative error that is committed when substituting $x \in \mathbb{R} - \{0\}$ with his corresponding $fl(x) \in \mathbb{F}$. It is limited by:

$$\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\epsilon_M$$

  where $x \neq 0$.

- Cardinality of the floating point set:

$$\#\mathbb{F} = 2\beta^{t-1}(\beta - 1)(U - L + 1) + 1$$

  where:

  - 2 is needed to consider both positive and negative numbers.
  - β^{t-1} is the cardinality of values that can be taken by all digits.
  - $(\beta - 1)$ is the cardinality of values that can be taken by a_1.
  - $(U - L + 1)$ considers all the possible variations for the exponent.
  - 1 is needed to consider also the zero.

- The biggest and the smallest numbers in the set are found with the formula:

$$x_{min} = \beta^{L-1}$$
$$x_{max} = \beta^{U}(1 - \beta^{-t})$$

Example: In MATHLAB the floating point set is defined with the following variables:

$$(\beta = 2, t = 53, L = -1021, U = 1024)$$

With the command *eps* we can find the machine epsilon, that in MATLAB case is:

$$\epsilon_M = 2.22 \cdot 10^{-16}$$

With the command *realmin* and *realmax* we can find the smallest and the largest numbers representable that are equal to:

$$x_{min} = 2.225073858507201 \cdot 10^{-308}$$

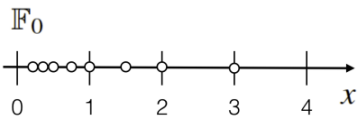
$$x_{max} = 1.797693134862316 \cdot 10^{308}$$

Since not all the numbers in the $\mathbb{R}$ set are also in the $\mathbb{F}$ set, in the second one there is no continuity. It is possible to demonstrate that while we are increasing the values of the numbers we are also increasing the distance between two consecutive numbers in $\mathbb{F}$.

Example : Let us consider the floating number set $\mathbb{F}(2, 2, -1, 2)$. The characteristic values of this set are:

- $\epsilon_M = \beta^{1-t} = 0.5$.
- $x_{min} = \beta^{L-1} = 0.25$.
- $x_{max} = \beta^U(1 - \beta^t) = 3$.
- $\#\mathbb{F} = 2\beta^{t-1}(\beta - 1)(U - L + 1) + 1 = 16$.

The exponent can have the values $-1, 0, 1$ and 2. The mantissa will be like $(a_1 a_2)_\beta$ because $t = 2$. The possible positive values are reported in the figure.



e	-1	0	1	2
$m = (10)_2 = 2$	$\frac{1}{4}$	$\frac{1}{2}$	1	2
$m = (11)_2 = 3$	$\frac{3}{8}$	$\frac{3}{4}$	$\frac{3}{2}$	3

The other important aspect is that the passage between the two sets causes the loss of two important properties such as associativity end the neutral number for the sum.

Chapter 2

Nonlinear equations

2.1 Iterative methods

To solve a nonlinear equation f we have to find $\alpha \in \mathbb{R}$ that is a zero of f such that $f(\alpha) = 0$. The set of all the polynomials of degree n is denoted by the symbol $\mathbb{P}_n$ that contains all the polynomial that have a grade less or equal to n.

Theorem (*Abel-Ruffini theorem*)

There is no solution in radicals to general polynomial equations of degree five or higher with arbitrary coefficients.

So, to solve polynomials with a degree higher than four we need to use the iterative methods. The general idea of those methods is the following:

1. Select an arbitrary initial value $x^{(0)}$ called initial guess, that is a hypothetical value for α.
2. Use the selected value as an input for a black-box function.
3. Use the output of the black-box function as the new $x^{(0)}$ and return to point one.

After some iterations (that depends on the chosen method) we will have a set of values $\{x^{(n)}\}$ convergent such that:

$$\lim_{n\to\infty} = \alpha$$

and the error related to the value found for α is equal to:

$$\lim_{n\to\infty} e^n = 0$$

That implies that the error can be also written as:

$$e^n = \alpha - x^{(n)}$$

Definition (*order of convergence*)

An iterative method for the approximation of the zero α of the function $f(x)$ is convergent with order p if and only if:

$$\lim_{k \to +\infty} \frac{\left| x^{(k+1)} - \alpha \right|}{\left| x^{(k)} - \alpha \right|^p} = \mu$$

where $\mu > 0$ is a real number independent of k, that is called asymptotic convergence factor. In the case of linear convergence ($p = 1$) it is necessary that $0 < \mu < 1$.

2.2 Bisection method

Theorem (*zeros of continuous functions*)

Let $f(x)$ be a continuous function on the interval $I = (a, b)$, that is $f \in C^0([a, b])$. If $f(a)f(b) < 0$, then there exists at least one zero $\alpha \in I$ of $f(x)$.

Let us assume that exist a unique zero, and let us call it α. The strategy of the bisection method is to have the given interval and select a sub-interval where f features a sign change. Following such procedure it is guaranteed that every interval selected this way will contain α. The sequence $\{x^{(k)}\}$ of the midpoints of these sub-intervals will inevitably tend to α since the length of the sub-intervals tends to zero as k tends to infinity.

The dimension of the k interval is given from the formula:

$$\left| I^{(k)} \right| = \frac{b - a}{2^k}$$

where $k > 0$. We said that the error is equal to $e^{(k)} := \left| x^{(k)} - \alpha \right|$. So, it is also possible to estimate the error with an upper bound, that is an error estimator. We have that.

$$e^{(k)} = \left| x^{(k)} - \alpha \right| < \frac{1}{2} \left| I^{(k)} \right| = (b - a) \left(\frac{1}{2} \right)^{(k+1)}$$

This result implies that this method is convergent, in fact the error is monotonic descending.

Given the tolerance $TOL > 0$ it is possible to calculate the minimum number of iterations of the bisection method, defined as k_{min}, such that $e^{(k_{min})} < TOL$. In fact, from the previous equation we obtain that:

$$k_{min} > \log_2 \left(\frac{b-a}{TOL} \right) - 1$$

STOPPING criterion

Algorithm 1 Algorithm for the bisection method

$a^{(0)} \leftarrow a$, $b^{(0)} \leftarrow b$, $x^{(0)} \leftarrow \dfrac{a^{(0)} + b^{(0)}}{2}$
while stopping criterion not satisfied **do**
    if $f(x^{(k+1)}) = 0$ **then**
        $\alpha \leftarrow x^{(k+1)}$
        return
    else
        if $f(x^{(k-1)}) f(a^{(k-1)}) < 0$ **then**
            $a^{(k)} \leftarrow a^{(k-1)}$
            $b^{(k)} \leftarrow x^{(k-1)}$
        end if
        if $f(x^{(k-1)}) f(b^{(k-1)}) < 0$ **then**
            $a^{(k)} \leftarrow x^{(k-1)}$
            $b^{(k)} \leftarrow b^{(k-1)}$
        end if
        $x^{(k)} \leftarrow \dfrac{a^{(k)} + b^{(k)}}{2}$
    end if
end while
return

2.3 Newton method

2.4 Fixed-point method

Chapter 3

Linear equations