Numerical analysis applied to financial issues

Yohan PENEL[1]
Mathieu GIRARDIN[2]

February 6, 2014

[1] Team ANGE (CETMEF–INRIA–UPMC–CNRS), 4 place Jussieu, 75005 Paris. Courriel : penel@ann.jussieu.fr (lectures, exercises)

[2] CEA Saclay, DEN/DANS/DM2S/STMF/LMEC (practical works)

Contents

Chapter 1

Introduction

1.1 Numerical analysis

That numerical analysis is both a science and an art is a cliché to specialists in the field but is misunderstood by nonspecialists. Is calling it an art and a science only a euphemism to hide the fact that numerical analysis is not a sufficiently precise discipline to merit being called a science? Is it true that "numerical analysis" is something of a misnomer because the classical meaning of analysis in mathematics is not applicable to numerical work? In fact, the answer to both these questions is no. The juxtaposition of science and art is due instead to an uncertainty principle which often occurs in solving problems, namely that to determine the best way to solve a problem may require the solution of the problem itself. In other cases, the best way to solve a problem may depend upon a knowledge of the properties of the functions involved which is unobtainable either theoretically or practically. [...]

As a science, then, numerical analysis is concerned with the processes by which mathematical problems can be solved by the operations of arithmetic. Sometimes this will involve the development of algorithms to solve a problem already in a form in which the solution can be found by arithmetic means, *e.g.* simultaneous linear equations. Often it will involve replacing quantites which cannot be calculated arithmetically, *e.g.* derivatives or integrals, by approximations which permit an approximate solution to be found. In this case, we shall naturally be interested in the errors incurred in our approximation. But in any case, the tools we shall use in developing the processes of numerical analysis will be the tools of exact mathematical analysis as classically understood.

As an art, numerical analysis is concerned with choosing that procedure (and suitably applying it) which is the "best" suited to the solution of a particular problem. This implies the need for anyone who wishes to practice numerical analysis to develop experience and with it – it is hoped – intuition.

in *A First Course in Numerical Analysis,* A. Ralston & P. Rabinowitz

As written by Ralston & Rabinowitz, Numerical Analysis is an answer to several users of mathematics. It helps provide an approximate solution to problems for which one is not able to exhibit an exact solution. Indeed, many systems of equations arising in physical, chemical, biological modelling may not have a "trivial" solution.

The main tools of numerical analysis are finite-dimensional linear systems, sequences and polynomials, which seem to be more practical than differential equations and functions. Operations with these former elements are purely arithmetic and thus achievable. The art of numerical analysis consists in formulating a new problem involving these practical tools and at the same time in controlling the error induced by the choice of this "nearby" problem.

The development of computers over the last decades increased the significance and the impact of numerical analysis. The high speed at which linear operations can be done indeed made this discipline very attractive. The fact remains that the error between the true[1] and the numerical solutions must be controlled, which leads to a characteristic issue: to improve the accuracy of the solution (*i.e.* by decreasing the error), sophisticated methods must be developed but they induce a (maybe much) larger computational time. The equilibrium between accuracy and efficiency is thus a major issue and the answer varies from one numerical analyst to another. It depends on what they want to focus on: either the overall behaviour of the solution or the very values of the solution at some points.

A critical concept in numerical analysis is **stability**. More precisely, the question deals with the fact that a numerical method which provides good results for an equation may turn to be not suited to a very close but different equation. This close problem could be the same equation but with a small perturbation on the boundary conditions. This can be due to the method (see below) ... or to the problem itself. In the latter case, the problem is said to be *ill-posed* as it is very sensitive to perturbations on data. A classical example is the Cauchy problem

$$\begin{cases} y'' - y' - 2y = 0, \\ y(0) = 1, \\ y'(0) = -1, \end{cases}$$

whose solution is $y(t) = e^{-t}$ which has a fast decay as t goes to infinity. If we modify (for some $\varepsilon > 0$) the initial condition as

$$\begin{cases} \tilde{y}'' - \tilde{y}' - 2\tilde{y} = 0, \\ \tilde{y}(0) = 1, \\ \tilde{y}'(0) = -1 + \varepsilon, \end{cases}$$

the solution becomes $\tilde{y}(t) = \left(1 - \frac{\varepsilon}{3}\right) e^{-t} + \frac{\varepsilon}{3} e^{2t}$. Hence for $\varepsilon = 10^{-3}$, we have $y(10) \approx 0.000045$ and $\tilde{y}(10) \approx 7.34$. This is the kind of instability that must be taken into account. Likewise, the numerical scheme must be stable in order to provide reliable results.

[1] The reader should bear in mind that any model aims at representing real life but cannot reproduce the whole complexity of reality. We may incur a large discrepancy compared to experiments. Modelling issues will not be investigated in this course.

1.2 Deterministic approach of finance

1.2.1 Model

Let us set in this section the model we shall simulate in this course, namely the Black & Scholes equation. This equation models the evolution of the price of a financial option (*put* for a selling option, *call* for a buying option). More precisely:

> An option is a contract giving the owner the right to buy [or sell] a fixed number of share of a specific common stock at a fixed price at a certain date.
>
> in *Option Market*, J. Cox & M. Rubinstein

The **underlying asset** is denoted by S. Its evolution is governed by another model which is not the topic of this course. The fixed price is called the **strike** and denoted by K. The date of expiration of the option is called the **maturity** and denoted by $\mathcal{T}$. The very issue is to evaluate the price V of the option, especially at time 0. Indeed, the part of the option who commited to sell [or buy] the shares to the owner of the option must ensure that they would not loose money. That is why they have to adapt the fee which is the price of the contract. They do not know what the price of the underlying will be in the future. However, it is possible to evaluate the value of the option at maturity:

- For a call option, if the value $S_{\mathcal{T}}$ of the underlying at time $\mathcal{T}$ is greater than the strike, the owner exercises the option and buys the underlying at price K. Otherwise, the owner buys the asset at the market price and do not exercise. Hence, the price of the option is

$$C(\mathcal{T}, S_{\mathcal{T}}) = \max\{S_{\mathcal{T}} - K, 0\}. \tag{1.1a}$$

- For a put option, the price becomes

$$P(\mathcal{T}, S_{\mathcal{T}}) = \max\{K - S_{\mathcal{T}}, 0\}. \tag{1.1b}$$

Under financial hypotheses (like the fact that the market rules out *arbitrage*) about which we shall not go into details, the Black & Scholes model for pricing options reads

$$\begin{cases} \dfrac{\partial V}{\partial t} + \dfrac{\sigma(t,S)^2 S^2}{2}\dfrac{\partial^2 V}{\partial S^2} + r(t) S \dfrac{\partial V}{\partial S} - r(t) V = 0, \qquad t \in (0, \mathcal{T}),\ S \geq 0, & (1.2a) \\ V(\mathcal{T}, S) = V_{\mathcal{T}}(S), \qquad S \geq 0. & (1.2b) \end{cases}$$

r is the **interest rate**, σ is the **volatility** and $V_{\mathcal{T}}(S)$ is the **payoff** given by (1.1a) or (1.1b). Notice that the prices for call and put options are related by the call–put parity formula

$$C(t,S) - P(t,S) = S - K \exp\left(-\int_t^{\mathcal{T}} r(\tau)\,\mathrm{d}\tau\right). \tag{1.3}$$

System (1.2) must be supplemented with boundary conditions in the "asset" domain $(0,+\infty)$. First, if the model is reasonable, we must have $C(t,S) \leq S$, which implies

$$C(t,0) = 0. \tag{1.4a}$$

The call–put parity formula (1.3) leads to

$$P(t,0) = K\exp\left(-\int_t^{\mathscr{T}} r(\tau)\,\mathrm{d}\tau\right). \tag{1.4b}$$

Likewise, we must have

$$P(t,S) \xrightarrow[S\to+\infty]{} 0, \tag{1.4c}$$

and thus

$$C(t,S) \underset{S\to+\infty}{\sim} S - K\exp\left(-\int_t^{\mathscr{T}} r(\tau)\,\mathrm{d}\tau\right). \tag{1.4d}$$

1.2.2 Simple case

When r and σ are assumed to be constant and equal to r_0 and σ_0, it is possible to derive explicit expressions for C and P. Indeed, through successive change of variables, equation (1.2a) comes down to the heat equation.

The first change of variables deals with the time reversal in order to make the terminal condition an initial condition. Set $\tilde{V}(\theta,S) = V(\mathscr{T}-\theta,S)$. The equation becomes

$$\begin{cases} -\dfrac{\partial\tilde{V}}{\partial\theta} + \dfrac{\sigma_0^2 S^2}{2}\dfrac{\partial^2\tilde{V}}{\partial S^2} + r_0 S\dfrac{\partial\tilde{V}}{\partial S} - r_0\tilde{V} = 0, & \theta\in(0,\mathscr{T}),\ S\geq 0, \\ \tilde{V}(0,S) = V_{\mathscr{T}}(S), & S\geq 0. \end{cases} \tag{1.5}$$

The next change of variables must lead to a PDE with constant coefficients, namely $\varphi(\theta,x) = \tilde{V}(\theta,e^x)$ and the equation reads

$$\begin{cases} -\dfrac{\partial\varphi}{\partial\theta} + \dfrac{\sigma_0^2}{2}\dfrac{\partial^2\varphi}{\partial x^2} + \left(r_0 - \dfrac{\sigma_0^2}{2}\right)\dfrac{\partial\varphi}{\partial x} - r_0\varphi = 0, & \theta\in(0,\mathscr{T}),\ x\in\mathbb{R}, \\ \varphi(0,x) = V_{\mathscr{T}}(e^x), & x\in\mathbb{R}. \end{cases} \tag{1.6}$$

The last one $\psi(\theta,x) = \varphi(\theta,x)e^{-a\theta-bx}$ with $b = \frac{1}{2} - \frac{r_0}{\sigma_0^2}$ and $a = -r_0 - \frac{\sigma_0^2}{2}b^2$ leads to the heat equation

$$\begin{cases} -\dfrac{\partial\psi}{\partial\theta} + \dfrac{\sigma_0^2}{2}\dfrac{\partial^2\psi}{\partial x^2} = 0, & \theta\in(0,\mathscr{T}),\ x\in\mathbb{R}, \\ \psi(0,x) = V_{\mathscr{T}}(e^x)e^{-bx}, & x\in\mathbb{R}. \end{cases} \tag{1.7}$$

The solution to (1.7) is

$$\psi(\theta,x) = \frac{1}{\sqrt{2\pi\sigma_0^2\theta}}\int_{\mathbb{R}}\exp\left[\frac{-y^2}{2\sigma_0^2\theta}\right]\psi(0,x-y)\,\mathrm{d}y.$$

Hence

$$V(t,S) = \frac{e^{a(\mathcal{T}-t)}}{\sqrt{2\pi\sigma_0^2(\mathcal{T}-t)}} \int_{\mathbb{R}} \exp\left[by - \frac{y^2}{2\sigma_0^2(\mathcal{T}-t)}\right] V_{\mathcal{T}}(Se^{-y})\,\mathrm{d}y.$$

We deduce the Black & Scholes formulae

$$\begin{cases} C(t,S) = S\Psi(d_1) - Ke^{-r_0(\mathcal{T}-t)}\Psi(d_2), & (1.8a) \\ P(t,S) = -S\Psi(-d_1) + Ke^{-r_0(\mathcal{T}-t)}\Psi(-d_2), & (1.8b) \end{cases}$$

with

$$\Psi(d) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d} e^{-x^2}\,\mathrm{d}x, \qquad d_1 = \frac{\ln S - \ln K + \left(r_0 + \frac{\sigma_0^2}{2}\right)(\mathcal{T}-t)}{\sigma_0\sqrt{\mathcal{T}-t}}, \qquad d_2 = d_1 - \sigma_0\sqrt{\mathcal{T}-t}.$$

Hence, in this simple case, we have an explicit solution but it may not seem practical due to its expression. In more realistic (and thus more complex) situations, we do not have any solution. That is why methods should be worked out to provide approximations of the solutions. It will be the matter in the following chapters.

The only statement about (1.2) is the following existence result:

Theorem 1.1 *Under the following hypotheses,* (1.2) *has a unique solution* $V \in \mathcal{C}^0\big([0,\mathcal{T}] \times \mathbb{R}_+\big)$ *which is of class* $\mathcal{C}^1$ *with respect to* t *and* $\mathcal{C}^2$ *with respect to* S*:*

- $\exists\, C_1 > 0,\ \forall\, t \in [0,\mathcal{T}],\ \forall\, (S_1,S_2) \in \mathbb{R}_+^2,\ |S_1\sigma(t,S_1) - S_2\sigma(t,S_2)| \le C_1|S_1 - S_2|;$
- $\exists\, C_3 > C_2 > 0,\ \forall\, (t,S) \in [0,\mathcal{T}] \times \mathbb{R}_+,\ C_2 \le \sigma(t,S) \le C_3;$
- $\exists\, C_4 > 0,\ \forall\, (t_1,t_2) \in [0,\mathcal{T}]^2,\ |r(t_1) - r(t_2)| \le C_3|t_1 - t_2|;$
- $\exists\, C_6 > C_5 > 0,\ \forall\, t \in [0,\mathcal{T}],\ C_5 \le r(t) \le C_6;$
- $\exists\, C_7 > 0,\ \forall\, S \in \mathbb{R}_+,\ 0 \le V_{\mathcal{T}}(S) \le C_7(1+S).$

1.3 Mathematical tools

We end this introduction by some reminders about tools that are often involved in numerical analysis.

1.3.1 Sequences

Sequences are a practical tool that may either be calculated directly or by means of a fast computation. Indeed, rather than manipulating a function, it seems easier to deal with countable sets of values of this function. A sequence $(y_n)_{n\ge1}$ is an ordered set of numbers, *i.e.* each element of the set is labelled by an index n. Sequences may be of two types:

- **"Explicit"** insofar as it is directly possible to compute any value of the sequence, *e.g.* y_{100}. For instance, if the sequence is defined by $y_n = 3n^2 + 1$, then $y_{100} = 30001$;
- **"Inductive"**: to compute y_{100}, it is necessary to compute y_{99} before, and so on till y_2. It is the case for example of the sequence defined by $y_1 = 4$ and $y_{n+1} = 3y_n^2 + 1$, $n \ge 1$: y_{100} is not directly achievable.

Nethertheless, some of them belong *a priori* to the second category but turn out to be explicit:

- **Arithmetic progressions** defined by the induction

  $$y_1 = \alpha,\ y_{n+1} = y_n + \beta,\ n \geq 1$$

  for some fixed $(\alpha, \beta) \in \mathbb{R}^2$. Then we have

  $$y_n = \alpha + (n-1)\beta.$$

- **Geometric progressions** defined by the induction

  $$y_1 = \alpha,\ y_{n+1} = \gamma \times y_n,\ n \geq 1$$

  for some fixed $(\alpha, \gamma) \in \mathbb{R}^2$. Then we have

  $$y_n = \alpha \times \gamma^{n-1}.$$

- **Arithmetico-geometric progressions** defined by the induction

  $$y_1 = \alpha,\ y_{n+1} = \delta \times y_n + \varepsilon,\ n \geq 1$$

  for some fixed $(\alpha, \varepsilon) \in \mathbb{R}^2$ and $\delta \neq 1$. Then we have

  $$y_n = \left(\alpha + \frac{\varepsilon}{\delta - 1}\right)\delta^{n-1} - \frac{\varepsilon}{\delta - 1}.$$

- **Rational progession** defined by the induction

  $$y_1 = \alpha,\ y_{n+1} = \frac{\delta y_n + \varepsilon}{\eta y_n + \zeta},\ n \geq 1$$

  for some real coefficients α, δ, ε, η, ζ such that $\eta \neq 0$ and $\delta\zeta \neq \varepsilon\eta$. We must first ensure that $y_n \neq \frac{-\zeta}{\eta}$ for all n so that the sequence is well-defined. Then, we determine the solutions to the characteristic equation

  $$\eta\ell^2 + (\zeta - \delta)\ell - \varepsilon = 0.$$

  If there are exactly two solutions ℓ_1 and ℓ_2, then the sequence $z_n = \frac{y_n - \ell_1}{y_n - \ell_2}$ is geometric. If there is a unique solution ℓ_0, then $z_n = \frac{1}{y_n - \ell_0}$ is arithmetic.

- **2nd–order linear inductions with constant coefficients** defined by

  $$y_1 = \alpha,\ y_2 = \beta,\ \zeta \times y_{n+2} + \eta \times y_{n+1} + \theta \times y_n = 0,\ n \geq 1$$

  for some fixed $(\alpha, \beta, \eta, \theta) \in \mathbb{R}^4$ and $\zeta \neq 0$.

Let us set $\Delta = \eta^2 - 4\zeta\theta$ which is the discriminant of the characteristic equation[2]

$$\zeta r^2 + \eta r + \theta = 0. \tag{1.9}$$

The solution depends on the sign of Δ:

- ◇ If $\Delta > 0$, (1.9) has two real solutions r_1 and r_2, and the sequence reads

  $$y_n = \kappa r_1^n + \lambda r_2^n,$$

  where κ and λ are deduced from α and β.
- ◇ If $\Delta = 0$, (1.9) has one real solution r_0 and the sequence reads

  $$y_n = (\kappa + \lambda n) r_0^n.$$

- ◇ If $\Delta < 0$, (1.9) has two complex solutions r_1 and $\overline{r_1}$, and the sequence reads

  $$y_n = |r_1|^n \left(\kappa \cos(n \operatorname{Arg} r_1) + \lambda \sin(n \operatorname{Arg} r_1)\right).$$

The general first-order inductive case reads

$$y_{n+1} = f(y_n),\ n \geq 1 \ \text{ with } y_1 \text{ given}$$

where f is a continuous function over a certain interval $I \subset \mathbb{R}$. The outline of the study is

1. Examine the existence of an interval $J \subseteq I$ such that $f(y) \in J$ for all $y \in J$: it is necessary for the sequence to be well-defined;
2. Calculate the potential fixed points of f, *i.e.* such that $f(y) = y$. **If the sequence converges, the limit is necessary a fixed point of** f;
3. Determine the monotonicity of f:
   (a) If f is monotone-increasing, then the sequence (y_n) is monotonic. If $f(y_1) > y_1$, then (y_n) is monotone-increasing. If $f(y_1) < y_1$, it is decreasing.
   (b) If f is monotone-decreasing, then the subsequences (y_{2n}) and (y_{2n+1}) are monotonic.

A useful result states that:

Proposition 1.1 *If a sequence is monotone-increasing and bounded, then it is convergent.*

1.3.2 Polynomials

Polynomials are the simplest functions in functional analysis insofar as they can be represented by a finite number of coefficients and as they are continuous and differentiable everywhere. That is why they are used as often as possible to approximate functions (see § 1.3.4).

[2] This equation arises when looking for sequences of the form $y_n = r^n$.

Usual definitions A polynomial is a combination of powers of the unknown, *i.e.* of the form

$$P = \sum_{k=0}^{n} a_k X^k$$

for some $n \in \mathbb{Z}_+$ (such that $a_n \neq 0$) and which is called the **degree** of P. All coefficients (a_k) are assumed to be complex.

Definition 1.1 *A root of P is a complex number ξ such that $P(\xi) = 0$.*

Definition 1.2 *The derivative of polynomial P is the polynomial $P' = \sum_{k=0}^{n-1} a_{k+1}(k+1)X^k$.*

Definition 1.3 *A root ξ of polynomial P is said to be of multiplicity $m \in \{1, \ldots, n\}$ if $P(\xi) = 0$, $P'(\xi) = 0$, $\ldots$, $P^{(m-1)}(\xi) = 0$.*

Definition 1.4 *Polynomial Q is said to divise P if there exists a polynomial R such that $P = QR$.*

Common properties

Proposition 1.2 *ξ is a root of polynomial P if $X - \xi$ divises P. Its multiplicity is m if $(X-\xi)^m$ divises P.*

Theorem 1.2 *Every polynomial of degree n has exactly n complex roots. In other words, there exist $(\xi_1, \ldots, \xi_r) \in \mathbb{C}^r$ and $(m_1, \ldots, m_r) \in \mathbb{Z}_+^r$ such that*

$$\sum_{k=1}^{r} m_k = n \text{ and } P = a_n \prod_{k=1}^{r} (X - \xi_k)^{m_k}.$$

Corollary 1.1 *The product of the roots of P is equal to $\prod_{k=1}^{r} \xi_k^{m_k} = (-1)^n \frac{a_0}{a_n}$. Their sum is $\sum_{k=1}^{r} m_k \xi_k = -\frac{a_{n-1}}{a_n}$.*

1.3.3 Linear algebra

In many cases, the numerical resolution of an ODE or a PDE amounts to solving a linear system. We recall in the sequel some definitions and results about matrices.[3]

Linear mappings

A linear mapping $f : E \to F$ (where E and F are two $\mathbb{C}$-vector spaces) is a function satisfying the property

$$\forall\, (x, y) \in E^2,\ \forall\, \lambda \in \mathbb{C}, f(x + \lambda y) = f(x) + \lambda f(y).$$

When E and F have finite dimensions n and p respectively, f can be represented in certain bases by a matrix with p rows and n columns.

For example, the linear system

$$\begin{cases} 2x + 3y + 4z = 1, \\ x - 2z = 0, \end{cases}$$

[3] We only deal with square matrices, which corresponds to the case where there are as many equations as unknowns.

can be interpreted as the functional equation

$$f(X) = Y$$

where $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$, $Y = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $f(X) = \begin{pmatrix} 2x+3y+4z \\ x-2z \end{pmatrix}$ or as the linear system

$$AX = Y$$

where

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 1 & 0 & -2 \end{pmatrix}.$$

The number of rows corresponds to the number of equations and the number of columns to the number of unknowns.

The issue is then to determine whether this linear system has a unique solution. In the sequel, we shall only focus on the square case $n = p$, *i.e.* when there are as many equations as unknowns. Solving a linear system comes down to studying the corresponding matrix.

Matrices

The set of square matrices with n rows, n columns and complex coefficients is denoted by $\mathscr{M}_n(\mathbb{C})$. A coefficient of a matrix $A \in \mathscr{M}_n(\mathbb{C})$ located at row i and column j is referred to as A_{ij} where $(i, j) \in \{1, \dots, n\}$.

Standard operations with matrices are

- the addition: $[A+B]_{ij} = A_{ij} + B_{ij}$ (addition term by term);
- the multiplication by a scalar $\lambda \in \mathbb{C}$: $[\lambda A]_{ij} = \lambda A_{ij}$;
- the matrix multiplication: $[AB]_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$. The multiplication is not commutative: generally, $AB \neq BA$;
- the transpose operation: $[A^T]_{ij} = A_{ji}$. Notice that $[AB]^T = B^T A^T$.

We note $\mathscr{I}_n$ the matrix whose coefficients are $\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$

We finally introduce the **determinant** which is a linear form upon matrix spaces and which determines wether the column vectors of the matrix are linearly dependent (in that case, $\det(A) = 0$). Notice that $\det(AB) = \det(A)\det(B)$ and $\det(\lambda A) = \lambda^n \det(A)$.

We then tackle the issue of the resolution of the linear system $Ax = b$ for a given matrix $A \in \mathscr{M}_n(\mathbb{C})$ and a given vector $b \in \mathbb{C}^n$. The number of solutions (0, 1 or ∞) depends on the invertibility of A.

Definition 1.5 *A matrix $A \in \mathscr{M}_n(\mathbb{C})$ is invertible if there exists a matrix B such that $AB = \mathscr{I}_n$. The set of invertible matrices is noted* $\mathrm{GL}_n(\mathbb{C})$.

Proposition 1.3 *If A and B are invertible, then AB is invertible and $[AB]^{-1} = B^{-1}A^{-1}$.*

Definition 1.6 *The kernel of A is the set*

$$\ker A = \{x \in \mathbb{C}^n \mid Ax = 0\}.$$

Looking for a vector x such that $Ax = 0$ amounts to looking for a linear combination such that $\sum_{i=1}^{n} x_i A_i = 0$ where A_i is the i-th column vector of A.

Definition 1.7 *The image of A is the set*

$$\mathrm{Im}(A) = \{y \in \mathbb{C}^n \mid \exists\, x \in \mathbb{C}^n,\ y = Ax\}.$$

Then, the linear system $Ax = b$ has

- 1 solution if A is invertible;
- 0 solution if A is not invertible and if $b \notin \mathrm{Im}(A)$: two (at least) equations are contradictory;
- infinitely many solutions if A is not invertible and if $b \in \mathrm{Im}(A)$: two (at least) equations are redundant.

Proposition 1.4 *A is invertible iff one of the following statements holds:*

- $\ker A = \{0\}$*;*
- $\mathrm{Im}(A) = \mathbb{C}^n$*;*
- *There exists no linear combination of the column vectors of A equal to* 0*;*
- $\det(A) \neq 0$.

Classification

The simplest matrices are diagonal, *i.e.* such that $A_{ij} = 0$ for $i \neq j$. The set of diagonal matrices is noted $\mathcal{D}_n(\mathbb{C})$.

Proposition 1.5 *Let A and B be two diagonal matrices.*

- *A is invertible iff $A_{ii} \neq 0$ for all $i \in \{1, \ldots, n\}$. Its inverse is also diagonal and $(A^{-1})_{ii} = \frac{1}{A_{ii}}$;*
- *$AB = BA$ is a diagonal matrix whose diagonal coefficients are $(AB)_{ii} = A_{ii}B_{ii}$;*
- *If A is invertible, then for all $b \in \mathbb{C}^n$, the solution to $Ax = b$ is given by $x_i = \frac{b_i}{A_{ii}}$.*

Another kind of simple matrices is the case of triangular systems, *i.e.* when $A_{ij} = 0$ for $i < j$ (upper, $\mathcal{T}_n^+(\mathbb{C})$) or $i > j$ (lower, $\mathcal{T}_n^-(\mathbb{C})$).

Proposition 1.6 *Let A and B be two lower triangular matrices.*

- *A is invertible iff $A_{ii} \neq 0$ for all $i \in \{1, \dots, n\}$. Its inverse is also lower triangular;*
- *AB is a lower triangular matrix;*
- *If A is invertible, then for all $b \in \mathbb{C}^n$, the solution to $Ax = b$ is unique and given by* ***Algorithm 1***.

This proposition naturally adapts to the case of upper triangular matrices and the corresponding algorithm is **Algorithm 2**.

We may also mention the following kinds of matrices:

Definition 1.8 *Matrix A is said to be*

- ***orthogonal*** *if $A^T A = \mathscr{I}_n$ and* ***unitary*** *if $\overline{A}^T A = \mathscr{I}_n$;*
- ***symmetric*** *if $A^T = A$ and* ***hermitian*** *if $\overline{A}^T = A$;*
- ***positive-definite*** *if for all $x \in \mathbb{C}^n$, $x \neq 0$, $x^T Ax > 0$.*

Decompositions

A famous procedure to solve the linear system $Ax = b$ where A is invertible is the Gaussian elimination. If the process is successful without permutation, then A can be decomposed as LU where $L \in \mathscr{T}_n^-(\mathbb{C})$ and $U \in \mathscr{T}_n^+(\mathbb{C})$. Hence, the resolution of $Ax = b$ can be split into 3 parts:

1. Compute L and U;
2. Resolution of the lower triangular system $Ly = b$;
3. Resolution of the upper triangular system $Ux = y$.

Algorithm 3 relies on the blockwise product

$$\left(\begin{array}{c|c|c} L^{(p)} & 0 & 0 \\ \hline \mathscr{L}^{(p)} & 1 & 0 \\ \hline \star & \star & \star \end{array}\right)\left(\begin{array}{c|c|c} U^{(p)} & \mathscr{U}^{(p)} & \star \\ \hline 0 & u^{(p)} & \star \\ \hline 0 & 0 & \star \end{array}\right) = \left(\begin{array}{c|c|c} L^{(p)}U^{(p)} & L^{(p)}\mathscr{U}^{(p)} & \star \\ \hline \mathscr{L}^{(p)}U^{(p)} & \mathscr{L}^{(p)}\mathscr{U}^{(p)} + u^{(p)} & \star \\ \hline \star & \star & \star \end{array}\right)$$

where $U^{(p)} \in \mathscr{T}_p^+(\mathbb{C})$, $L^{(p)} \in \mathscr{T}_p^-(\mathbb{C})$, $\mathscr{U}^{(p)} \in \mathbb{C}^p$, $\left[\mathscr{L}^{(p)}\right]^T \in \mathbb{C}^p$ and $u^{(p)} \in \mathbb{C}$.

Proposition 1.7 *Matrix A has a LU-decomposition if* ***Algorithm 3*** *reaches $p = n$,* i.e. *if $u^{(p)} \neq 0$ for all p. This occurs when all extracted matrices $(A_{ij})_{1 \leq i,j \leq p}$, $p_{\leq} n$, are invertible.*

The interest of the LU-decomposition relies on the fact that rather than solving directly the system $Ax = b$, we solve two triangular systems, namely $Ly = b$ and $Ux = y$. The very cost of this procedure concerns the computation of L and U.

There exist other decompositions that may help the resolution of particular linear systems.

Proposition 1.8 (Schur) *Any matrix can be decomposed as $\overline{U}^T TU$ where U is unitary and $T \in \mathscr{T}_n^+(\mathbb{C})$.*

Algorithm 1 Resolution of a lower triangular system

```
Data: $A \in \mathscr{T}_n^-(\mathbb{C})$, $b \in \mathbb{C}^n$
for $i$ from 1 to $n$ do
    if $A_{ii} = 0$ then
        Matrix $A$ is not invertible
        Algorithm stopped
    else
        $x_i \leftarrow \dfrac{b_i - \sum_{k=1}^{i-1} A_{ij} x_j}{A_{ii}}$
    end if
end for
```

Algorithm 2 Resolution of an upper triangular system

```
Data: $A \in \mathscr{T}_n^+(\mathbb{C})$, $b \in \mathbb{C}^n$
for $i$ from $n$ to 1 do
    if $A_{ii} = 0$ then
        Matrix $A$ is not invertible
        Algorithm stopped
    else
        $x_i \leftarrow \dfrac{b_i - \sum_{k=i+1}^{n} A_{ij} x_j}{A_{ii}}$
    end if
end for
```

Algorithm 3 Factorization

```
Data: $A \in \mathrm{GL}_n(\mathbb{C})$
$U_{11} \leftarrow A_{11}$
$L_{11} \leftarrow 1$
$p \leftarrow 1$
while $p \leq n$ and $U_{pp} \neq 0$ do
    $\mathscr{U} \leftarrow \left[(L_{ij})_{1\leq i\leq p, 1\leq j\leq p}\right]^{-1} (A_{i,p+1})_{1\leq i\leq p}$
    $\mathscr{L} \leftarrow (A_{p,j})_{1\leq j\leq p} \left[(U_{ij})_{1\leq i\leq p, 1\leq j\leq p}\right]^{-1}$
    $(U_{i,p+1})_{1\leq i\leq p} \leftarrow \mathscr{U}$
    $(L_{p+1,j})_{1\leq j\leq p} \leftarrow \mathscr{L}$
    $U_{p+1,p+1} \leftarrow A_{p+1,p+1} - \mathscr{L}\mathscr{U}$
    $L_{p+1,p+1} \leftarrow 1$
    $p \leftarrow p + 1$
end while
```

Proposition 1.9 (Cholesky) *Any real symmetric positive-definite matrix can be decomposed as $B^T B$ where $B \in \mathscr{T}_n^+(\mathbb{R})$. Moreover, if we impose $B_{ii} > 0$ for all $i \in \{1, \ldots, n\}$, then B is unique.*

Proposition 1.10 (QR) *Any real invertible matrix can be decomposed as QR where Q is orthogonal and $R \in \mathscr{T}_n^+(\mathbb{R})$. The decomposition is unique if we suppose that all diagonal coefficients from R are positive.*

Eigenvalues

The analysis of the spectrum of a matrix provides important properties.

Definition 1.9 *$\lambda \in \mathbb{C}$ is said to be an **eigenvalue** of a matrix $A \in \mathscr{M}_n(\mathbb{C})$ if there exists $x \in \mathbb{C}^n \backslash \{0\}$ such that $Ax = \lambda x$. x is called an **eigenvector** of A associated to λ. The set of all eigenvalues of A is called the **spectrum** of A.*

Definition 1.10 *Let λ_i and x_i the eigenvalues of A and some corresponding eigenvectors. A is said to be **diagonalizable** if the eigenvectors x_i form a basis of $\mathbb{C}^n$. Then there exist $D \in \mathscr{D}_n(\mathbb{C})$ and $U \in \mathrm{GL}_n(\mathbb{C})$ such that*

$$A = UDU^{-1}.$$

Moreover, we have $D_{ii} = \lambda_i$ and the column vectors of U are the eigenvectors x_i.

Proposition 1.11 *Any matrix $A \in \mathscr{M}_n(\mathbb{C})$ has n eigenvalues. There are the roots of polynomial* $\det(A - X\mathscr{I}_n)$.

Proposition 1.12 *A matrix is positive-definite if and only if its eigenvalues are stricly positive.*

Proposition 1.13 *Any real symmetric matrix is diagonalizable and U is orthogonal.*

Linear stability

We finally get interested in the stability of linear systems, *i.e.* its sensitivity to small perturbations of the matrix or of the right hand side. A key-point in this matter is the condition number of a matrix. Let $\|\cdot\|$ be a norm on $\mathbb{C}^n$. The corresponding norm on $\mathscr{M}_n(\mathbb{C})$ is

$$|||A||| = \sup_{x \in \mathbb{C}^n, x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

Definition 1.11 *The **condition number** of a matrix $A \in \mathrm{GL}_n(\mathbb{C})$ is the positive real number*

$$\mathrm{cond}_{\|\cdot\|}(A) = |||A||| \times \left|\left|\left|A^{-1}\right|\right|\right|.$$

Actually, this number provides a criterion to determine whether the matrix is “easy” to invert: the closer to 1, the easier the resolution of the linear system. When small changes occur in the linear system, we control the change on the solution thanks to the two following results.

Proposition 1.14 *Let $A \in \mathrm{GL}_n(\mathbb{C})$ and $\Delta A \in \mathscr{M}_n(\mathbb{C})$ be two matrices such that $A + \Delta A \in \mathrm{GL}_n(\mathbb{C})$. For $b \in \mathbb{C}^n \backslash \{0\}$, let x and y be respectively the solutions of the linear systems $Ax = b$ and $(A + \Delta A)y = b$. Then*

$$\frac{\|x - y\|}{\|y\|} \leq \mathrm{cond}_{\|\cdot\|}(A) \frac{|||\Delta A|||}{|||A|||}.$$

Proposition 1.15 *For $A \in \mathrm{GL}_n(\mathbb{C})$, $b \in \mathbb{C}^n \backslash \{0\}$ and $\Delta b \in \mathbb{C}^n$, let x and y be respectively the solutions of the linear systems $Ax = b$ and $Ay = b + \Delta b$. Then*

$$\frac{\|x-y\|}{\|x\|} \leq \mathrm{cond}_{\|\cdot\|}(A) \frac{\|\Delta b\|}{\|b\|}.$$

Iterative methods

The previous methods are called direct but they may not apply to any matrix. Another possibility is to use an iterative method, *i.e.* to build a sequence x^p converging to the solution of $Ax = b$ as p goes to ∞.

Definition 1.12 *An iterative method is the process of defining a sequence*

$$\begin{cases} x^0 \in \mathbb{C}^n, \\ \forall\, p \geq 0,\ x^{p+1} = M^{-1}\left(Nx^p + b\right). \end{cases}$$

M and N are two matrices such that $A = M - N$ and $M \in \mathrm{GL}_n(\mathbb{C})$.

Proposition 1.16 *The iterative method is convergent if* $\max\left\{|\lambda| : \lambda \in \mathrm{Sp}(M^{-1}N)\right\} < 1$.

1.3.4 Taylor series expansions

This paragraph is devoted to approximation theory for scalar real functions. The extension of those results to vector functions will be stated in **Chapter 3**.

Proposition 1.17 (Taylor–Young) *Let ϕ be a function in $\mathscr{C}^n([a,b],\mathbb{R})$. Then*

$$\phi(x) = \sum_{k=0}^{n} \frac{(x-a)^k}{k!}\phi^{(k)}(a) + o\big((x-a)^n\big).$$

If in addition $\phi \in \mathscr{C}^{n+1}([a,b],\mathbb{R})$, then the remainder is $\mathcal{O}\big((x-a)^{n+1}\big)$.

This result means that in the neighbourhood of a, function ϕ can be approximated by the polynomial $\sum_{k=0}^{n} \frac{\phi^{(k)}(a)}{k!}(x-a)^k$. Notice that for polynomials, this formula holds for any n and is exact (the remainder is equal to 0) for n greater that the degree of the polynomial.

In the general case, it can be interesting to specify the remainder in order to have a precise idea of the error.

Proposition 1.18 (Taylor–Lagrange) *Let ϕ be a function in $\mathscr{C}^{n+1}([a,b],\mathbb{R})$.*

$$\forall\, x \in [a,b],\ \exists\, c \in (a,x),\ \phi(x) = \sum_{k=0}^{n} \frac{(x-a)^k}{k!}\phi^{(k)}(a) + \frac{(x-a)^{n+1}}{(n+1)!}\phi^{(n+1)}(c).$$

Proposition 1.19 (Taylor with integral remainder) *Let ϕ be a function in $\mathscr{C}^{n+1}([a,b],\mathbb{R})$. Then*

$$\forall\, x \in [a,b],\ \phi(x) = \sum_{k=0}^{n} \frac{(x-a)^k}{k!}\phi^{(k)}(a) + \int_a^x \frac{(x-y)^n}{n!}\phi^{(n+1)}(y)\,\mathrm{d}y.$$

Example 1.1 *Applying* ***Prop. 1.17****, **1.18** and **1.19** to $\phi(t) = \exp t$, knowing that for all $k \in \mathbb{Z}_+$, $\phi^{(k)} = \phi$, we obtain for $a = 0$, $x = t$ and any n*

$$\exp t = \sum_{k=0}^{n} \frac{t^k}{k!} + \begin{cases} \mathcal{O}(t^{n+1}), \\ \dfrac{t^{n+1}}{(n+1)!} e^{c_t}, \quad c_t \in [0, t], \\ \displaystyle\int_0^t \frac{(t-\tau)^n}{n!} e^{\tau} \, \mathrm{d}\tau. \end{cases}$$

Chapter 2

Ordinary differential equations (ODE)

2.1 Introduction

An ordinary differential equation (ODE) is an equation whose unknown is a function and which relates the function to its derivatives. The general 1st–order case reads

$$\begin{cases} \hat{y}'(t) = f\big(t, \hat{y}(t)\big), & (2.1\text{a}) \\ \hat{y}(t_0) = y_0. & (2.1\text{b}) \end{cases}$$

The scalar variable t lies in a given open interval $I \subset \mathbb{R}$. The unknown is the function $\hat{y} : I \subset \mathbb{R} \longrightarrow \mathbb{R}^d$, $d \in \mathbb{Z}_+$. The Cauchy data are $t_0 \in I$ and $y_0 \in \mathbb{R}^d$.

Function f is a datum which depends on the scalar variable t and on the vector y. It satisfies the following hypothesis:[1]

$$f : I \times \mathbb{R}^d \longrightarrow \mathbb{R}^d \text{ is of class } \mathscr{C}^1 \text{ with respect to } (t, y). \tag{H}$$

Such differential problems arise in several domains like biology or physics. For instance, if f represents a velocity field, then $\hat{y}(t)$ is the location of a particle at time t which was located at y_0 at time t_0. Thus the curve $t \in I \mapsto \hat{y}(t)$ is the trajectory of the particle.

Given (f, t_0, y_0), the issue boils down to stating whether a solution exists and over which interval $J \subseteq I$. Then, an important matter consists in proving properties of solutions (positivity, monotonicity, ...) and even in deriving explicit expressions.

Indeed, some of problems (2.1) can be solved directly (see § 2.2). For other cases, we cannot provide an explicit expression for the solution. The art of numerical analysis then consists in computing an approximation of the solution with a control over the error by means of a computer and a numerical method.

2.2 Classical examples

The simplest ODEs are the **linear 1st–order ODE with constant coefficients**

$$\hat{y}'(t) = \alpha\, \hat{y}(t), \quad \hat{y}(t_0) = y_0, \tag{2.2}$$

[1] The classical results is stronger than this statement. It suffices to have f Lipschitz-continuous with respect to y: there exists a constant $C_f > 0$ such that $\forall\ t \in I$, $\forall\ (y, z) \in \mathbb{R}^d \times \mathbb{R}^d$, $\|f(t, y) - f(t, z)\| \leqslant C_f \|y - z\|$.

for some $\alpha \in \mathbb{R}$. This equation corresponds to (2.1) for $f(t,y) = f(y) = \alpha y$. The solution reads

$$\hat{y}(t) = y_0 \exp\big[\alpha(t-t_0)\big].$$

The **affine** case can be treated similarly ($\alpha \in \mathbb{R}^*$, $\beta \in \mathbb{R}$):

$$\begin{cases} \hat{y}'(t) = \alpha \hat{y}(t) + \beta, \\ \hat{y}(t_0) = y_0. \end{cases} \qquad \Longrightarrow \qquad \hat{y}(t) = \left(y_0 + \frac{\beta}{\alpha}\right)\exp\big[\alpha(t-t_0)\big] - \frac{\beta}{\alpha}.$$

It is also achievable to solve **linear 2nd–order ODE with constant coefficients**

$$\begin{cases} \alpha \hat{y}''(t) + \beta \hat{y}'(t) + \gamma \hat{y}(t) = 0, \\ \hat{y}(0) = \delta, \quad \hat{y}'(0) = \varepsilon, \end{cases} \tag{2.3}$$

where $(\alpha,\beta,\gamma) \in \mathbb{R}^3$, $\alpha \neq 0$. We are looking for a solution like $\hat{y}(t) = e^{rt}$ for a certain $r \in \mathbb{R}$. Then r satisfies the equation

$$\alpha r^2 + \beta r + \gamma = 0. \tag{2.4}$$

Three cases must be considered depending on the sign of the discriminant $\Delta = \beta^2 - 4\alpha\gamma$.

- If $\Delta > 0$, then Equation (2.4) has two real solutions r_1 and r_2. The solutions to ODE (2.3) have the form

  $$\hat{y}(t) = \eta e^{r_1 t} + \zeta e^{r_2 t}.$$

  Coefficients η and ζ are computed thanks to initial conditions (2.1b).

- If $\Delta = 0$, then Equation (2.4) has a unique solution r_0. ODE (2.3) has the following solution

  $$\hat{y}(t) = (\eta + \zeta t)e^{r_0 t}.$$

- If $\Delta < 0$, then Equation (2.4) has two complex solutions r_1 and $\overline{r_1}$. Hence the solutions to (2.4)

  $$\hat{y}(t) = e^{\mathrm{Re}(r_1)t}\left[\eta \cos\big(\mathrm{Im}(r_1)t\big) + \zeta \sin\big(\mathrm{Im}(r_1)t\big)\right].$$

We can also handle some variable coefficient cases. Let α and β two functions of t defined over an interval $[t_1, t_2]$. We then focus on the equation

$$\hat{y}'(t) = \alpha(t)\hat{y}(t) + \beta(t), \quad \hat{y}(t_0) = y_0. \tag{2.5}$$

The domain of existence of $\hat{y}$ is at most $[t_1, t_2]$. Let A be a primitive of α. Then the solution of Equation (2.5) is

$$\hat{y}(t) = e^{A(t)} \int_{t_0}^{t} \beta(\tau) e^{-A(\tau)}\, \mathrm{d}\tau + e^{A(t)-A(t_0)} y_0.$$

A last case which can be solved explicitly is the **autonomous case**, *i.e.* (2.1) when $f(t, y) = \psi(y)$ assuming that ψ is continuous and never vanishes (and thus does not change its sign). Then (2.1) may be written

$$\frac{\hat{y}'(t)}{\psi\big(\hat{y}(t)\big)} = 1 \qquad \Longrightarrow \qquad \Psi\big(\hat{y}(t)\big) - \Psi(y_0) = t - t_0,$$

where $\Psi'(y) = \frac{1}{\psi(y)}$. Hence

$$\hat{y}(t) = \Psi^{-1}\big(\Psi(y_0) + t - t_0\big).$$

Example 2.1 *Let us solve the following ODE:*

$$\begin{cases} \hat{y}'(t) = \hat{y}(t)^2 + 1, \\ \hat{y}(0) = 3. \end{cases}$$

It corresponds to the previous case with $\phi(t) = 1$ *and* $\psi(y) = y^2 + 1$*. Then* $\Phi(t) = t$*,* $\Psi(y) = \arctan y$ *and*

$$\arctan \hat{y}(t) = t + \arctan 3.$$

As $\tan$ *is the inverse function of* $\arctan$ *over* $(-\pi, \pi)$*, we have*

$$\hat{y}(t) = \tan(t + \arctan 3) \ \textit{mod}\ 2\pi.$$

2.3 Theoretical results

2.3.1 Vocabulary

Let us give some definitions of classical terms that will be used in the sequel.

Definition 2.1 *The* ***order*** *of ODE* (2.1) *corresponds to the highest degree of derivation in Equation* (2.1a).

Example 2.2 *Equation* (2.2) *is of first–order while* (2.3) *is of second–order. However, any* n*-th–order ODE can be written as a 1st–order providing all initial data are given at the same time* t_0*. For example,* (2.3) *reads* $\hat{Y}'(t) = A\hat{Y}(t)$ *analogous to* (2.2)*, with*

$$\hat{Y}(t) = \begin{pmatrix} \hat{y}(t) \\ \hat{y}'(t) \end{pmatrix} \qquad \textit{and} \qquad A = \begin{pmatrix} 0 & 1 \\ -\frac{\gamma}{\alpha} & -\frac{\beta}{\alpha} \end{pmatrix}.$$

However, if $\hat{y}'(0) = \varepsilon$ is replaced in (2.3) by $\hat{y}'(1) = \zeta$, then this is not a Cauchy problem anymore and it does not read as a 1st–order problem.

Definition 2.2 *ODE (2.1) is said to be **linear** if, for any solutions y_1 and y_2 to (2.1a) and any constants c_1 and c_2, then $c_1 y_1 + c_2 y_2$ is also a solution.*

Definition 2.3 *ODE (2.1) is said to be **autonomous** if function f does not depend on t.*

2.3.2 Existence and uniqueness

Theorem 2.1 (Cauchy-Lipschitz) *Let f be a function satisfying Hypothesis (H). Then for any data $(t_0, y_0) \in I \times \mathbb{R}^d$, there exists a unique maximal solution $y \in \mathscr{C}^1(J)$ to System (2.1) where $J \subset I$ is an open neighbourhood of t_0.*

The term *maximal* means that there does not exist any solution over an interval $K \supsetneq J$.

Proposition 2.1 *If f is of class $\mathscr{C}^k$ with respect to (t, y), then the unique solution y is of class $\mathscr{C}^{k+1}(J)$.*

Proposition 2.2 (Cauchy) *If f is continuous with respect to (t, y) and affine with respect to y, then the solution to (2.1) exists over the whole interval I.*

Proposition 2.3 *If the maximal solution $\hat{y}$ is bounded over the interval J, then $J = I$.*

These results do not provide any solution but ensure that there exists a solution. This is important from a modelling point of view insofar as the model is not absurd and leads to a solution. The fact that this solution has a physical meaning is another matter.

2.4 One–step methods

We are now tackling the resolution of ODEs like (2.1) for which we are not able to provide explicit solutions but to which we can apply **Theorem 2.1**, which ensures the existence of a solution $\hat{y}$ on an interval $J = (t_0, t_0 + \mathscr{T})$ for a certain $\mathscr{T} > 0$.

Even if nothing has to be done when we know exact solutions, Section 2.2 is worth of interest. Indeed, we are going in the sequel to work out several numerical techniques to approximate solutions to ODEs. The processing of simple ODEs for which exact solutions are provided will enable us to assess these techniques by means of **comparisons** between the numerical solution and the exact one.

We do not intend to provide approximations of the solution at any time[2] t but only at time samples (called **nodes** and noted $t^1 := t_0 < t^2 < t^3 < \ldots < t^N = t_0 + \mathscr{T}$) as **the memory of a computer is finite**. This process is called **discretization**. The successive time steps are noted $\Delta t^n = t^{n+1} - t^n$.

[2] Variable t does not necessarily represent time but so is it called in the sequel for clarity.

	Continuous	**Discrete**	**Numerical**
Problem	ODE	Induction relation	Loop
Variable	$t \in I$	$(t^n)_{n\geq 1}$	$n \in \{1,\dots,N\}$
Unknown	$\hat{y}$	$(y_n)_{n\geq 1}$	$Y(n)$, $n = 1,\dots,N$
Type	Function	Sequence	Vector

Table 2.1: Different approaches of ODEs

For the sake of simplicity, we consider a homogeneous[3] grid $t^n = t_0 + (n-1)\Delta t$ where the **time step** is equal to $\Delta t = \frac{\mathcal{T}}{N-1}$ for a fixed number of intervals N. The aim of this part is to conceive a method yielding a sequence of approximations of the values $\hat{y}(t^n)$. To do so, we have to deal with derivatives in (2.1) that we must approximate using only values of the unknown.

Considering results from the previous section, we have

$$\hat{y}'(t) = \begin{cases} \dfrac{\hat{y}(t+\Delta t) - \hat{y}(t)}{\Delta t} + \mathcal{O}(\Delta t), \\ \dfrac{\hat{y}(t) - \hat{y}(t-\Delta t)}{\Delta t} + \mathcal{O}(\Delta t), \\ \dfrac{\hat{y}(t+\Delta t) - \hat{y}(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2). \end{cases}$$

Each one provides a numerical scheme. For instance, if we consider the first line above, we have

$$f\big(t^n, \hat{y}(t^n)\big) = \hat{y}'(t^n) = \frac{\hat{y}(t^{n+1}) - \hat{y}(t^n)}{\Delta t} + \mathcal{O}(\Delta t).$$

If we “ignore” the error term, the equality does not hold rigorously anymore. That is how we construct the sequence y_n:[4]

$$f(t^n, y_n) = \frac{y_{n+1} - y_n}{\Delta t}.$$

Several concepts and examples are given in the sequel. The overall process of the numerical resolution of an ODE is summarized in Table 2.1.

The point of this procedure is that the larger N, the smaller Δt, the more accurate the numerical solution. This fact will be conceptualized in **Definition 2.6**. Hence, a user would like to set N as large as possible. But as N represents the number of iterations, increasing N makes the computational time larger. There is thus an equilibrium to find. Likewise, if the error goes to 0 as Δt decreases, the convergence is not the same for all schemes, which provides a criterion to choose a scheme rather than another (see **Definition 2.8**).

[3] Homogeneous means that $\Delta t^n = \Delta t$ for all n.

[4] This is the well-known explicit Euler scheme.

2.4.1 Main definitions

Definition 2.4 *A numerical scheme derived to solve* (2.1) *is said to be* ***one–step*** *if it can read*

$$\frac{y_{n+1}-y_n}{\Delta t} = \Phi(t^n, \Delta t, y_n). \tag{2.6}$$

Function $\Phi : I \times \mathbb{R}_+ \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$ is thus specific to each scheme.

Definition 2.5 *Scheme* (2.6) *is said to be* ***convergent*** *if*

$$\lim_{\Delta t \to 0} \max_n \left\| y_n - \hat{y}(t^n) \right\| = 0.$$

Hence the smaller Δt (*i.e.* the larger N), the better the approximation y_n of $\hat{y}(t^n)$.

Definition 2.6 *Scheme* (2.6) *is said to be* ***consistant*** *if*

$$\lim_{\Delta t \to 0} \max_n \left\| \frac{\hat{y}(t^{n+1}) - \hat{y}(t^n)}{\Delta t} - \Phi\big(t^n, \Delta t, \hat{y}(t^n)\big) \right\| = 0.$$

Definition 2.7 *Scheme* (2.6) *is said to be* ***stable*** *if, for any sequence* (ε_n) *in* $\mathbb{R}^d$ *and for* (z_n) *defined by* $z_0 \in \mathbb{R}^d$ *and the induction formula*

$$\frac{z_{n+1}-z_n}{\Delta t} = \Phi(t^n, \Delta t, z_n) + \varepsilon_n,$$

there exist some constants $M_1 \geq 0$ *et* $M_2 \geq 0$ *independent from* y_n *and* z_n *such that*

$$\max_n \left\| y_n - z_n \right\| \leqslant M_1 \| y_0 - z_0 \| + M_2 \sum_{k=0}^{n} \varepsilon_k.$$

The most important theorem is the following:

Theorem 2.2 (Lax-Richtmyer) *A numerical scheme is convergent if it is stable and consistant.*

Definitions 2.6 and 2.7 may be quite puzzling. That is why more practical characterizations are given:

Proposition 2.4 *Scheme* (2.6) *is consistant if and only if* $\Phi(t, 0, y) = f(t, y)$ *for all* $t \in I$ *and* $y \in \mathbb{R}^d$.

Proposition 2.5 *Scheme* (2.6) *is stable if and only if function* $y \longmapsto \Phi(t, \Delta t, y)$ *is Lipschitz-continuous for all* $t \in I$ *and* Δt *small enough.*

Finally, a quite useful notion is:

Definition 2.8 *Scheme* (2.6) *is of* ***order*** p *(*$p \in \mathbb{Z}_+$*) if*

$$\left\| \frac{\hat{y}(t^{n+1}) - \hat{y}(t^n)}{\Delta t} - \Phi\big(t^n, \Delta t, \hat{y}(t^n)\big) \right\| = \mathcal{O}(\Delta t^p).$$

Hence the larger p, the more accurate the scheme.

2.4.2 Classical schemes

Name	Formula	Order	Explicit
Forward Euler	$y_{n+1} = y_n + \Delta t f(t^n, y_n)$	1	Yes
Backward Euler	$y_{n+1} = y_n + \Delta t f(t^{n+1}, y_{n+1})$	1	No
Enhanced Euler	$y_{n+1} = y_n + \Delta t f\big(t^n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} f(t^n, y_n)\big)$	2	Yes
Heun	$y_{n+1} = y_n + \frac{\Delta t}{2}\big[f(t^n, y_n) + f\big(t^n + \Delta t, y_n + \Delta t f(t^n, y_n)\big)\big]$	2	Yes
Crank-Nicholson	$y_{n+1} = y_n + \frac{\Delta t}{2}\big[f(t^n, y_n) + f(t^{n+1}, y_{n+1})\big]$	2	No
Runge–Kutta 2 ($\lambda \in [0,1]$)	$y_{n+1} = y_n + \Delta t\big[(1-\lambda) f(t^n, y_n) + \lambda f\big(t^n + \frac{\Delta t}{2\lambda}, y_n + \frac{\Delta t}{2\lambda} f(t^n, y_n)\big)\big]$	2	Yes
Runge–Kutta 3	$y_{n+1} = y_n + \frac{\Delta t}{6}(k_1 + 4k_2 + k_3)$, $k_1 = f(t^n, y_n)$, $k_2 = f\big(t^n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} k_1\big)$, $k_3 = f\big(t^{n+1}, y_n + \Delta t(2k_2 - k_1)\big)$	3	Yes
Runge–Kutta 4	$y_{n+1} = y_n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$, k_1 and k_2 as for RK3, $k_3 = f\big(t^n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} k_2\big)$, $k_4 = f\big(t^{n+1}, y_n + \Delta t k_3\big)$	4	Yes

2.4.3 Summary

A numerical method to approximate solutions of (2.1) thus consists in constructing a sequence (y_n) such that

$$y_n = \hat{y}(t^n) + \mathcal{O}(\Delta t^p).$$

To derive a method, we used some Taylor expansions to obtain an approximation of $\hat{y}'(t^n)$ in

$$\hat{y}'(t^n) = f\big(t^n, \hat{y}(t^n)\big)$$

or an approximation of the integral in

$$\hat{y}(t^{n+1}) - \hat{y}(t^n) = \int_{t^n}^{t^{n+1}} f\big(\tau, \hat{y}(\tau)\big)\, \mathrm{d}\tau.$$

Once the approximation is chosen, the user has to ensure that the sequence is well defined (which may not be the case for implicit methods) by expressing y_{n+1} as a function of y_n. The sequence is initialized thanks to the initial condition (2.1b) by setting $y_1 = \hat{y}(t^1) = \hat{y}(t_0) = y_0$. Then, for a given scheme (*i.e.* a given way of constructing the sequence), Δt governs the accuracy. We say on Figure 2.1 the results for various N (and thus various Δt of the Forward Euler scheme applied to the ODE $\hat{y}' = -15\hat{y}$, $\hat{y}(0) = 1$. The smaller Δt, the closer the numerical solution to the exact solution.

2.5 Multi–step schemes

In the previous section, only two consecutive terms of the sequence (y_n) were involved in the scheme, namely y_{n+1} and y_n. There also exist multi–step schemes.

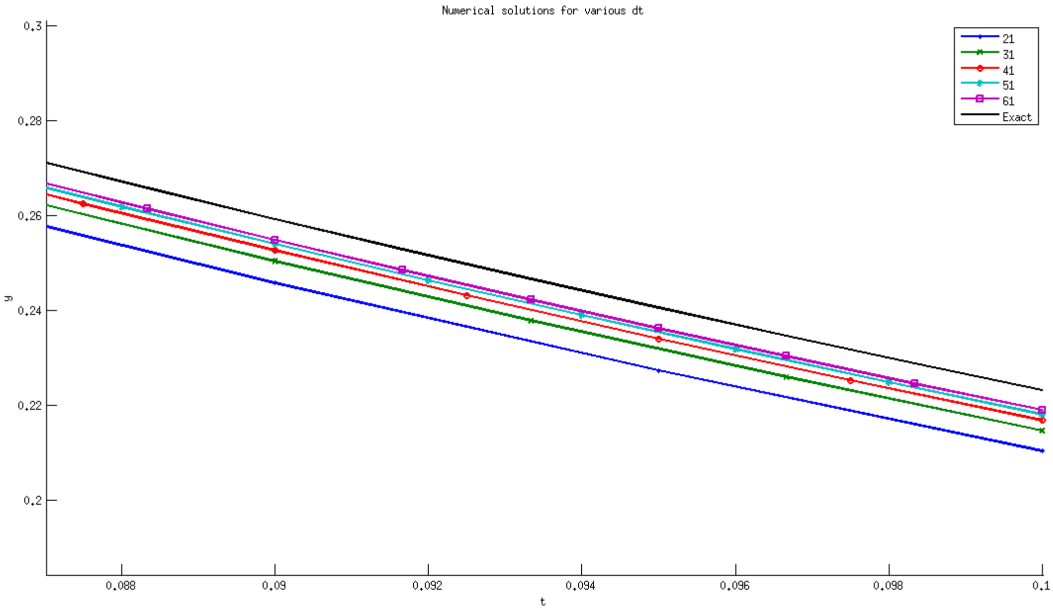

Figure 2.1: Numerical solutions obtained by means of the Forward Euler scheme for various Δt

Definition 2.9 *A numerical scheme aimed at solving* (2.1) *is said to be* ***multi–step*** *(with q steps) if there exist coefficients* $(\alpha_k)_{0\leq k\leq q}$ *and* $(\beta_k)_{0\leqslant k\leqslant q}$ *such that* $\alpha_q \neq 0$, $|\alpha_0| + |\beta_0| \neq 0$ *and*

$$\sum_{k=0}^{q} \alpha_k y_{n+k} = \Delta t \sum_{k=0}^{q} \beta_k f(t^{n+k}, y_{n+k}). \tag{2.7}$$

Notions of consistency, stability and order adapt to this new kind of schemes. The following results provide practical assessments.

Proposition 2.6 *Scheme* (2.7) *is consistant if and only if* $\sum_{k=0}^{q} \alpha_k = 0$ *and* $\sum_{k=0}^{q} k\alpha_k = \sum_{k=0}^{q} \beta_k$.

Proposition 2.7 (Dahlquist) *Let* ρ *be the polynomial*

$$\rho(x) = \sum_{k=0}^{q} \alpha_k x^k.$$

Scheme (2.7) *is stable if and only if the roots* ξ *of* ρ *satisfy* $|\xi| \leq 1$ *with* $|\xi| = 1$ *only if* ξ *is simple (*ρ *can be factorized by* $x - \xi$ *but not by* $(x-\xi)^2$*).*

Proposition 2.8 *Scheme* (2.7) *is of order* $p \geq 2$ *if and only if*

$$\forall\, i \in \{2, \ldots, p\},\ \sum_{k=0}^{q} k^i \alpha_k = i \sum_{k=0}^{q} k^{i-1} \beta_k.$$

Theorem 2.2 also applies to this situation.

Chapter 3

Partial differential equations (PDE)

This chapter is devoted to differential equations involving functions of multiple variables. These equations are called Partial Differential Equations (PDEs). Generally, variables represent time and space coordinates.

3.1 Reminder of functional analysis

Let f be a smooth function from $\mathbb{R}^d$ into $\mathbb{R}$. The variable is denoted by $\boldsymbol{x} \in \mathbb{R}^d$. Classical differential operators are:

- The k–**th partial derivative** noted $\dfrac{\partial f}{\partial x_k}$;
- The **gradient** which is the vector of all partial derivatives and noted

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{pmatrix};$$

- The **Hessian matrix**, which is a collection of all 2nd–order derivatives and whose coefficients are

$$[\mathrm{Hess}\, f]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j};$$

- The **Laplacian operator**, which is the trace of the Hessian matrix

$$\Delta f = \sum_{k=1}^{d} \frac{\partial^2 f}{\partial x_k^2}.$$

When $\boldsymbol{f} = (f_1, \ldots, f_d)$ is a function from $\mathbb{R}^d$ into $\mathbb{R}^d$, the **divergence** operator reads

$$\nabla \cdot \boldsymbol{f} = \sum_{k=1}^{d} \frac{\partial f_k}{\partial x_k}.$$

The notion of Taylor series expansion can be extended to multiple variable functions. For $\boldsymbol{h} \in \mathbb{R}^d$, we have

$$f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \sum_{k=1}^{d} h_k \frac{\partial f}{\partial x_k}(\boldsymbol{x}) + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} h_i h_j \frac{\partial^2 f}{\partial x_i \partial x_j}(\boldsymbol{x}) + o\left(\|\boldsymbol{h}\|^2\right),$$

or equivalently

$$f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \left\langle \nabla f(\boldsymbol{x}), \boldsymbol{h}\right\rangle + \frac{1}{2}\left\langle \operatorname{Hess} f(\boldsymbol{x})\boldsymbol{h}, \boldsymbol{h}\right\rangle + o\left(\|\boldsymbol{h}\|^2\right).$$

Symbols $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ denote the euclidian inner product and the corresponding norm on $\mathbb{R}^d$.

3.2 Analysis of PDEs

3.2.1 Classification of PDEs

PDEs can be distinguished through several parameters such as

1. order (highest degree of derivation) with respect to each variable;
2. linearity;
3. “nature”.

The nature of a PDE describes physical properties of solutions. It can be of three kinds:[1]

1. **hyperbolic equations** (transport equation, Euler equations, Saint-Venant equations, wave equation, ...); solutions are characterized by a finite propagation speed and the equations are well-posed under suitable boundary conditions (only on parts of the boundary where the characteristics come in);
2. **elliptic equations** (Poisson equation, Laplace equation, ...); unlike the previous kind, these equations have a regularizing effect with an infinite propagation speed;
3. **parabolic equations** (heat equation, Black & Scholes model, ...); they have similar properties to elliptic equations; boundary conditions are required on the whole boundary.

3.2.2 Boundary conditions (BC)

It is often necessary to get information from the boundary to supplement the equations. Indeed, if the equation in itself provides a global behaviour, it is not enough to compute the solution. For instance, if we consider the 1st–order problem

$$\partial_x u = t \quad \text{over} \quad (0,1),$$

we know that u takes the form $u(t,x) = tx + c(t)$ for some function c to be determined. Without information at $x = 0$ or $x = 1$, we cannot determine entirely the solution. Imposing $u(t,0) = t$ for example leads to $u(t,x) = t(x+1)$.

[1] The mathematical analysis of the nature of PDEs will not be detailed here.

	Continuous	**Discrete**	**Numerical**
Problem	PDE	Induction relation	Loops
Variable	$t \in I,\ x \in \Omega$	$(t^n)_{1\le i\le N_t},\ (x_i)_{1\le i\le N_x}$	$n \in \{1,\dots,N_t\},\ i \in \{1,\dots,N_x\}$
Unknown	u	$(u_i^n)_{1\le n\le N_t,1\le i\le N_x}$	$U(n,k),\ n=1,\dots,N_t, k=1,\dots,N_x$
Type	Function	Sequence	Matrix

Table 3.1: Different approaches of PDEs

Likewise, for the second–order problem

$$\partial^2_{xx} u = t,$$

we deduce that

$$u(t,x) = \frac{tx^2}{2} + c_1(t)x + c_2(t).$$

Hence, two boundary conditions are required to fix c_1 and c_2.

BC are quite natural in dimension 1 as they can be easily interpreted. In higher dimensions, they are as important but as the boundary is not a single point, they may be more tricky to impose.

There exists an infinite number of BCs. The most common BCs are:

- **Dirichlet boundary condition**: the unknown is imposed on a part of the boundary. For instance, $u(t,x=0)=0$ is of this type. If u represents the velocity of a fluid, this BC means that the fluid cannot slip over the boundary.

- **Neumann boundary condition**: the normal derivative of the unknown is imposed on a part of the boundary. This corresponds to the flux of the unknown through the boundary. It reads $\partial_x u(t,0)=0$ in 1D, $\nabla u \cdot \boldsymbol{n} = 0$ in higher dimensions, where $\boldsymbol{n}$ is the unit outward normal vector to the boundary.

- **Robin boundary condition**: it is a linear combination of the two previous ones.

3.3 The Finite Difference Method (FDM)

As stated earlier, this method relies on Taylor series expansions and can be summarized in Table 3.1.

3.3.1 Laplacian

The most classical example of PDEs is the Poisson equation in a bounded domain $\Omega \subset \mathbb{R}^d$:

$$\begin{cases} -\Delta u \overset{\Omega}{=} f, \\ u \overset{\partial\Omega}{=} g. \end{cases}$$

Functions f and g are some given functions on Ω and on its boundary $\partial\Omega$. Unknown u may for instance represent the velocity potential in fluid mechanics or the electric potential in electrostatics. There is an extensive literature about this equation and particularly about the regularity of its solutions.

Its one-dimensional version reads

$$\begin{cases} -u''(x) = f(x),\ x \in (0,1), & \text{(3.1a)} \\ u(0) = \gamma, \quad u(1) = \delta. & \text{(3.1b)} \end{cases}$$

System (3.1) is not a Cauchy problem since all BCs are not given at the same point. Thus we cannot state the existence of a solution through **Theorem 2.1**. However, we can exhibit a solution. By means of 2 successive integrations, we obtain

$$u(x) = \gamma + \left(\delta - \gamma + \int_0^1 (1-z) f(z)\,\mathrm{d}z\right) x - \int_0^x (x-z) f(z)\,\mathrm{d}z.$$

The Cauchy counterpart is for example

$$\begin{cases} -u''(x) = f(x),\ x \in (0,1), & \text{(3.2a)} \\ u(0) = \gamma, \quad u'(0) = \varepsilon, & \text{(3.2b)} \end{cases}$$

whose solution is

$$u(x) = \gamma + \varepsilon x - \int_0^x (x-z) f(z)\,\mathrm{d}z.$$

However, even if the exact solutions are known in both cases, it may be difficult to compute them. Indeed, the integrals appearing in both solutions may not be computable for some tricky f. That is why it seems necessary to develop methods to provide good approximations of u.

To do so, we will apply the **Finite Difference Method** on a homogeneous cartesian grid. Let N be a positive integer, $\Delta x = \frac{1}{N-1}$ the mesh size, $x_k = (k-1)\Delta x \in (0,1)$ for $k \in \{1,\dots,N\}$ the mesh nodes. The purpose of the method is to construct a vector

$$U = \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_N \end{pmatrix} \approx \begin{pmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_N) \end{pmatrix}.$$

As there are some derivatives of u appearing in (3.1a) and (3.2a), we have to approach u'' using only values of u. That is why this method relies on the Taylor expansions. Indeed, we can verify that

$$u''(x) = \frac{u(x-\Delta x) - 2u(x) + u(x+\Delta x)}{\Delta x^2} + \mathcal{O}(\Delta x^2),$$

which reads for $x = x_k$, $k \in \{2,\dots,N-1\}$

$$u''(x_k) = \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{3.3}$$

This equality holds rigorously. To construct U, we neglect the error term $\mathcal{O}(\Delta x^2)$ and we replace $u(x_k)$ by U_k. This strategy applied to (3.1a) and (3.2a) leads to

$$-U_{k-1} + 2U_k - U_{k+1} = \Delta x^2 f_k, \tag{3.4}$$

where f_k is equal to $f(x_k)$ or is an approximation of $f(x_k)$. (3.4) is an induction relation (linear, 2nd–order) which defines the sequence U_k. The sequence has though to be initialized. This can be done much more easily for (3.2) than for (3.1). In the former case, we have directly

$$U_1 = u(x_1) = \gamma.$$

Then, due to some Taylor expansion

$$u(x_2) = u(x_1) + \Delta x u'(x_1) + \mathcal{O}(\Delta x^2),$$

we choose to take

$$U_2 = \gamma + \varepsilon \Delta x.$$

From (3.4), we have

$$U_1 = \gamma,\ U_2 = \gamma + \varepsilon \Delta x,\ \forall\ k \geq 1,\ U_{k+1} = 2U_j - U_{k-1} - \Delta x^2 f_k.$$

By induction, we can compute all components of U.

In the latter case which is (3.1), the calculations are different. BCs (3.1b) yield

$$U_1 = \gamma,\ U_N = \delta.$$

As U_2 is not known, we cannot compute $U_3, U_4, \ldots$ and the sequence is implicit. Then, the linear system

$$\begin{cases} U_1 = \gamma \\ -U_1 + 2U_2 - U_3 = \Delta x^2 f_2 \\ \quad\vdots \\ -U_{N-2} + 2U_{N-1} - U_N = \Delta x^2 f_{N-1} \\ U_N = \delta \end{cases}$$

reads as a matrix equation

$$\mathbf{A}U = \mathbf{f}, \quad \text{where} \quad \mathbf{f} = \begin{pmatrix} \gamma \\ \Delta x^2 f_2 \\ \vdots \\ \Delta x^2 f_{N-1} \\ \delta \end{pmatrix} \quad \text{and} \quad \mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

There exist several methods (direct, iterative, ...) to solve linear systems (see § 1.3.3). However, BCs made the matrix nonsymmetric which may cause extra computational costs. However BCs can be taken into account differently. Scheme (3.4) for $k = 2$ can be written

$$2U_2 - U_3 = \Delta x^2 f_2 + U_1 = \Delta x^2 f_2 + \gamma.$$

Likewise

$$2U_{N-1} - U_{N-2} = \Delta x^2 f_{N-1} + \delta.$$

Hence U_1 and U_N are not considered as unknowns anymore and the linear system is smaller:

$$\tilde{\mathbf{A}}\tilde{U} = \tilde{\mathbf{f}}, \quad \text{where} \quad \tilde{U} = \begin{pmatrix} U_2 \\ \vdots \\ U_{N-1} \end{pmatrix}, \quad \tilde{\mathbf{f}} = \begin{pmatrix} \Delta x^2 f_2 + \gamma \\ \vdots \\ \Delta x^2 f_{N-1} + \delta \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{A}} = \begin{pmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix}.$$

Even if we managed to fasten the resolution of the linear system, the fact remains that solving (3.1) is more expensive than solving (3.2) from a computational point of view since it requires the inversion of a matrix at each time iteration.

3.3.2 Transport equation

The previous paragraph highlighted the influence of BCs on the efficiency of the scheme. However, variables played the same role. That is why we focus in this section on another equation where time and space variables induce additional numerical difficulties.

Given a constant velocity field u_0 and an initial datum $Y_0(x)$, we consider the 1D hyperbolic transport equation

$$\begin{cases} \partial_t \hat{Y}(t,x) + u_0 \partial_x \hat{Y}(t,x) = 0, \quad t \geq 0,\ x \in (0,1), & \text{(3.5a)} \\ \hat{Y}(0,x) = Y_0(x), \quad x \in (0,1). & \text{(3.5b)} \end{cases}$$

The unknown is function $\hat{Y}(t,x)$. This equation is a perfect illustration of numerical issues as it will be emphasized below.

Exact solution As a hyperbolic equation, (3.5) may require boundary conditions. As it is suggested by the name of the equation, function $\hat{Y}$ is transported at velocity u_0. But we need additional information where the velocity field comes in the domain: if $u_0 > 0$ (which will be assumed in the sequel[2]), we must impose a BC at $x = 0$. We thus supplement (3.5) with

$$\hat{Y}(t,0) = \mathscr{Y}(t), \tag{3.5c}$$

where $\mathscr{Y}$ is given. The exact solution to (3.5) is

$$\hat{Y}(t,x) = \begin{cases} Y_0(x - u_0 t), & \text{if } x - u_0 t \geq 0, \\ \mathscr{Y}\left(t - \dfrac{x}{u_0}\right), & \text{otherwise.} \end{cases} \tag{3.6}$$

[2] Otherwise, a BC is required at $x = 1$.

Finite differences Set $\mathscr{T} = 1$ the final time. Let us discretized the spatio-temporal domain $[0,1] \times [0,1]$. To do so, let $N_x \geq 2$ be the number of space intervals and $N_t \geq 2$ the number of time iterations (to be specified later). The mesh is given by

$$t^n = (n-1)\Delta t,\ 1 \leq n \leq N_t, \qquad x_i = (i-1)\Delta x,\ 1 \leq i \leq N_x,$$

where the time step is equal to $\Delta t = \frac{1}{N_t - 1}$ and the mesh size to $\Delta x = \frac{1}{N_x - 1}$.

In the spirit of Section 2.4, the idea is to replace every derivative in (3.5a) by approximation formulae in order to construct the sequence Y_i^n which approximates $\hat{Y}(t^n, x_i)$. Here are some natural attempts:

- *Upwind scheme*:
$$\frac{Y_i^{n+1} - Y_i^n}{\Delta t} + u_0 \frac{Y_i^n - Y_{i-1}^n}{\Delta x} = 0; \tag{3.7a}$$
- *Downwind scheme*:
$$\frac{Y_i^{n+1} - Y_i^n}{\Delta t} + u_0 \frac{Y_{i+1}^n - Y_i^n}{\Delta x} = 0; \tag{3.7b}$$
- *Implicit scheme*:
$$\frac{Y_i^{n+1} - Y_i^n}{\Delta t} + u_0 \frac{Y_i^{n+1} - Y_{i-1}^{n+1}}{\Delta x} = 0; \tag{3.7c}$$
- *Centered scheme*:
$$\frac{Y_i^{n+1} - Y_i^n}{\Delta t} + u_0 \frac{Y_{i+1}^n - Y_{i-1}^n}{2\Delta x} = 0. \tag{3.7d}$$

To compare these schemes and choose the "best" one, we get interested in qualitative properties like consistency, stability and order. **Definitions 2.6**, **2.7** and **2.8** adapt naturally to the multiple variable case. For instance, to study the consistency of schemes (3.7), we replace Y_i^n by $\hat{Y}(t^n, x_i)$ into the schemes and we use Taylor expansions to determine the error as $\mathcal{O}(\Delta x^k + \Delta t^p)$.

As for stability, it depends on the norm used in **Def. 2.7**. For instance, the L^∞ stability corresponds to the $\|\cdot\|_\infty$ norm and means that the numerical solution is bounded at all iterations. Scheme (3.7a) can be rewritten as

$$Y_i^{n+1} = \lambda Y_{i-1}^n + (1-\lambda) Y_i^n, \qquad \lambda = \frac{u_0 \Delta t}{\Delta x}.$$

Hence, through a barycentric analysis, we prove that the upwind scheme is L^∞–stable if $\lambda \in [0,1]$. Indeed, under that hypothesis, if *e.g.* $Y_{i-1}^n < Y_i^n$, then $Y_i^{n+1} \in (Y_{i-1}^n, Y_i^n)$ and the numerical solution cannot increase (or decrease) infinitely. Similar analyses can be performed for each scheme using mathematical techniques (Van Neumann, ...). Condition $\lambda \in [0,1]$ implies a constraint upon the time step which reads

$$\Delta t \leq \frac{\Delta x}{u_0}. \tag{3.8}$$

This expresses the fact that the numerical propagation speed must be smaller than the real velocity field. Such stability constraints are usually named the CFL condition.[3]

[3] CFL stands for *Courant*, *Friedrichs* and *Lewy* referring to the authors of the pioneering article on that topic.

Properties of schemes (3.7) are detailled below:

- Upwind scheme (3.7a) is consistant up to order 1 in space and time, and L^{∞}–stable under (3.8);
- Downwind scheme (3.7b) is consistant up to order 1 in space and time, and unconditionally unstable;
- Implicit scheme (3.7c) is consistant up to order 1 in space and time, and unconditionally stable;
- Centered scheme (3.7d) is consistant up to order 1 in time and 2 in space, and unconditionally unstable.

We infer that the combination of reasonable formulae for approximating derivatives may not lead to an efficient scheme. Indeed, the fact that two parameters (Δt and Δx) are involved may require compatibility conditions to ensure stability. As stated by Ralston & Rabinowitz, only practice can bring to the user the expertise to make the right choices of approximation formulae. On the one hand, we notice (3.8) shows that if we refine the spatial mesh, we must automatically refine the time mesh (linearly). On the other hand, we should bear in mind that accuracy corresponds to small parameters. Hence, even if scheme (3.7c) is unconditionally stable, we must take Δx and Δt small enough (independently from each other) to provide an accurate solution.

3.3.3 Black & Scholes model

As we assessed some numerical methods for close problems, we now focus on the simulation of the Black & Scholes model.

A first issue is to determine which formulation is the most relevant from a numerical point of view. As derived in Section 1.2, we have the original formulation (1.2), the time reversed formulation (1.5), the logarithmic-price version (1.6), those three being valid even for variable volatility and rate interest, and the heat equation (1.7) whose equivalence only holds for constant parameters.

The second issue is to take into account the fact that we can only deal with finite domains while the price S ranges from 0 to $+\infty$ and the logarithmic price x from $-\infty$ to $+\infty$. We thus have to truncate the domain and consequently to derive new boundary conditions for the new boundaries.

The third issue consists in evaluating the performances of each scheme especially in regard to the CPU time and the accuracy.

Heat equation We first present numerical schemes for the heat equation (1.7). The domain is restricted to $[\underline{x}, \overline{x}]$ where $\underline{x} \ll \ln K \ll \overline{x}$, $\underline{x} < 0$, $\overline{x} > 0$. We then have to impose boundary conditions at $\underline{x}$ and $\overline{x}$. This choice depends on the option (call or put) we are interested in, let us say a put. Remind that

$$\psi(\theta, x) = P(\mathcal{T} - \theta, e^{x})e^{-a\theta - bx}.$$

From (1.4c), we choose to impose

$$\psi(\theta, \overline{x}) = 0. \tag{3.9a}$$

Likewise, we take

$$\psi(\theta,\underline{x}) = Ke^{-(a+r_0)\theta - b\underline{x}}. \tag{3.9b}$$

Given $N_x \geq 2$, we consider a uniform spatio-temporal mesh

$$x_i = \underline{x} + (i-1)\Delta x,\ 1 \leq i \leq N_x,\ \Delta x = \frac{\overline{x}-\underline{x}}{N_x - 1}, \qquad \theta^n = (n-1)\Delta\theta,\ 1 \leq n \leq N_\theta,\ \Delta\theta = \frac{\mathscr{T}}{N_\theta - 1}$$

for some N_θ deduced from stability considerations. A natural explicit scheme reads

$$\begin{cases} \dfrac{\psi_i^{n+1}-\psi_i^n}{\Delta\theta} - \dfrac{\sigma_0^2}{2}\dfrac{\psi_{i+1}^n - 2\psi_i^n + \psi_{i-1}^n}{\Delta x^2} = 0, & 2 \leq i \leq N_x - 1, \\ \psi_1^{n+1} = Ke^{-(a+r_0)\theta^{n+1} - b\underline{x}}, \\ \psi_{N_x}^{n+1} = 0. \end{cases} \tag{3.10}$$

Scheme (3.10) is of order 1 in time, 2 in space and is stable under condition

$$\Delta\theta \leq \frac{\Delta x^2}{\sigma_0^2/2}$$

which is very prohibitive as the number of iterations must be like the square of the number of nodes. An alternative is to choose the implicit scheme

$$\begin{cases} \dfrac{\psi_i^{n+1}-\psi_i^n}{\Delta\theta} - \dfrac{\sigma_0^2}{2}\dfrac{\psi_{i+1}^{n+1} - 2\psi_i^{n+1} + \psi_{i-1}^{n+1}}{\Delta x^2} = 0, & 2 \leq i \leq N_x - 1, \\ \psi_1^{n+1} = Ke^{-(a+r_0)\theta^{n+1} - b\underline{x}}, \\ \psi_{N_x}^{n+1} = 0, \end{cases}$$

or equivalently

$$\begin{pmatrix} 2\left(1+\frac{\Delta x^2}{\sigma_0^2\Delta\theta}\right) & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2\left(1+\frac{\Delta x^2}{\sigma_0^2\Delta\theta}\right) \end{pmatrix} \Psi^{n+1} = \frac{2\Delta x^2}{\sigma_0^2\Delta\theta}\Psi^n + \begin{pmatrix} Ke^{-(a+r_0)\theta^{n+1}-b\underline{x}} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where $\Psi^n = (\psi_2^n, \ldots, \psi_{N_x-1}^n)^T$. The inversion is time-consuming but the scheme is always stable. However, as we are interested in pricing the option *i.e.* evaluating

$$P(0,S) = \psi(\mathscr{T}, \ln S)e^{a\mathscr{T}}S^b,$$

we have to go back to the primitive variable P.

But the uniform mesh in x will not be uniform in S anymore and will even be quite distorted: the resulting (exponential) mesh will be fine close to $S=0$ but very coarse for large S. That is why we can think of using a nonuniform mesh in x (corresponding to a uniform mesh in S) which makes formulae more complex: considering a nonuniform grid x_i with mesh sizes $\Delta x_i = x_{i+1} - x_i$, Scheme (3.10) is replaced by

$$\begin{cases} \dfrac{\psi_i^{n+1} - \psi_i^n}{\Delta\theta} - \sigma_0^2 \left(\dfrac{\psi_{i+1}^n}{\Delta x_i(\Delta x_{i-1} + \Delta x_i)} - \dfrac{2\psi_i^n}{\Delta x_{i-1}\Delta x_i} + \dfrac{\psi_{i-1}^n}{\Delta x_{i-1}(\Delta x_{i-1}+\Delta x_i)} \right) = 0, \qquad 2 \le i \le N_x - 1, \\ \psi_1^{n+1} = Ke^{-(a+r_0)\theta^{n+1} - b\underline{x}}, \\ \psi_{N_x}^{n+1} = 0. \end{cases} \tag{3.11}$$

which yields a nonsymmetric matrix.

We can also think of simulating directly the primitive formulation (1.5). We propose the following implicit scheme over a uniform grid for S

$$-\frac{V_i^{n+1} - V_i^n}{\Delta\theta} + \frac{\sigma_0^2}{2} S_i^2 \frac{V_i^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}}{\Delta S^2} + r_0 S_i \frac{V_{i+1}^{n+1} - V_{i-1}^{n+1}}{2\Delta S} - r_0 V_i^{n+1} = 0.$$

The resulting matrix is however not symmetric due to the variable coefficients of the PDE. Moreover, there is a constraint upon $\Delta\theta$ which must be smaller than a constant depending on σ_0^2 and r_0.

3.4 The Finite Element Method (FEM)

The FD method relies on a smoothness property of the solution (*i.e.* to apply Taylor expansions). To deal with less regular cases, another numerical method has been developed. Finite Element Methods are based on integral formulations of the underlying PDE. We only give a brief overview about this method which allows for general geometries.

The functional spaces involved in FEM are

$$\mathrm{L}^2(\Omega) = \left\{ f : \Omega \longrightarrow \mathbb{R} \middle| \int_\Omega |f(x)|^2 \,\mathrm{d}x < \infty \right\}, \qquad \mathrm{H}^1(\Omega) = \left\{ f \in \mathrm{L}^2(\Omega) \middle| \nabla f \in \left(\mathrm{L}^2(\Omega)\right)^2 \right\}.$$

In particular, $\mathrm{H}_0^1(\Omega)$ denotes the set of functions in $\mathrm{H}^1(\Omega)$ whose trace is 0 on the boundary $\partial\Omega$ of Ω. The scalar product over $\mathrm{L}^2(\Omega)$ is given by

$$\langle f, g \rangle_{\mathrm{L}^2} := \int_\Omega f(x) g(x) \,\mathrm{d}x.$$

To simplify the formulation, we first comes down to homogeneous Dirichlet conditions (indeed, BC (3.9) are not homogeneous) by setting

$$\Psi(\theta, x) = \psi(\theta, x) - Ke^{-(a+r_0)\theta - b\underline{x}} \frac{\overline{x} - x}{\overline{x} - \underline{x}}.$$

This function satisfies

$$\begin{cases} \dfrac{\partial \Psi}{\partial \theta} - \dfrac{\sigma_0^2}{2}\dfrac{\partial^2 \Psi}{\partial x^2} = f(\theta, x), \\ \Psi(\theta, \underline{x}) = \Psi(\theta, \overline{x}) = 0, \\ \Psi(0, x) = \max\{K - e^x, 0\}e^{-bx} - Ke^{-b\underline{x}}\dfrac{\overline{x} - x}{\overline{x} - \underline{x}}, \end{cases} \tag{3.12}$$

where $f(\theta, x) := (a + r_0)Ke^{-(a+r_0)\theta - b\underline{x}}\dfrac{\overline{x} - x}{\overline{x} - \underline{x}}$.

Let χ be a test-function in $\mathrm{H}_0^1(\underline{x}, \overline{x})$. Then multiplying the equation by χ and integrating over $\Omega = (\underline{x}, \overline{x})$, we obtain

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\theta}\int_\Omega \Psi(\theta, x)\chi(x)\,\mathrm{d}x &= \frac{\sigma_0^2}{2}\int_\Omega \frac{\partial^2 \Psi}{\partial x^2}(\theta, x)\chi(x)\,\mathrm{d}x + \int_\Omega f(\theta, x)\chi(x)\,\mathrm{d}x \\ &= -\frac{\sigma_0^2}{2}\int_\Omega \frac{\partial \Psi}{\partial x}(\theta, x)\frac{\partial \chi}{\partial x}(x)\,\mathrm{d}x + \int_\Omega f(\theta, x)\chi(x)\,\mathrm{d}x. \end{aligned} \tag{3.13}$$

If Ψ satisfies (3.13) for all $\chi \in \mathrm{H}_0^1(\underline{x}, \overline{x})$, then Ψ is called a *weak* solution of the heat equation and (3.13) is the *weak formulation* of the heat equation. It can be proven[4] that there exists a unique weak solution to this problem.

The weak formulation can be written as

$$\text{Find } \Psi \text{ such that:} \qquad \forall\, \chi \in \mathrm{H}_0^1(\underline{x}, \overline{x}),\ \frac{\mathrm{d}}{\mathrm{d}\theta}\langle \Psi(\theta, \cdot), \chi\rangle_{\mathrm{L}^2} = -\mathscr{A}\big(\Psi(\theta, \cdot), \chi\big) + \langle f(\theta, \cdot), \chi\rangle_{\mathrm{L}^2},$$

where

$$\mathscr{A}(f, g) := \frac{\sigma_0^2}{2}\int_\Omega \frac{\partial f}{\partial x}\frac{\partial g}{\partial x}\,\mathrm{d}x.$$

The FEM consists in constructing an approximate solution to the weak formulation by replacing the test space H_0^1 by a vector space $H_N \subset \mathrm{H}_0^1$ with finite dimension N. The approximate problem now reads

$$\text{Find } \widehat{\Psi} \in \mathscr{C}^0\big([0, \mathscr{T}], H_N\big) \text{ such that:} \qquad \forall\, \chi \in H_N,\ \frac{\mathrm{d}}{\mathrm{d}\theta}\langle \widehat{\Psi}(\theta, \cdot), \chi\rangle_{\mathrm{L}^2} = -\mathscr{A}\big(\widehat{\Psi}(\theta, \cdot), \chi\big) + \langle f(\theta, \cdot), \chi\rangle_{\mathrm{L}^2}.$$

Let $(\chi_1, \ldots, \chi_N)$ a basis of H_N. Then the previous formulation is equivalent to

$$\text{Find } \widehat{\Psi} \in \mathscr{C}^0\big([0, \mathscr{T}], H_N\big) \text{ such that:} \qquad \forall\, i \in \{1, \ldots, N\},\ \frac{\mathrm{d}}{\mathrm{d}\theta}\langle \widehat{\Psi}(\theta, \cdot), \chi_i\rangle_{\mathrm{L}^2} = -\mathscr{A}\big(\widehat{\Psi}(\theta, \cdot), \chi_i\big) + \langle f(\theta, \cdot), \chi_i\rangle_{\mathrm{L}^2}.$$

We decompose $\widehat{\Psi}(\theta, x) = \sum_{k=1}^N \phi_k(\theta)\chi_k(x)$. Hence the problem comes down to

$$\begin{aligned} &\text{Find } (\phi_1, \ldots, \phi_N) \in \mathscr{C}^0\big([0, \mathscr{T}]\big)^N \text{ such that:} \\ &\qquad \forall\, i \in \{1, \ldots, N\},\ \sum_{k=1}^N \phi_k'(\theta)\langle \chi_k, \chi_i\rangle_{\mathrm{L}^2} = -\sum_{k=1}^N \phi_k(\theta)\mathscr{A}\big(\chi_k, \chi_i\big) + \langle f(\theta, \cdot), \chi_i\rangle_{\mathrm{L}^2}, \end{aligned}$$

which is nothing but a linear ODE

$$M\Phi'(\theta) = -A\Phi(\theta) + F$$

[4] The Lax-Milgram theory is not the topic of this course and will thus not be developed in the sequel.

where M and A are matrices with entries $M_{ik} = \langle \chi_i, \chi_k \rangle_{\mathrm{L}^2}$ and $A_{ik} = \mathscr{A}(\chi_i, \chi_k)$ and F is the vector with components $F_k = \langle f(\theta, \cdot), \chi_k \rangle_{\mathrm{L}^2}$. As the family χ is a basis, M is invertible. We can thus apply any numerical scheme dedicated to ODEs (see the previous chapter). For instance, the forward Euler scheme yields

$$M\Phi^{n+1} = (M - \Delta t A)\Phi^n + \Delta t F^n.$$

The choice of the basis largely influences the computational efficiency. Indeed, if ϕ is orthonormal in L^2, then matrix M is diagonal!

The stability of such methods provides in 1D similar CFL conditions as for the FDM. However, it is very efficient in higher dimensions, *e.g.* for a basket of several underlyings.