

1 Introduction :

Mathematics originally had to deal with practical problems such as astronomy, navigation, etc., hence its numerical nature: its goal was to obtain a solution in the form of a "number." Throughout history, mathematicians have been interested in the numerical solution of problems.

The greatest scientists of the past studied the phenomena of nature by describing them mathematically, and then analyzing this description. The study of more complicated models gave rise to special numerical methods due in particular to Newton (Sir Isaac 1642-1727), Euler (Leonard 1707-1783), Labolchevski (Ivanovitch 1777-1855), Hermite (Charles 1822-1901) [1]. With the development of computers, the scope of mathematics and numerical methods has expanded, giving rise to new methods and new algorithms.

The term scientific computing refers to any calculation used in science. Numerical computing ishas both science and art [2]. Numerical computing is the branch of mathematics that proposes, develops, analyzes, and applies scientific computing methods to various fields, including analysis, linear algebra, geometry, approximation theory, functional equations, optimization, and differential equations. Other disciplines, such as physics, chemistry, natural and biological sciences, engineering, economics, and finance, frequently generate such problems that require scientific computing to solve.

1.1 Algorithms :

An algorithm is a calculation process that only implements arithmetic and logical operations. The word "algorithm" is of Arabic origin; it is a contraction and derivation of the name of the Arab mathematician Al-Khwarizmi (Muhammad, 780-850), who published two major works, one of which gave rise to the word "algebra" (الجبر و المقارنة). Three centuries later, the book, translated into Latin, was given the name *Algorismus*. In any case, it is a convenient term that serves to designate the set of operations that occur during a demonstration leading to the statement of a theorem [3,4].

To describe algorithms, there are methods such as flowcharts, for the sequence of tasks, and decision tables for the description of logical choices.

1.2 Computer science :

Computer science is the science of automatic information processing by computers. The term "computer science" comes from the contraction of the words "information" and "automatic". In 1957, the German engineer Karl Steinbuch coined the term "Informatik" for his essay entitled *Informatik: Automatische Informationsverarbeitung*, which can be rendered in French as "Informatique : traitement automatique de l'information"[5]. It was proposed in 1962 by Philippe Dreyfus and accepted by the French Academy in 1966.

¹

Computer science was born with the advent of computers in the 1950s, and a formal approach to computer science as a science was introduced.

²

¹ H.H. Goldstine 1977 « A history of Numerical Analysis from 16th through the 19th century » Springer-Verlag

²H.H. Goldstine 1977 « A history of Numerical Analysis from 16th through the 19th century » Springer-Verlag

The American mathematician Wiener (Nobel 1864-1966) established the foundations of cybernetics (Kybernetik) while another mathematician, Claude Elwood Shannon, developed information theory.

1.3 Computers :

Strictly speaking, a computer is nothing more than a machine that performs simple operations on sequences of electrical signals, which are conditioned in such a way that they can only take two states (for example, a maximum or minimum electrical potential).)[6]. The sequences of signals obey an "all or nothing" type logic and can therefore be conventionally considered as sequences of numbers never taking more than the two values 0 and 1. A digital system thus limited to two digits is called a binary system.

It is important to understand that, internally, a computer is completely incapable of processing anything other than binary numbers. Any other type of information must be converted, or encoded, into binary format. This is true not only for the data we wish to process (texts, images, sounds, numbers, etc.), but also for programs—that is, the sequences of instructions we provide to the machine to tell it what to do with this data.

1.4 Programming languages :

To simplify their work, computer scientists created intermediate languages, simpler than binary. Today, hundreds of programming languages exist. Each of these languages has its own specific characteristics. This is how many languages, considered high-level, came about. Most have fallen into oblivion, but a few have become legendary: Fortran, Basic, Cobol, Pascal, Ada, C, Visual Basic, Delphi, C++, Java, C#, PHP, Python, Octave[7,8]...

Practical work relatedThese books will be created using the Python language.**Python**Python was conceived in the late 1980s and officially released in 1991 by Guido van Rossum at Centrum Wiskunde & Informatica (Netherlands). Unlike C or C++, Python was designed from the outset as a simple and readable language, inspired by several existing languages such as ABC, C, and Modula-3. Guido van Rossum's main objective was to create a language that was easy to learn, allowing the writing of clear and concise programs, while integrating modern features such as object-oriented programming, functional programming, and automatic memory management.

1.5 The Errors

An important part of numerical analysis therefore consists of containing the effects of the errors thus introduced, which come from three main sources:

- modeling errors;
- errors in computer display;
- truncation errors.

1.5.1 Model errors

Modeling errors, as their name suggests, arise from the mathematical modeling stage of the physical phenomenon under study. This stage involves identifying the most significant causes of the observed phenomenon and expressing them in equation form. Due to the complexity of the phenomenon, it is necessary to simplify and disregard certain components.

Example:

1. Use classical mechanics rather than quantum mechanics in atomic physics.
2. Hooke's Law (linear model) for a spring $F=k \cdot x$ while $F=k(x)x$ (non-linear model)..

1.5.2 Truncation errors

Truncation errors are related to the accuracy of the algorithm used. They can be controlled by the algorithm itself. If a function is approximated by its Taylor series expansion, the truncation error will be obtained by evaluating the remainder of the expansion. Controlling this error will be achieved by setting an upper bound on this remainder.

Example::

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \approx \sum_{n=0}^{100} \frac{x^n}{n!} \quad (1.1.1)$$

An infinite process is truncated to obtain a finite procedure => Errors

1.5.3 Computer representation errors (Rounding errors) :

The rounding errors are imposed by the calculator.[6] Since the computer's memory representation of a number is finite, any real number is only known to a given precision of q significant digits. For any number between 0 and 1, the machine will write, for example:

$$x=0.a_1a_2a_3\dots a_n$$

When manipulating these numbers, the machine will have to choose between truncation or rounding to the nearest decimal place.

What is the value of 0.1 in base 2?

$$(0.1)_{10} = (0.0001100110011)_2$$

This cannot be stored exactly on the computer.

1.5.4 Methodological errors :

Method errors occur when an expression is poorly balanced and mixes values with a significant difference. This is a numerical calibration problem that is sensitive to rounding errors. In most cases, the algorithm needs to be modified. Consider the quadratic equation.

- 1_ Errors in mathematical description, including data errors.
- 2_ Insufficient precision of the method used où The exact solution requires a very large number of operations and a method error.
- 3_ Rounding errors in the data.

We give these errors the following names:

- _ Inherent errors.
- _ Method errors.
- Calculation errors.

And I the exact value of the parameter being sought

I the value corresponding to the mathematical description.

$\tilde{I}$ the corresponding value obtained by a numerical method without rounding

$\tilde{I}_h$ the corresponding value obtained through actual calculations.

SO :

$$\rho_0 = I - \tilde{I} : \text{inherent error.}$$

$$\rho_1 = \tilde{I} - \tilde{I}_h : \text{methodological error.}$$

$\rho_2 = \tilde{I}_h - \tilde{I}_h^*$ calculation error

The total error is:

$$\rho_3 = \rho_0 + \rho_1 + \rho_2 = I - \tilde{I}_h^* \quad (I.1.2)$$

1.5.5 Absolute and relative errors

If a is the exact value of a quantity and a^* an approximate value, we call absolute error on a^* and we note $\Delta(a^*)$ is defined as the quantity:

$$\Delta(a^*) = |a^* - a| \quad (I.1.3)$$

The relative error is the expression $\delta(a^*)$ described by:

$$\delta(a^*) = \left| \frac{a^* - a}{a} \right| \quad (I.1.4)$$

In practice, it is more convenient to use relative error to define the approximation than absolute error, since absolute error is often insufficient to characterize the degree of error.

Example :

if we take two time intervals $t_1 = 50.23 \pm 0.1$ s And $t_2 = 1.14 \pm 0.1$ s Although they have the same absolute error, the first one is more precise.

The approximate value is written according to the error used:

$$\begin{aligned} a &= a^* \pm \Delta(a^*) && \text{si on utilise l'erreur absolu.} \\ a &= a^* (1 \pm \delta(a^*)) && \text{si on utilise l'erreur relatif.} \end{aligned}$$

Al-Khwarizmi

Al-Khwarizmi clearly explained in his treatise how to solve problems with one unknown using two operations:

- Al-jabr: transposing certain terms in an equation to eliminate negative values. For example:
transformer $x = 40 - 4x$ in $5x = 40$.
- Al-muqabala: balancing the remaining positive values in the equation. For example:
 $50 + x^2 = 29 + 10x$ becomes $x^2 + 21 = 10x$.

All of this may seem childish, but algebraic notation did not exist at the time, and manipulations and equations were stated in everyday language. The word al-jabr became algebra in Latin and “algèbre” in French.

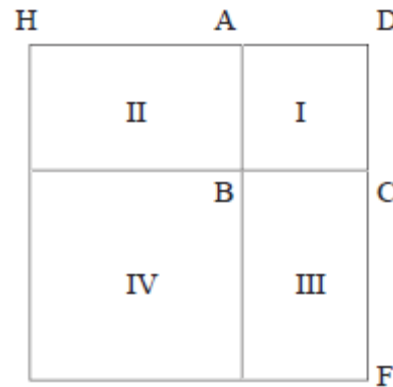


Fig. 1 1: Diagram used by Al-Khwarizmi to explain his solution to
the quadratic equation

Al-Khwarizmi explains the solution to the quadratic equation, in particular the process of completing the square, geometrically, drawing inspiration from Euclid. Let's see how he proceeds with an example (fig. 1). Suppose we have to solve the equation $x^2 + 10x = 39$. Let AB be the length representing the unknown x . Let's construct a square ABCD with this length. Then, let's extend segments DA to DH and DC to DF, so that the lengths AH and CF are half the coefficient of x in the equation, or 5. The sum of the areas of regions I, II, and III is then $x^2 + 2 * 5x$, and therefore equal to 39. Let's complete the square by adding region IV, with an area of $5 * 5 = 25$, to each side of the equation. The total area of the large square is then $(x + 5)^2 = 39 + 25 = 64$. We then simply take the square root and $x + 5 = 8$, or $AB + CF = 8$. Therefore, $AB = 8 - 5 = 3$, which is the root we are looking for. Of course, this geometric approach prevents us from considering negative roots (here, $x = -13$) and even more so complex roots. It should also be noted that literal algebra is used here only to support the explanation and was unknown at the time.