Lecture Notes on

COMPUTATIONAL PHYSICS
and
NUMERICAL APPROACHES FOR PDEs

Emmanuel Dormy

Ecole Normale Supérieure, 2017-2018

Chapter 1

Where the troubles begin...

1.1 Introduction

Most of the laws of macroscopic physics (and at any rate, all laws considered in these notes) are expressed in terms of partial differential equations (PDEs). These relate physical quantities to their derivatives in space and in time.

When possible, physicists will most often seek analytical solutions to these equations. However in most cases such solutions do not exist... One then has to resort to trying to get apporached solutions using a computer.

Partial differential equations can model many types of applications: physics of materials, fluid mechanics, electromagnetism ... granular media are an exception as they obey different rules. One can sometimes find analytical solutions of PDEs. However most of the time there are no analytical solutions to the governing PDEs. When this is the case, one must resort to seeking approximate solutions on machines which, unfortunately, do not like "infinity". This raises two issues:

- First, functions correspond to an infinite number of degrees of freedom (or unknown). In order to derive a problem involving only a finite number of unknown, it is necessary to "discretise" the problem.
- Even once discretised, the equations involve real numbers... In order to represent these numbers using only a finite amount of information ("bits" in a computer) one must introduce rounding errors.

Let us consider in somewhat formal manner the transition from a continuous to a discrete problem. We can formally express our original continuous problem in the form

$$\begin{cases} L(\boldsymbol{u}) = \boldsymbol{g} & \text{in} \quad \Omega\,, \\ B(\boldsymbol{u}) = \boldsymbol{\gamma} & \text{on} \quad \partial\Omega\,, \end{cases} \tag{1.1}$$

where $\boldsymbol{u}$ stands for the unknown function. L and B representing differential operators respectively corresponding to the PDE in the domain of interest and the boundary conditions (which may also involve partial derivatives).

The discrete problem can then be expressed in the form

$$\begin{cases} L_h(\boldsymbol{u}_h) = \boldsymbol{g} & \text{in} \quad \Omega\,, \\ B_h(\boldsymbol{u}_h) = \boldsymbol{\gamma} & \text{on} \quad \partial\Omega\,, \end{cases} \tag{1.2}$$

where the subscript h identifies the typical discretisation step.
As we will ponder on the many possible ways to construct a discrete problem, a few questions should guide our thinking:

- Does L_h offer a good approximation to L ?
  In more mathematical terms, does $L_h \underset{h\to 0}{\longrightarrow} L$?
- Does $\boldsymbol{u}_h$ offer a good approximation to $\boldsymbol{u}$?
  Does $\boldsymbol{u}_h \underset{h\to 0}{\longrightarrow} \boldsymbol{u}$?

The first concern refers to a notion we will call *consistancy*, while the second will be named *convergence*.

The truncation error can then be defined as

$$R_h(\boldsymbol{u}) = L_h(\boldsymbol{u}) - L(\boldsymbol{u}) = L_h(\boldsymbol{u}) - \boldsymbol{g}\,. \tag{1.3}$$

Another way to express consistancy is then $R_h(\boldsymbol{u}) \underset{h\to 0}{\longrightarrow} 0\,.$

The distinction between consistancy and convergence may seem a bit technical at this stage. The reader may even wonder whether it was really necessary to introduce two notions. We will soon find out that the distinction between these notions is quite essential and that they should not be confused.

1.2 A first example

We will consider a problem of thermal convection. This can be studied experimentally and we will try to build a numerical model.

Considering the Fourier law

$$\boldsymbol{q} = -k\boldsymbol{\nabla}T \quad \text{where } k \text{ is the thermal conductivity,} \tag{1.4}$$

the temporal evolution of the energy is

$$\rho\frac{\partial e}{\partial t} = -\boldsymbol{\nabla}\cdot\boldsymbol{q} \quad \text{where } e = cT \text{ and } c \text{ is the calorific capacity.} \tag{1.5}$$

Assuming k to be constant in space and introducing $\kappa = k/\rho c$, one gets

$$\frac{\partial T}{\partial t} = \kappa\,\Delta T\,. \tag{1.6}$$

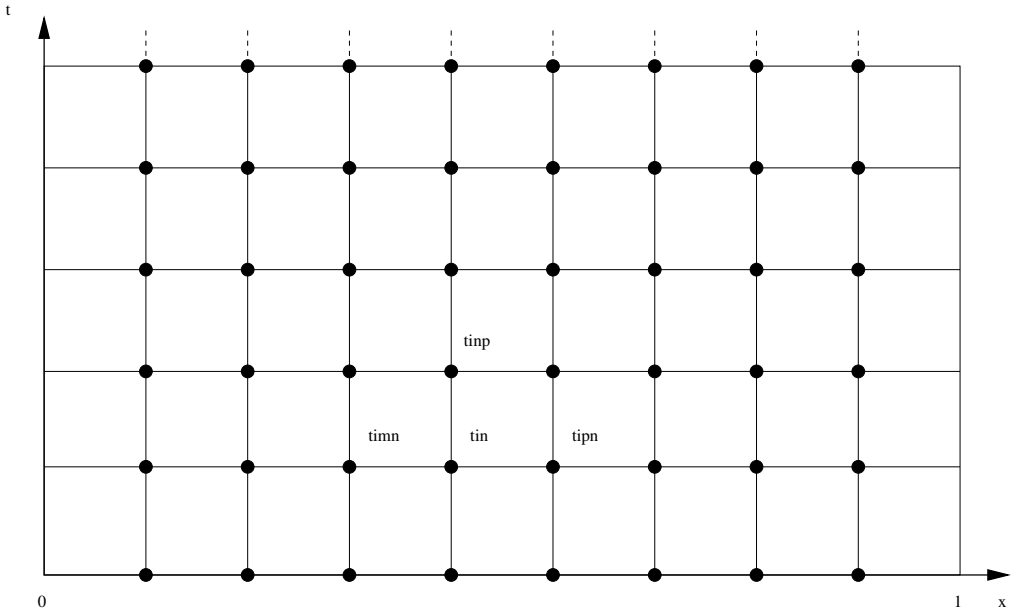


Figure 1.1: Regular grid in space and time.

We can further simplify the problem by modeling a one spatial dimension problem (the vertical dimension). Consider first that the temperature is kept constant at both sides. We can then write for $x \in [0, 1]$

$$\begin{cases} \dfrac{\partial T}{\partial t} = \kappa \dfrac{\partial^2 T}{\partial x^2}\,, \\ T = 0\,, \qquad \text{at } x = 0 \text{ and } x = 1,\, \forall t\,, \\ \\ T = \sin(2\pi\, x)\,, \qquad \text{at } t = 0\,. \end{cases} \tag{1.7}$$

If one discretises time using a step Δt and space using a step Δx, each unknown can be identified using its indices j in space and n in time. If N is the number of unknowns in space (values at $x = 0$ and $x = 1$ are known), we have $\Delta x = 1/(N + 1)$ and $x = j\,\Delta x$, $t = n\,\Delta t$.

We can intuitively derive the following scheme

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \kappa\, \frac{\frac{T_{j+1}^n - T_j^n}{\Delta x} - \frac{T_j^n - T_{j-1}^n}{\Delta x}}{\Delta x}\,, \tag{1.8}$$

which we can rewrite

$$T_j^{n+1} = T_j^n + c\,(T_{j-1}^n - 2\,T_j^n + T_{j+1}^n)\,, \qquad \text{with} \quad c \equiv \frac{\Delta t\,\kappa}{\Delta x^2}\,. \tag{1.9}$$

This stencil can be derived more formally by writting a Taylor expansion in space

$$\begin{aligned} T^n_{j+\alpha} &= T^n_j + \alpha\,\Delta x \left(\frac{\partial T}{\partial x}\right)^n_j + \alpha^2\,\frac{\Delta x^2}{2}\left(\frac{\partial^2 T}{\partial x^2}\right)^n_j + \alpha^3\,\frac{\Delta x^3}{3!}\left(\frac{\partial^3 T}{\partial x^3}\right)^n_j \\ &\quad +\alpha^4\,\frac{\Delta x^4}{4!}\left(\frac{\partial^4 T}{\partial x^4}\right)^n_j + \alpha^5\,\frac{\Delta x^5}{5!}\left(\frac{\partial^5 T}{\partial x^5}\right)^n_j + \mathcal{O}(\Delta x^6)\,, \end{aligned} \tag{1.10}$$

and a similar expansion in time

$$\begin{aligned} T^{n+\alpha}_j &= T^n_j + \alpha\,\Delta t \left(\frac{\partial T}{\partial t}\right)^n_j + \alpha^2\,\frac{\Delta t^2}{2}\left(\frac{\partial^2 T}{\partial t^2}\right)^n_j + \alpha^3\,\frac{\Delta t^3}{3!}\left(\frac{\partial^3 T}{\partial t^3}\right)^n_j \\ &\quad +\alpha^4\,\frac{\Delta t^4}{4!}\left(\frac{\partial^4 T}{\partial t^4}\right)^n_j + \alpha^5\,\frac{\Delta t^5}{5!}\left(\frac{\partial^5 T}{\partial t^5}\right)^n_j + \mathcal{O}(\Delta t^6)\,, \end{aligned} \tag{1.11}$$

(with $\alpha \in I\!N$). Adding the expressions corresponding to (1.10) with $\alpha = 1$ and $\alpha = -1$, all the odd numbers cancel out and we are left with

$$T_{j-1} + T_{j+1} = 2T_j + \Delta x^2\,\frac{\partial^2 T}{\partial x^2}\bigg|^n_j + \frac{\Delta x^4}{12}\,\frac{\partial^4 T}{\partial x^4}\bigg|^n_j + \mathcal{O}(\Delta x^6)\,, \tag{1.12}$$

thus the approximation

$$\frac{\partial^2 T}{\partial x^2}\bigg|^n_j = \frac{T^n_{j-1} - 2T^n_j + T_{j+1^n}}{\Delta x^2} - \frac{\Delta x^2}{12}\,\frac{\partial^4 T}{\partial x^4}\bigg|^n_j + \mathcal{O}(\Delta x^4)\,. \tag{1.13}$$

We deduce from this short calculation that the stencil we proposed in (1.9) is accurate to second order in space.

The same reasonning using (1.11) with $\alpha = 1$, yields

$$\frac{\partial T}{\partial t}\bigg|^n_j = \frac{T^{n+1}_j - T^n_j}{\Delta t} - \frac{\Delta t}{2}\,\frac{\partial^2 T}{\partial t^2}\bigg|^n_j - \mathcal{O}(\Delta t^2)\,. \tag{1.14}$$

Our stencil is thus only first order accurate in time (i.e. the truncation error term vanishes to 0 as Δt).

We can now identify our expressions with the general formalism we just introduced, then

$$L(T) = \frac{\partial T}{\partial t} - \kappa\frac{\partial^2 T}{\partial x^2} = 0\,, \tag{1.15}$$

and we can then write

$$\begin{aligned} L(T) &= \frac{T(x_j, t_{n+1}) - T(x_j, t_n)}{\Delta t} - \kappa\frac{T(x_{j-1}, t_n) - 2T(x_j, t_n) + T(x_{j+1}, t_n)}{\Delta x^2} \\ &\quad -\frac{\Delta t}{2}\,\frac{\partial^2 T}{\partial t^2}\bigg|^n_j + \kappa\frac{\Delta x^2}{12}\,\frac{\partial^4 T}{\partial x^4}\bigg|^n_j + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^4)\,, \end{aligned} \tag{1.16}$$

and thus

$$R_h(T) = \frac{\Delta t}{2} \left.\frac{\partial^2 T}{\partial t^2}\right|_j^n - \kappa \frac{\Delta x^2}{12} \left.\frac{\partial^4 T}{\partial x^4}\right|_j^n + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^4)\,. \tag{1.17}$$

A convergence analysis on the computer however reveals that the error does not always vanish...

1.3 Von Neumann stability analysis

We ensured that the discrete equation was "close" to the initial problem, the difference being defined as the truncation error. However, during the numerical resolution, we will rely on this approximation of the continuous problem by the discrete stencil a high number of times. Small errors performed at each time step could therefore accumulate and the numerical solution could then deviate from the solution of the original problem.

An efficient approach to investigate the numerical stability of a stencil is to perform the Von Neumann stability analysis. Let us consider an infinite, or periodic, domain, with a regular grid $x_j = j\Delta x$ and a harmonic pertubation at a given time of the form $f = \mathrm{e}^{ikx}$.

The time evolution of this perturbation can be established by considering the amplification factor, which we define as

$$\xi = \frac{f^{n+1}}{f^n}\,. \tag{1.18}$$

This complex factor multiplies the considered perturbation at each time step. By injecting this expressions in (1.9), we get

$$\xi = 1 + c\,(\mathrm{e}^{ik\Delta x} - 2 + \mathrm{e}^{-ik\Delta x})\,, \tag{1.19}$$

with $\mathrm{e}^{ik\Delta x} + \mathrm{e}^{-ik\Delta x} = 2\cos(k\Delta x)\,,$

$$\xi = 1 - 2\,c\,\left[1 - \cos k\Delta x\right]\,, \tag{1.20}$$

and $2\,\sin^2 x = 1 - \cos 2x$, thus we get

$$\xi = 1 - 4\,c\,\sin^2\left(\frac{k\Delta x}{2}\right)\,. \tag{1.21}$$

For the perturbation not to be amplified in time, it is necessary to require that $|\xi| \leq 1$.

In our case $\xi \in I\!R$ and we always have $\xi < 1$. To ensure that $-1 \leq \xi$, one can consider the wake number k which maximises the sinus, one then gets

$$-1 \;\leq\; 1 - 4c\,, \tag{1.22}$$

$$c \;\equiv\; \kappa\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}\,. \tag{1.23}$$

Having Δt and Δx small is therefore not enough. If they do not match this criterion, the scheme, eventhough consistant, will not offer a good approximation to our original continuous problem, as it will amplify perturbations. It can be consistant yet unstable. For it to be stable, relation (1.23) needs to be satisfied.

1.4 The Lax theorem

Can we be sure to have identified all the possible difficulties?
Are the consistancy and the stability the two only relevant properties?

Lax theorem :

For a well-posed linear initial value problem, wellposed in the sens of Hadamard[1] *consistancy* and *stability* of the numerical scheme imply the *convergence* of the solution of the discrete problem to that of the original problem.

[1] A problem is said to be well-posed in the sens of Hadamard if there exists one, and only one, solution which changes smoothly when the parameters are varied.

Chapter 2

Taylor expansion and Finite Differences

2.1 Stencils and orders

Relying on the Taylor expansion approach introduced in the previous chapter [see equation (1.10)], one can straightforwardly derive many stencils (or schemes) to approximate, say, the first derivative

$$\partial_x u|_j = \frac{u_{j+1}-u_{j-1}}{2\Delta x} - \frac{\Delta x^2}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_j + \dots, \tag{2.1}$$

$$\partial_x u|_j = \frac{u_j-u_{j-1}}{\Delta x} + \frac{\Delta x}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_j - \frac{\Delta x^2}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_j + \dots, \tag{2.2}$$

$$\partial_x u|_j = \frac{u_{j+1}-u_j}{\Delta x} - \frac{\Delta x}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_j - \frac{\Delta x^2}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_j + \dots. \tag{2.3}$$

Let us introduce the following notations to identify the three above schemes

$$\delta_0 u\ |_j = \frac{u_{j+1}-u_{j-1}}{2\Delta x}, \qquad \delta_- u\ |_j = \frac{u_j-u_{j-1}}{\Delta x}, \text{and } \delta_+ u\ |_j = \frac{u_{j+1}-u_j}{\Delta x}.$$

The very fact that we are given the choice between three competing schemes to represent one given derivative is in it self a bit alarming. Surely the choice is not obvious and will not be free of consequences...

Having established the above formula we are naturally pushed by the irresistible temptation to involve more neighbouring points. One can for example use three points and write the non-symmetric second order stencils

$$\partial_x u|_j = \frac{-3u_j+4u_{j+1}-u_{j+2}}{2\Delta x} + \frac{\Delta x^2}{3}\left(\frac{\partial^3 u}{\partial x^3}\right) + \dots, \tag{2.4}$$

$$\partial_x u|_j = \frac{3u_j-4u_{j-1}+u_{j-2}}{2\Delta x} + \frac{\Delta x^2}{3}\left(\frac{\partial^3 u}{\partial x^3}\right) + \dots, \tag{2.5}$$

or widen the stencil and write the four points fourth order formula

$$\partial_x u|_j = \frac{-u_{j+2} + 8u_{j+1} - 8u_{j-1} + u_{j-2}}{12\Delta x} + \frac{\Delta x^4}{3}\left(\frac{\partial^5 u}{\partial x^5}\right) + \dots . \tag{2.6}$$

Following a similar approach, schemes of various order can be written for the second order derivative

$$\partial_{xx} u|_j = \frac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2} - \frac{\Delta x^2}{12}\left(\frac{\partial^4 u}{\partial x^4}\right) + \dots , \tag{2.7}$$

$$\partial_{xx} u|_j = \frac{-u_{j-2} + 16u_{j-1} - 30u_j + 16u_{j+1} - u_{j+2}}{12\Delta x^2} + \frac{\Delta x^4}{90}\left(\frac{\partial^6 u}{\partial x^6}\right) + \dots . \tag{2.8}$$

These are only a few (probably the most classical) of the possible stencils. The methodology introduced here makes it straightforward to construct further expressions, either involving a different choice of points or representing higher derivatives.

2.2 Lagrange polynomials

An alternative approach to derive finite difference formula is to seek for a polynomial expression of degree N through $N+1$ neighbouring points. As an example, assume that we know $u(0) = u_0$, $u(\Delta x) = u_1$, $u(2\Delta x) = u_2$, and seek for a polynomial of the form $u(x) = a + b\,x + c\,x^2$, through these points.

We can easily get $\qquad b = \dfrac{-3u_0 + 4u_1 - u_2}{2\Delta x}\,, \qquad 2\,c = \dfrac{u_0 - 2u_1 + u_2}{\Delta x^2}\,.$

Computing the exact expressions for the derivative of this polynomial at point $x = 0$ yields the second order expression corresponding to (2.4). A first order expression for the second derivative can also easily be derived from $2\,c$. Interestingly, this yields a similar expression of the second derivative as (2.7), but evaluated at point $j-1$. This point to the uniqueness of the parabola passing through three points, in much the same way as (2.1)-(2.3) point to the uniqueness of the line passing through two points.

A general way to proceed is to use the Lagrange polynomial. Assuming, for the simplicity of the example, the points

$$(x_{j-1}, f_{j-1})\,, \qquad (x_j, f_j)\,, \qquad (x_{j+1}, f_{j+1})\,,$$

are known (it is easy to generalise to more points and higher order polynomials), the polynomial takes the form

$$F(x) = \frac{(x - x_j)(x - x_{j+1})}{(x_{j-1} - x_j)(x_{j-1} - x_{j+1})}\, f_{j-1} + \frac{(x - x_{j-1})(x - x_{j+1})}{(x_j - x_{j-1})(x_j - x_{j+1})}\, f_j + \frac{(x - x_{j-1})(x - x_j)}{(x_{j+1} - x_{j-1})(x_{j+1} - x_j)}\, f_{j+1}\,. \tag{2.9}$$

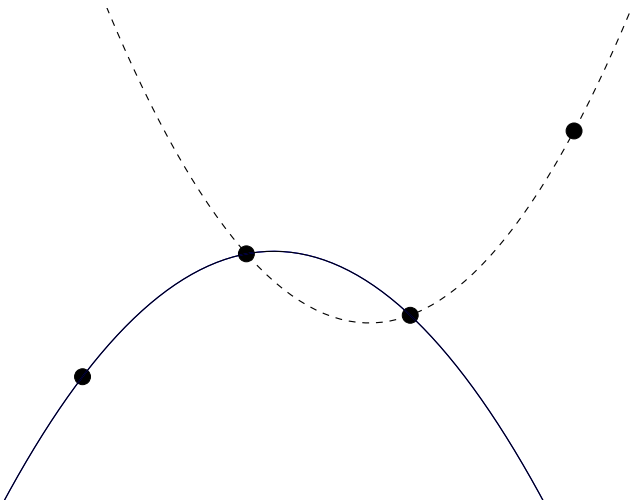


Figure 2.1: Lagrange polynomials offer an easy way to derive finite difference formula they however clearly do not correspond to a reconstruction of the function between the discretisation points. The figure highlights that it does not provide a unique reconstruction between two neighbouring points.

It corresponds to the sum of three parabola, respectively passing through the points $[(x_{j-1}, f_{j-1}), (x_j, 0), (x_{j+1}, 0)]$, $[(x_{j-1}, 0), (x_j, f_j), (x_{j+1}, 0)]$, $[(x_{j-1}, 0), (x_j, 0), (x_{j+1}, f_{j+1})]$.

We can now differentiate the analytical expression (2.9) with respect to x and get

$$\begin{aligned} F'(x) &= \frac{2x - (x_j + x_{j+1})}{(x_{j-1} - x_j)(x_{j-1} - x_{j+1})} f_{j-1} + \frac{2x - (x_{j-1} + x_{j+1})}{(x_j - x_{j-1})(x_j - x_{j+1})} f_j \\ &\quad + \frac{2x - (x_{j-1} + x_j)}{(x_{j+1} - x_{j-1})(x_{j+1} - x_{j-1})} f_{j+1} \,. \end{aligned} \tag{2.10}$$

Evaluating this derivative of the polynomial at the reference point x_j and denoting $\Delta x_j = x_j - x_{j-1}$, we get

$$\begin{aligned} F'_j &= \frac{\Delta x_{j+1}}{\Delta x_j (\Delta x_j + \Delta x_{j+1})} f_{j-1} + \frac{\Delta x_{j+1} - \Delta x_j}{\Delta x_j \Delta x_{j+1}} f_j \\ &\quad + \frac{\Delta x_j}{\Delta x_{j+1} (\Delta x_j + \Delta x_{j+1})} f_{j+1} \,. \end{aligned} \tag{2.11}$$

Note that the above expression degenerates to δ_0 if $\Delta x_{j+1} = \Delta x_j$. This approach therefore provides a convenient way to compute Finite Difference formula on an irregular grid. Such formula allow to stretch the grid near the regions of physical interest or rapid variation of the solution (e.g. boundary layers in fluid dynamics).

Lagrange polynomials provides a systematic approach to derive the finite difference coefficients (which can easily be extended to stretched grids). It is however important to note that this is **not** a reconstruction of the function (as illustrated in figure 2.1).

2.3 Two dimensional problems

A straightforward way to handle two dimensional problems is to use one dimensional discretisation in two orthogonal directions, e.g.

$$\Delta u|_{i,j} = \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)\bigg|_{i,j} = \frac{u_{i-1,j} - 2\,u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2\,u_{i,j} + u_{i,j+1}}{\Delta y^2}\,. \quad (2.12)$$

This is the most natural way, but certainly not the only one.

An alternative scheme one could consider involves derivation accross the diagonals. This scheme is

$$\Delta u|_{i,j} = \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)\bigg|_{i,j} = \frac{(u_{i+1,j+1} - 2\,u_{i,j} + u_{i-1,j-1}) + (u_{i+1,j-1} - 2\,u_{i,j} + u_{i-1,j+1})}{\Delta x^2 + \Delta y^2} \quad (2.13)$$

It however introduces two decoupled grids in the form of a checker board (see `C2.1_2D_heati_vect.py`).

2.4 Dual grid and staggered meshes

The notion of dual grid is essential in computational physics. It is simple, but occures very often and sometimes unexpectedly. It is directly related to the fact that three different expressions (δ_0, δ_+, and δ_-) were obtained for the first derivative when involving only neighbouring points. The best way to understand it is to appreciate that a minimal definition of the first derivative would relate to the slope of the line (1st order polynomial) passing through two points(in the sense of teh Lagrange polinomials previously introduced). It is easy to see that this expression is second order only at the center of the interval defined by these two points. As a result the first derivative is naturally defined between the grid points at which the function is known.

To illustrate this, let us now assume that we now consider a problem in which the thermal conductivity coefficient is varying in space

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\kappa(x)\frac{\partial T}{\partial x}\right)\,. \quad (2.14)$$

It is then tempting to develop this expression as

$$\frac{\partial T}{\partial t} = \kappa(x)\frac{\partial^2 T}{\partial x^2} + \frac{\partial}{\partial x}\kappa(x)\frac{\partial T}{\partial x}\,. \quad (2.15)$$

The first term is familiar. The second term, however involves first derivatives. This implies a selection between δ_0, δ_+, δ_- which is far from obvious. Such analytical development is to be avoided, and it is much more appropriate to discretise (2.14) rather than (2.15).

In order to achieve this, it is useful to note that combinations of δ_+ and δ_- provide

$$\delta_-\delta_+ = \delta_+\delta_- = \frac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2}\,, \quad (2.16)$$

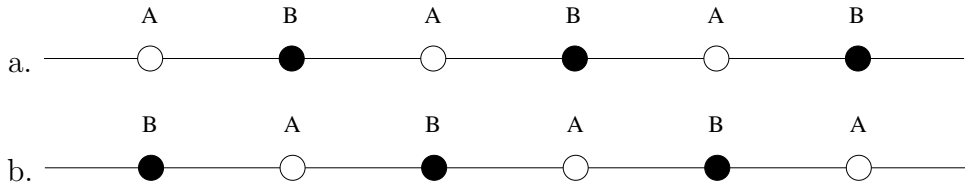


Figure 2.2: Two decoupled grids when using the centered scheme in one dimension of space.

i.e. the second order formula (2.7) for the second derivative that we used so far. It is interesting that these combinations of two first order stencils result in a dense second order approximation. The combination of the second order formula with itself $\delta_0\delta_0$ would instead result in a sparse stencil (by sparse we mean that it "skips" the direct neighbours)

$$\delta_0\delta_0 = \frac{u_{j-2} - 2u_j + u_{j+2}}{(2\,\Delta x)^2}\,. \tag{2.17}$$

This is to be avoided as it introduces decoupled grids.

If one uses the above centered scheme and define each quantity

$$q = -\kappa\,\frac{\partial T}{\partial x}\,,$$
$$\frac{\partial T}{\partial t} = -\frac{\partial}{\partial x} q\,,$$

at all points, one can easily see that the scheme results in two decoupled grids (see figure 2.2). The first grid involves T at odd indices and q at even indices, and the other one involves the reverse. This is not only a waist of computational power (the computation could be performed with the same precision on T at a given point using only one of these grids), it is also a common source of numerical difficulties, in particular if the two grids become weakly coupled (for example via an extra term in the equation).

The natural way to discretise (2.14) in the light of the above remarks is thus to use a staggered grid (i.e. only one of the above mentioned decoupled grids). This approach is illustrated on figure 2.3 in which $q = -\kappa\partial T/\partial x$ is defined on a dual grid, shifted half a grid point with respect to the original grid on which T is defined.

One can then easily introduce the following discretisation

$$q_{j+1/2} = -\,\kappa\frac{\partial T}{\partial x}\bigg|_{j+1/2} \simeq -\kappa_{j+1/2}\,\frac{T_{j+1} - T_j}{\Delta x}\,, \tag{2.18}$$

$$\text{and} \qquad -\,\frac{\partial q}{\partial x} = \frac{\partial}{\partial x}\kappa(x)\frac{\partial T}{\partial x} \simeq \frac{\kappa_{j+1/2}\,(T_{j-1} - T_j)/\Delta x - \kappa_{j-1/2}\,(T_j - T_{j-1})/\Delta x}{\Delta x}\,. \tag{2.19}$$

In this expression κ may be given analytically and directly estimated on the dual grid, or it could be interpolated as $\kappa_{j+1/2} = (\kappa_j + \kappa_{j+1})/2$ if only known at the grid points. This would yield

$$\frac{\partial}{\partial x}\kappa(x)\frac{\partial}{\partial x} \simeq \frac{\kappa_{j+1/2}\,T_{j+1} - (\kappa_{j+1/2} + \kappa_{j-1/2})\,T_j + \kappa_{j-1/2}\,T_{j-1}}{\Delta x^2}\,. \tag{2.20}$$

Figure 2.3: A staggered grid in one dimension of space.

This provides a natural way to discretise the equation and does not introduce any arbitrary choice between δ_0, δ_+, δ_-, as opposed to (2.15). It also prevents difficulties associated with grid decoupling.

It is interesting to note that the situation gets even worse in two dimensions of space as the heat flux becomes a vector

$$\boldsymbol{q} = -\kappa(\boldsymbol{x})\boldsymbol{\nabla} T\,, \qquad \frac{\partial T}{\partial t} = -\boldsymbol{\nabla}\cdot\boldsymbol{q}\,. \tag{2.21}$$

If we use the centered scheme but define all quantities at all points, we would then get four decoupled grids (see figure 2.4).

Similar grid decoupling occurs for other equations which can be reformulated in the form of a system of first order equations, for example the wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}\,, \tag{2.22}$$

which can be rewritten in the form

$$\frac{\partial u}{\partial t} = -c\frac{\partial P}{\partial x}\,, \tag{2.23}$$

$$\frac{\partial P}{\partial t} = -c\frac{\partial u}{\partial x}\,, \tag{2.24}$$

a system which yields the same decoupling in space and also introduces a decoupling in time!

On the one hand, the naive discretisation of the above system is

$$\frac{u^{n+1}-u^{n-1}}{2\Delta t} = -c\frac{P^n_{j+1}-P^n_{j-1}}{2\Delta x}\,, \tag{2.25}$$

$$\frac{P^{n+1}-P^{n-1}}{2\Delta t} = -c\frac{u^n_{j+1}-u^n_{j-1}}{2\Delta x}\,. \tag{2.26}$$

It introduces decoupled grids.

The staggered grid formulation for this problem, on the other hand, is

$$\frac{u^{n+1}-u^n}{\Delta t} = -c\frac{P^{n-1/2}_{j+1+2}-P^{n+1/2}_{j-1/2}}{\Delta x}\,, \tag{2.27}$$

$$\frac{P^{n+1/2}_{j+1/2}-P^{n-1/2}_{j+1/2}}{\Delta t} = -c\frac{u^n_{j+1}-u^n_j}{\Delta x}\,, \tag{2.28}$$

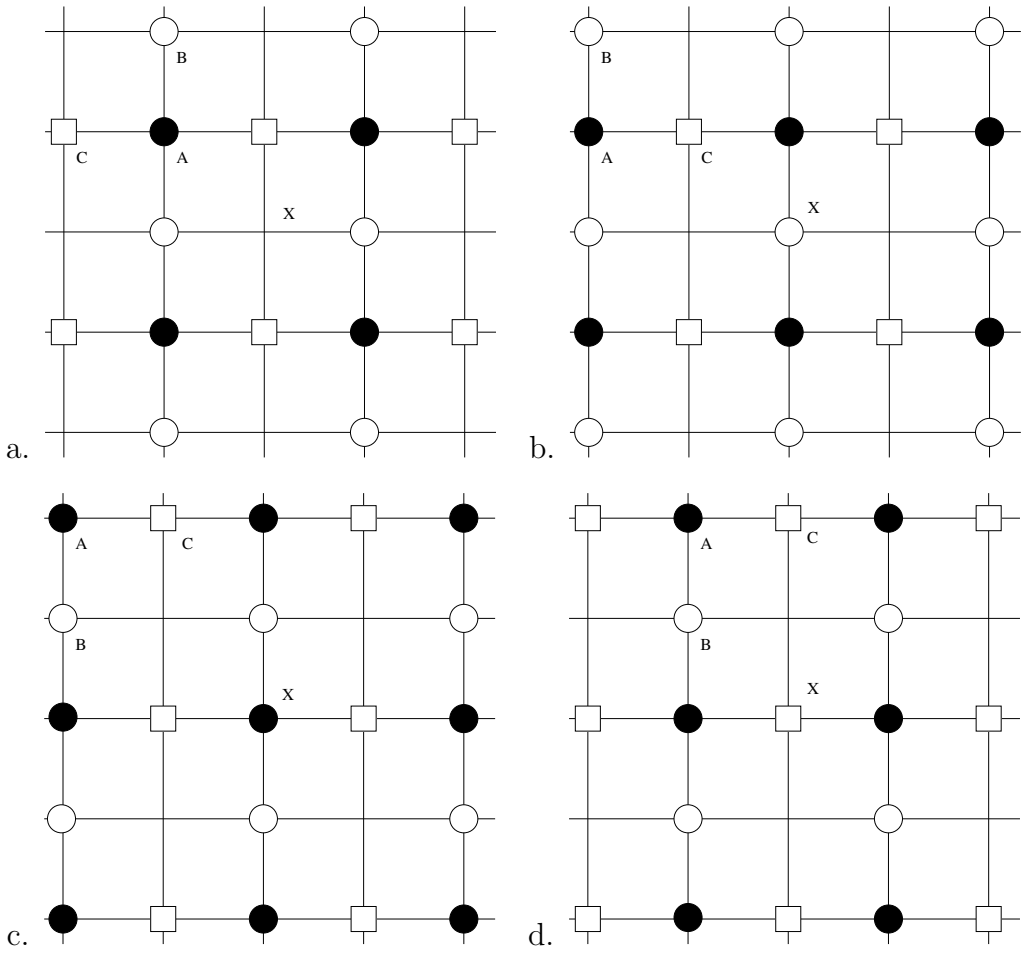


Figure 2.4: In two dimensions of space, four decoupled grid arise when using a centered scheme to discretise (2.21). The four grids are represented here using different symbols for T, $q_x = \boldsymbol{q} \cdot \boldsymbol{e}_x$ and $q_y = \boldsymbol{q} \cdot \boldsymbol{e}_y$. The symbol X is used to identify the same point on all four grids.

One can eliminate P from the above expression and recover the classical scheme

$$\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\Delta t^2} = c^2 \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} \,. \tag{2.29}$$

If the non staggered grid (2.25)-(2.26) had been used, however, the same elimination would have provided the sparse stencil

$$\frac{u_j^{n+2} - 2u_j^n + u_j^{n-2}}{4\Delta t^2} = c^2 \frac{u_{j+2} - 2u_j + u_{j-2}}{4\Delta x^4} \,, \tag{2.30}$$

which would introduce grid decoupling.

2.5 Boundary Conditions

Different types of boundary conditions can be considered depending on the physical problem. In some cases the problem can be considered in a periodic domain, which circumvents the need to implement boundary conditions (as in most previous examples), but this is not always possible! Let us envisage here the most classical types of boundary conditions.

Tosimplify the writting, we will assume a domain $x \in [0, 1]$ and try to specify the boundary condition at $x = 0$ ($j = 0$). The first point inside the domain then corresponds to $j = 1$.

2.5.1 The Dirichlet boundary contions

The value of u is then assumed to be fixed at the boundary (say $u = g$). This is straightforwardly implemented. u_0 is then known for all times . The first point that needs to be computed is u_1.

For example for the heat equation we considered so far, the equation for the evolution of u_1 would be

$$\frac{u_1^{n+1} - u_1^n}{\Delta t} = \kappa \frac{u_2^n - 2u_1^n + g}{\Delta x^2} \,. \tag{2.31}$$

2.5.2 The Neumann and Robin boundary conditions

An alternative form of boundary conditions often met is the *Neumann* boundary condition

$$\left.\frac{\partial u}{\partial x}\right|_0 = 0 \,, \tag{2.32}$$

which is much less straightforward to implement. A reasonnable approach is to implement it by expressing this derivative using a finite difference scheme

$$\frac{u(1) - u(0)}{\Delta x} = 0 \,. \tag{2.33}$$

However, if the boundary is located at 0 this use of the δ_+ formula will yield a first order aproximation, because a first order stencil was used at this stage (even if the rest of the domain is discretised using a second order formula).

One could alternatively consider

$$\frac{-3u_0 + 4u_1 - u_2}{2\Delta x} = 0 \quad \text{which yields} \quad u_0 = \frac{4}{3}u_1 - \frac{1}{3}u_2\,. \tag{2.34}$$

This does provide a second order expression, but at the cost of changing the width of the stencil at the boundary (the corresponding linear system would be more difficult to resolve).

An alternative approach is to use "ghost points", i.e. non physical computational points located on the other side of the boundary. Introducing a ghost point $u(-1)$ and using the second order formula

$$\left.\frac{\partial u}{\partial x}\right|_0 = \frac{u(1) - u(-1)}{2\Delta x} + \mathcal{O}(\Delta x^2) = 0\,, \tag{2.35}$$

and so for a homogeneous boundary condition, $u(-1) = u(1)$.

For the heat equation, this results in

$$\begin{aligned} u^{n+1}(0) &= u^n(0) + \frac{\kappa\Delta t}{\Delta x^2}\left[u^n(-1) - 2\,u^n(0) + u^n(1)\right] &(2.36)\\ &= u^n(0) + \frac{\kappa\Delta t}{\Delta x^2}\left[-2u^{\,n}(0) + 2\,u^n(1)\right]. &(2.37)\end{aligned}$$

Finally, one can use the dual grid introduced in the previous section and have the boundary on the dual grid. If the boundary is implemented at point $j = 1/2$, one gets

$$\left.\frac{\partial u}{\partial x}\right|_{1/2} = \frac{u(1) - u(0)}{\Delta x} + \mathcal{O}(\Delta x^2) = 0\,, \tag{2.38}$$

and so $u(0) = u(1)$.

This expression appears similar to (2.33), but should not be confused as Δx has been modified when shifting the boundary position. This restores second order accuracy.

In the case of *Robin* or *Fourier* boundary condition, of the form $\partial u/\partial x = \alpha u$, the procedure examplified above should be used, but rather in the form of (2.35) than (2.38), as both $\partial u/\partial x$ and u need to be known on the boundary.

2.6 Matrix formulation of finite differences

The scheme we considered for the heat equation

$$u_j^{n+1} - u_j^n = \frac{\kappa\Delta t}{\Delta x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n)\,, \tag{2.39}$$

can be written as

$$(u)^{n+1} = [A]\ (u)^n\ , \tag{2.40}$$

in which $[A]$ is a tri-diagonal matrix.

If we use an implicit scheme instead

$$u_j^{n+1} - u_j^n = \frac{\kappa \Delta t}{\Delta x^2}(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1})\,, \tag{2.41}$$

this matrix formulation will correspond to a linear system to be resolved.

The same sort of systems occur when solving for a potential problem

$$\Delta u = -f\,, \tag{2.42}$$

say for example the gravity potential $\Delta\phi = 4\pi G\mu$.

The matrix formulation is natural in one dimension of space. In two or three dimensions of space it becomes much more complicated. The construction of the resulting matrices is examplified in `C2.3_matrix.py` and uses the Kronecker product. It takes advantage of

$$\Delta_{2D} = Id \otimes \Delta_{1D} + \Delta_{1D} \otimes Id\,. \tag{2.43}$$

An essential point is that the matrices associated with finite differences stencils are always sparse (the matrix are composed essentially of zeros with a few non vanishing diagonals).

A direct consequence of this fact is that solving the linear system associated with the matrix will require $\mathcal{O}(N)$ operations in general (for a matrix of size $N \times N$). The inverse of the matrix however will be a full matrix of size $N \times N$. As a result the product of a vector with the inverse matrix would require $\mathcal{O}(N^2)$ operations. For this reason it is essential not to invert the matrix, but instead to solve for the linear system, even if this needs to be done at each timestep.

2.7 Compact Finite Differences

It is from formula in section 2.1 that in order to achieve higher order finite formula for a given derivative, one needs to involve more neighbouring points (i.e. a wider stencil). The extra degrees of freedom allow to cancel more points in the Taylor expansion and thus to derive higher order expressions (this is clearly examplified by comparing expressions (2.7) and (2.8)). This also corresponds, in the formalism of section 2.2, to the use of higher degree polynomials.

While the formula are formaly of higher order, the approximation is not necessarily better. One is using a wider (non-local) stencil in order to get a finer estimate of a local quantity (the derivative at a point). If the function is not "smooth" at the scale of the grid this is not a sensible strategy.

An alternative strategy is to use a compact stencil ("compact" in the sens they only involve nearest neighbours) but introduce extra degrees of freedom by combining expressions of the unknown and not only the known quantities at these points. These expressions are also referred to as "Paddé" finite differences.

This is best illustrated on an example. Let us assume we want to derive an approximation of $\partial^2 u/\partial x^2 = -f$, using a stencil of the form

$$\alpha u_{j+1} + \beta u_j + \gamma u_{j-1} = a f_{j-1} + b f_j + c f_{j+1} \,. \tag{2.44}$$

Let us first consider the standard finite difference stencil (2.7)

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} - \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4} + \mathcal{O}(\Delta x^4) \,. \tag{2.45}$$

$$\text{From} \quad \frac{\partial^2 u}{\partial x^2} = -f \,, \qquad \text{we get} \quad -\frac{\partial^4 u}{\partial x^4} = \frac{\partial^2 f}{\partial x^2} = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2) \,. \tag{2.46}$$

We can then propose the fourth order compact formula

$$\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} = -f_j - \frac{f_{j+1} - 2f_j + f_{j-1}}{12} + \mathcal{O}(\Delta x^4) \tag{2.47}$$

$$= -\frac{f_{i+1} + 10 f_i + f_{i-1}}{12} + \mathcal{O}(\Delta x^4) \,. \tag{2.48}$$

A similar approach can be used for the first order derivative $\partial u/\partial x = f$. Using the δ_0 standard scheme (see (2.1)), we write

$$\partial_x u|_j = \frac{u_{j+1} - u_{j-1}}{2\Delta x} - \frac{\Delta x^2}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_j + \mathcal{O}(\Delta x^4) \,, \tag{2.49}$$

with

$$\frac{\partial^3 u}{\partial x^3} = \frac{\partial^2 f}{\partial x^2} = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2) \,. \tag{2.50}$$

This provides the following compact formula

$$\frac{u_{j+1} - u_{j-1}}{2\Delta x} = \frac{1}{6} f_{j-1} + \frac{2}{3} f_j + \frac{1}{6} f_{j+1} + \mathcal{O}(\Delta x^4) \,. \tag{2.51}$$

An alternative approach to derive the compact finite difference scheme consists in introducing additional points in the finite difference formula, and then combine the expressions for neighboring points to cancel the contributions of these extra points.

Let us illustrate this approach on the second order derivative. We seek for an expression of the form

$$\Delta x^2 \left[\alpha f''(x - \Delta x) + \beta f''(x) + \gamma f''(x + \Delta x)\right] \simeq a\, f(x - \Delta x) + b\, f(x) + c\, f(x + \Delta x) \,, \tag{2.52}$$

in which we want to determine the coefficients α, β, γ, a, b, c. In order to achieve that, let us first compute the standard finite difference formula adding two fictitious points at which f is evaluated. Because we use more points, we can write fourth order expressions for the second derivative. We get

$$\begin{aligned}
f''_{j-1} &= \frac{35}{12} f_{j-1} - \frac{26}{3} f_j + \frac{19}{2} f_{j+1} - \frac{14}{3} f_{j+2} + \frac{11}{12} f_{j+3} + \mathcal{O}(\Delta x^4) \,, \\
f''_{j} &= \frac{11}{12} f_{j-1} - \frac{5}{3} f_j + \frac{1}{2} f_{j+1} + \frac{1}{3} f_{j+2} - \frac{1}{12} f_{j+3} + \mathcal{O}(\Delta x^4) \,, \\
f''_{j+1} &= \frac{-1}{12} f_{j-1} + \frac{4}{3} f_j - \frac{5}{2} f_{j+1} + \frac{4}{3} f_{j+2} - \frac{1}{12} f_{j+3} + \mathcal{O}(\Delta x^4) \,.
\end{aligned} \tag{2.53}$$

We can now seek for the linear combination of the above three relation, which will suppress the contribution from the two fictitious points. The linear combination of these fourth order expressions will remain accurate to the fourth order. We easily derive from this $\alpha = \gamma = 1$ and $\beta = 10$. The resulting compact scheme, being a linear combination of fourth order schemes is fourth order as well, it is

$$\Delta x^2 \left[f''_{j-1} + 10 f''_j + f''_{j+1} \right] \simeq 12 f_{j-1} - 24 f_j + 12 f_{j+1} \,, \tag{2.54}$$

we thus recover this way expression (2.48).

2.8 Sources of physical errors

Let us now extend our heat equation to the case of a moving fluid

$$\frac{dT}{dt} = \frac{\partial T}{\partial t} + \boldsymbol{u}.\boldsymbol{\nabla} T = \kappa \Delta T \,, \tag{2.55}$$

or in one dimension of space

$$\frac{\partial T}{\partial t} + c \frac{\partial T}{\partial x} = \kappa \frac{\partial^2 T}{\partial x^2} \,. \tag{2.56}$$

If we discretise this equation using a second order scheme, we obtain

$$\frac{T^{n+1} - T^n}{\Delta t} = \kappa \frac{T^n_{j+1} - 2T^n_j + T^n_{j-1}}{\Delta x^2} - c \left(\frac{T^n_{j+1} - T^n_{j-1}}{2\Delta x} \right) \,. \tag{2.57}$$

Let us introduce $\alpha = \kappa \Delta t / \Delta x^2$ and $\beta = c \Delta t / \Delta x$

$$T^{n+1}_j = T^n_j - \beta (T^n_{j+1} - T^n_{j-1}) + \alpha (T^n_{j-1} - 2T^n_j + T^n_{j+1}) \,. \tag{2.58}$$

We will now investigate the stability of this scheme in the purely advective case $\alpha = 0$, this leaves us with

$$T^{n+1}_j = T^n_j - \beta (T^n_{j+1} - T^n_{j-1}) \,. \tag{2.59}$$

$$\xi = 1 - \beta \left(\frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2} \right) = 1 - \beta i \sin(k\Delta x) \,. \tag{2.60}$$

It follows that $|\xi| \geqslant 1$, and the scheme is thus unconditionnaly unstable. If α does not vanish, the diffusive term can stabilise the whole scheme.

The instability can be understood in physical terms by considering the modified equation

$$\frac{u^{n+1} - u^n}{\Delta t} - \frac{\Delta t}{2} \left(\frac{\partial^2 u}{\partial t^2} \right) = -c \left(\frac{u_{j+1} - u_{j-1}}{2\Delta x} - \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3} \right) \,. \tag{2.61}$$

The truncation error is then

$$R_h = \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} + c \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3} \,, \tag{2.62}$$

using the same approximation as with Compact Finite Differences, we write

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \,. \tag{2.63}$$

Then at leading order, we have

$$\frac{\partial u}{\partial t} = c\frac{\partial u}{\partial x} - c^2 \frac{\Delta t}{2} \frac{\partial^2 u}{\partial x^2} \,. \tag{2.64}$$

The truncation therefore introduces a reverse diffusion, which yields the instability (retrograd heat equation).

In order to achieve stability using the centered scheme δ_0, a simple approach is to compensate explicitely for this term. Two famous schemes were introduced in order to achieve this.

The first one is the *Lax-Friedrich* scheme, the overall idea is to replace the first term in the RHS of (2.59) with an average of the two neighbouring points. This introduces an explicit diffusion in the scheme

$$T_j^{n+1} = \tfrac{1}{2}\left(T_{j+1}^n + T_{j+1}^n\right) - \beta(T_{j+1}^n - T_{j-1}^n) \,. \tag{2.65}$$

This scheme is stable provided $|c| \leqslant \Delta x/\Delta t$ but diffusive.

An alternative idea is to try a compensate as closely as possible the destabilizing truncation error by using a finite difference approximation of the diffusing needed to balance it exactly. This yields the Lax-Wendroff scheme

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - \frac{c^2 \Delta t}{2} \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2} = 0 \,. \tag{2.66}$$

This scheme is stable provided $|c| \leqslant \Delta x/\Delta t$ but dispersive.

Let us now consider the discretisation of the advective term using the δ_- stencil

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -c\frac{u_j^n - u_{j-1}^n}{\Delta x} \,. \tag{2.67}$$

This scheme is first order in space and in time. We can investigate its stability

$$u_j^{n+1} = u_j^n - \beta(u_j^n - u_{j-1}^n) \,, \tag{2.68}$$

$$\xi = 1 - \beta(1 - e^{-ik\Delta x}) \,, \tag{2.69}$$

and the scheme is thus stable if

$$0 < \beta < 1 \,. \tag{2.70}$$

This condition requires $c > 0$ and $\Delta t < \Delta x/c$. The symmetric scheme δ_+, yields a similar condition with $c < 0$ and $\Delta t < \Delta x/|c|$.

A stable discretisation of the advective term therefore requires a discretisation involving the upwind (opposite to the direction of the velocity) differenciation. This is known as the upwind scheme.

2.9 Modified wave number

The concept of modified wave number is very useful in numerics as it allows to quantify the numerical distortion introduced on the physical problem. We will use Fourier analysis to probe the transfer behavior of a wave-like input. This point of view is borrowed from digital signal processing; we will thus treat the differentiation stencil as a finite-length filter that produces an approximation to the derivative of the signal via a convolution with the original signal. The comparison of the finite approximation to exact differentiation as a function of the input wavenumber will reveal the transfer function of a finite-difference formula.

This concept is directly related to the numerical diffusion encountered for the upwind scheme, or the numerical dispersion encountered with Lax-Wendroff. It is the source of numerical diffusion, numerical dispersion and numerical anisotropy.

Let us consider a function of the form $f = e^{i(kx)}$, computing its discrete derivative via the centered scheme δ_0 yields

$$\frac{f_{j+1} - f_{j-1}}{2\Delta x} = i\, k'\, f_j \,, \tag{2.71}$$

where k' is a modified wave number.

One easily shows that

$$\frac{k'}{k} = \frac{\sin(k\Delta x)}{k\Delta x} \,. \tag{2.72}$$

Clearly $k'/k \to 1$ when $k\Delta x \to 0$, so if the Fourier mode is well sampled it is not too distorted. However, when the wavelength become comparable with the grid size, the effective wave number k' as seen by the stencil, significantly differs from the real wave number k (see figure 2.5). This introduces the notion of "modified wavenumber".

High-wavenumber modes will therefore be poorly represented on the grid. Only modes with a large number of points per wavelength will be accurately represented. If one solves a wave equation using finite differences, the modified wave number can easily turn a non-dispersive equation to a dispersive one. This effect, known as numercial dispersion, is the exact analogue of the numerical diffusion we introduced before. It corresponds to the effect encountered with the Lax-Wendroff scheme. One can understand it from two angles: either that of the modified equation, with a dispersive term, or equivalently that of the modified wavenumber.

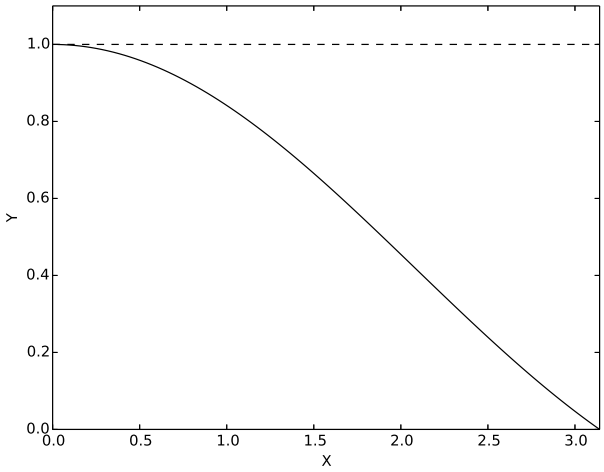


Figure 2.5: Modified wave number for the centered scheme.

Chapter 3

Averaged quantities and Finite Volumes

A few useful Green's formulae

$$\int_{\mathcal{V}} ((\boldsymbol{\nabla}\Phi)\cdot\mathbf{V} + \Phi(\boldsymbol{\nabla}\cdot\mathbf{V}))\,\mathrm{dV} = \int_{\partial\mathcal{V}} \Phi(\mathbf{V}\cdot\mathbf{n})\,\mathrm{dS} \tag{3.1}$$

$$\int_{\mathcal{V}} (\boldsymbol{\nabla}\Phi_1\cdot\boldsymbol{\nabla}\Phi_2 + \Phi_1\Delta\Phi_2)\,\mathrm{dV} = \int_{\partial\mathcal{V}} \Phi_1\boldsymbol{\nabla}\Phi_2\cdot\mathrm{d}\mathbf{S} \tag{3.2}$$

$$\int_{\mathcal{V}} (\Phi_1\Delta\Phi_2 - \Phi_2\Delta\Phi_1)\,\mathrm{d}V = \int_{\partial\mathcal{V}} (\Phi_1\boldsymbol{\nabla}\Phi_2 - \Phi_2\boldsymbol{\nabla}\Phi_1)\cdot\mathrm{d}\mathbf{S} \tag{3.3}$$

$$\int_{\mathcal{V}} (\mathbf{V}_1\cdot(\boldsymbol{\nabla}\wedge\mathbf{V}2) - (\boldsymbol{\nabla}\wedge\mathbf{V}_1)\cdot\mathbf{V}2)\,\mathrm{dV} = \int_{\partial\mathcal{V}} (\mathbf{V}_1\wedge\mathbf{n})\cdot\mathbf{V}_2\,\mathrm{dS} \tag{3.4}$$

3.1 Introduction to Finite Volumes

A very successful method for solving the advection equation arises from a different view point, the Finite-Volume viewpoint. Rather than describing a continuous function by a set of points located on a grid, the function is approximated by small volumes whose average represents the average of the function over these small volumes.

The domain Ω is thus decomposed in $\Omega = \bigcup \Omega_i$ such that $\Omega_i \cap \Omega_j = 0 \forall i \neq j$.

The finite volume formulation introduces

$$u_j^n = \frac{1}{\Delta x}\int_{x_j-1/2}^{x_{j+1/2}} u(x,t)\mathrm{d}x\,, \tag{3.5}$$

and

$$f_{j+1/2}^{n+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u(x_{j+1/2},t)\mathrm{d}t\,, \tag{3.6}$$

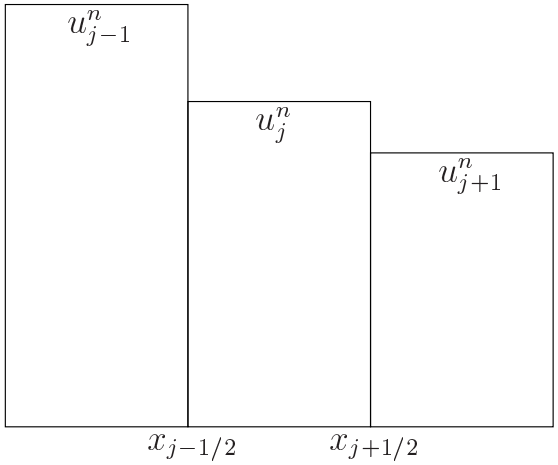


Figure 3.1: The one dimensional Finite Volumes discretisation, with fluxes defined at the cell boundaries.

is the flux function. The time evolution of the cell averaged is then given by

$$u_j^{n+1} - u_j^n = -\frac{\Delta t}{\Delta x}\left(f_{j+1/2}^{n+1/2} - f_{j-1/2}^{n+1/2}\right) . \tag{3.7}$$

The strength of this formulation is that it is *conservative* at the discrete level.

In the case of the heat equation, the flux is readily estimated using centered derivatives.

In the case of pure advection $f(u) = c\,u$, however, the difficulty is then to determine how to estimate $u_{j+1/2}$.

Interpreting the upwind scheme as a finite volume method, the numerical flux is

$$f_{j+1/2}^{up} = \begin{cases} c\,u_j & \text{if} \quad c > 0 \\ c\,u_{j+1} & \text{if} \quad c < 0 \end{cases} \tag{3.8}$$

while the flux function corresponding to the Lax-Wendroff method is

$$f_{j+1/2}^{LW} = \frac{1}{2}c\,(u_j + u_{j+1}) - \frac{\Delta t}{2\Delta x}c^2\,(u_{j+1} - u_j) . \tag{3.9}$$

3.2 Non-linear hyperbolic problems

A generic conservation law can we written in the form

$$\frac{\partial u}{\partial t} = -\boldsymbol{\nabla}\cdot\mathbf{F}(u)\,, \tag{3.10}$$

or in one dimension of space

$$\frac{\partial u}{\partial t} = -\frac{\partial f(u)}{\partial x}\,. \tag{3.11}$$

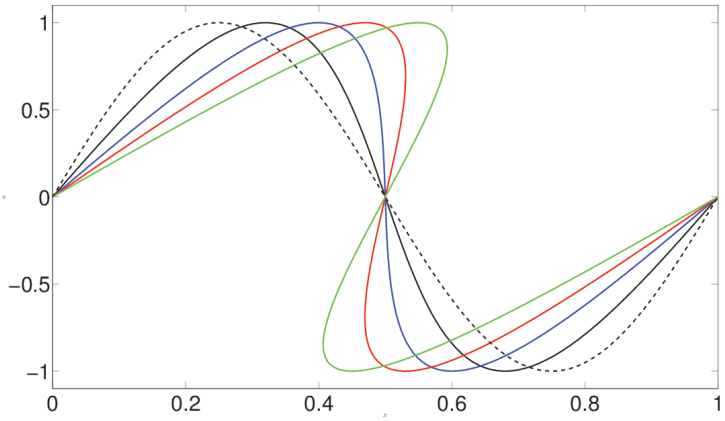


Figure 3.2: Illustration of the formation of a shock with the inviscid Burgers equation.

Where u can be a vector of unknowns. Such is the case when dealing with compressible fluids. In practice, the compressible Euler equations, can be viewed as a system of such non-linear conservation laws.

The approach behind the Godunov method for non-linear hyperbolic conservation laws is to solve exactly for local one dimensional Riemann problem at the faces. The Riemann problem $RP\left[u_L, u_R\right]$ is then defined as the solution of the one-dimensional problem with initial condition

$$u(x,0) = \begin{cases} u_L & \text{if } x < 0 \\ u_R & \text{if } x > 0 \end{cases} \tag{3.12}$$

The idea of the Godunov method is then to solve for

$$u^{\star}_{j+1/2}(x/t) = RP\left[u^n_j, u^n_{j+1}\right] , \tag{3.13}$$

and then

$$f^{n+1/2}_{j+1/2} = f(u^{\star}_{j+1/2}(0)) . \tag{3.14}$$

The solution is a set of piece-wise constant states.

Let us for simplicity consider the scalar one-dimensional inviscid Burgers equation as a model for the complexity arising when the flux function is non-linear. The Burgers equation corresponds to

$$f(u) = \frac{u^2}{2} , \qquad \text{or} \qquad \frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} . \tag{3.15}$$

This equation is a good initial model for compressible hydrodynamics, because it can create shocks from initially smooth solutions. This is illustrated on Figure 3.2.

Physically, the discontinuity is usually regularised at small scale by reintroducing either a small viscosity in the model, the viscous Burgers equation

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) + \varepsilon\frac{\partial^2 u}{\partial x^2} , \tag{3.16}$$

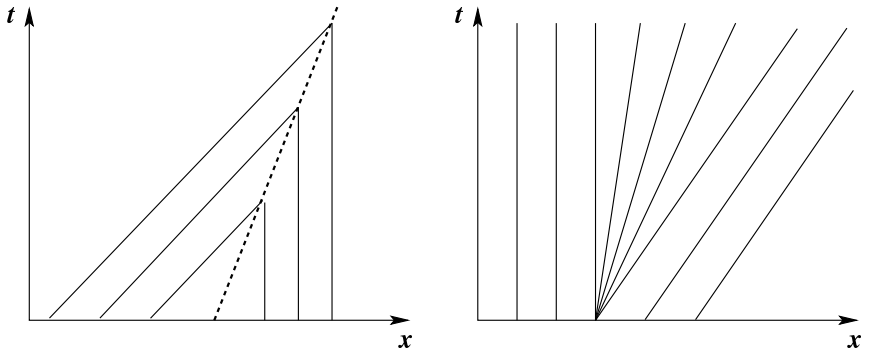


Figure 3.3: Characteristic curves for a shock (left), and rarefaction wave (right). The trajectory of the shock is indicated by the dashed line.

or by introducing dispersion, the KdV equation

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) - \alpha\frac{\partial^3 u}{\partial x^3}\,. \tag{3.17}$$

Numerically, as we saw, similar effects will be introduced via the modified equations.

Let us for the purpose of illustration introduce the solution of the Riemann problem in the case of the Burgers equation.

Two cases must be considered $u_L > u_R$ we are dealing with a shock, whereas if $u_L \leq u_R$ we are dealing with a rarefaction wave. In the case of a shock, we need to introduce the shock speed

$$S = \frac{u_L + u_R}{2}\,, \tag{3.18}$$

and then

$$u(\frac{x}{t}) = \begin{cases} u_L \text{ if } S \geqslant x/t \\ u_R \text{ if } S < x/t \end{cases} \quad \text{or} \quad \begin{cases} u_L \text{if} S > 0 \\ u_R \text{if} S < 0 \end{cases} \tag{3.19}$$

In the case of a rarefaction ($u_L \leqslant u_R$),

$$u(\frac{x}{t}) = \begin{cases} u_L & \text{if} \quad x/t \leqslant u_L \\ x/t & \text{if } u_L < x/t < u_R \\ u_R & \text{if } u_R \leqslant x/t \end{cases} \text{ or } \begin{cases} u_L & \text{if} \quad 0 \leqslant u_L \\ 0 & \text{if } u_L < 0 < u_R \\ u_R & \text{if } u_R \leqslant 0 \end{cases} \tag{3.20}$$

Chapter 4

Variational formulations and Finite Elements

The Finite Element Method is one of the most used methods for numerical solving of differential problems. It does not act directly on the differential equations. The governing equations are first rewritten in a variational (integral) form. Then the integrals are decomposed as sums of integrals on subdomains and the unknown functions are locally approximated by polynomials on those subdomains.

The variational form introduces a rigorous mathematical foundation. This method also has important advantages, in particular the possibility to solve problems on domains with an arbitrary geometry and different type of boundary conditions, and the possibility to use unstructured grids.

This methods is widely used by ingeneers in computational mechanics. It is widely used in industrial codes. It is also the method of choice of applied mathematicians, because it relies on a reformulation of the equations, known as the weak formulation, which clearly specifies the functional spaces. It is seldomely used in research involving computational physics, because it involves, as we will see, quite heavy developments, as a result finite element codes are too often used as black boxes. It can however prove useful for some problems and it is worth investigating.

4.1 The strength of the weak form

Let us consider an example of partial differential equation, in its classical form, which we will here refer to as the strong form

$$\begin{cases} \Delta u = -f & \text{in } \Omega\,, \\ u = 0 & \text{on } \partial\Omega\,. \end{cases} \tag{4.1}$$

In order to define the weak form for this problem, we first need to define a functional space. Let us introduce the H^1 Sobolev space such that

$$H^1(\Omega) = \{w \in L^2(\Omega),\ \frac{\partial w}{\partial x},\ \frac{\partial w}{\partial y} \in L^2(\Omega)\}\,. \tag{4.2}$$

We define the weak formulation of this problem as

$$\text{Find} \quad u \in H_0^1(\Omega) \qquad \text{such that} \quad \int_\Omega \Delta u \, w \, \mathrm{d}\mathbf{x} = -\int_\Omega f \, w \, \mathrm{d}\mathbf{x} \,, \qquad \forall w \in H_0^1(\Omega) \,, \quad (4.3)$$

(where $H_0^1(\Omega)$ corresponds to the space of functions which belong to $H^1(\Omega)$ and in addition vanish on the boundary $\partial\Omega$).

It is obvious that weak form (4.3) can be rewritten as

$$\text{Find} \quad u \in H_0^1(\Omega) \qquad \text{such that} \quad \int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} w \, \mathrm{d}\mathbf{x} = \int_\Omega f w \, \mathrm{d}\mathbf{x} \,, \qquad \forall w \in H_0^1(\Omega) \,. \quad (4.4)$$

The underlying idea, is that instead of asking for the partial differential equation to be satisfied as such, we ask for its projection against any function is a given functional space to be satisfied.

A simple way to think of this formulation is to consider a simple vector equality. One can state that two vectors are equal (this would be the equivalent of the strong form): $\boldsymbol{a} = \boldsymbol{b}$. Alternatively, one can state that the scalar product of both vector against any vector are equal $\boldsymbol{a} \cdot \boldsymbol{w} = \boldsymbol{b} \cdot \boldsymbol{w}$, $\forall \boldsymbol{w}$. This would correspond to the weak form. The next natural step is obviously to introduce a basis for this "any vector". This would result in two dimensions of space with the standard orthonormal basis ($\boldsymbol{e}_1$, $\boldsymbol{e}_2$) in $\boldsymbol{a} \cdot \boldsymbol{e}_1 = \boldsymbol{b} \cdot \boldsymbol{e}_1$, and $\boldsymbol{a} \cdot \boldsymbol{e}_2 = \boldsymbol{b} \cdot \boldsymbol{e}_2$, and thus equality of the components in this basis ($a_1 = b_1$, $a_2 = b_2$). The Finite Element Method indeed follows a very similar path.

In order to illustrate the mathematical strength of the weak form, we can state that if u is a function in $H_0^2(\Omega)$, then u is a solution of the strong field problem (4.1) if and only if it is a solution of the variational problem (4.3).

4.2 The Finite Elements method

In order to reduce the problem to a finite number of degrees of freedom, we need introduce a discrete problem. This can be done by defining a suite of functional spaces $(H_0^1)_h$ of finite dimension with $(H_0^1)_h \underset{h\to 0}{\longrightarrow} H_0^1$.

We then want $\forall u \in H_0^1$

$$\exists u_h \in (H_0^1)_h \text{ such that } \|u - u_h\|_1 \underset{h\to 0}{\longrightarrow} 0 \,, \quad (4.5)$$

with

$$\|u\|_1 = \left(\int_\Omega u(x)^2 \, \mathrm{d}x + \int_\Omega \left| \frac{du(x)}{dx} \right|^2 \mathrm{d}x \right)^{\frac{1}{2}} . \quad (4.6)$$

Let us consider the simple problem

$$\frac{\partial^2 u}{\partial x^2} = -f \quad \text{on a 1D periodic domain } \Omega \,. \quad (4.7)$$

This corresponds to the strong form of the problem. The weak form can be expressed as

$$\int_\Omega \frac{\partial^2 u}{\partial x^2} w \,\mathrm{d}x = -\int_\Omega f w \,\mathrm{d}x \qquad \forall w \in H^1(\Omega)\,, \tag{4.8}$$

which can be rewritten after an integration by part as

$$\int_\Omega \frac{\partial u}{\partial x}\frac{\partial w}{\partial x} \,\mathrm{d}x = \int_\Omega f w \,\mathrm{d}x \qquad \forall w \in H^1(\Omega)\,. \tag{4.9}$$

Let us now discretise Ω in N elements Ω_i and approximate $H^1(\Omega)$ by the space of piecewise linear functions $H^1_h(\Omega)$ (linear on each Ω_i).

Introducing the function ϕ_i which satisfies $\phi_i(x_i) = 1$, $\phi_i(x < x_{i-1}) = \phi(x > x_{i+1}) = 0$, and ϕ linear between x_i and $x_{i\pm1}$, i.e.

$$\phi_i(x) = \begin{cases} 0 & \text{if} \quad x \leqslant x_{i-1}\,, \\ \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if} \ x_{i-1} < x \leqslant x_i\,, \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i} & \text{if} \quad x_i < x \leqslant x_{i+1}\,, \\ 0 & \text{if} \ x_{i+1} < x. \end{cases} \tag{4.10}$$

We can then write

$$u(x) = \sum_{i=1}^{N} u_i \phi_i(x)\,, \quad f(x) = \sum_{i=1}^{N} f_i \phi_i(x)\,, \quad w(x) = \sum_{i=1}^{N} w_i \phi_i(x)\,. \tag{4.11}$$

As the weak form needs to be met for all possible choice of the test function w in the chosen space, it follows that

$$\int_\Omega \sum_i u_i \frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} \,\mathrm{d}x = \int_\Omega \sum_i f_i \phi_i \phi_j \,\mathrm{d}x\,, \qquad \forall j \in [1,\, N]\,. \tag{4.12}$$

Assuming that the domain was uniformly decomposed with elements of the same size h, we get

$$\int_\Omega \frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} \,\mathrm{d}x = \begin{cases} 2/h \ \text{ if } i = j\,, \\ -1/h \text{if } |i-j| = 1\,, \\ 0 \quad \text{otherwise}\,. \end{cases} \tag{4.13}$$

and

$$\int_\Omega \phi_i \,\phi_j \,\mathrm{d}x = \begin{cases} 2h/3 \text{if } i = j\,, \\ h/6 \text{ if } |i-j| = 1\,, \\ 0 \quad \text{otherwise}\,. \end{cases} \tag{4.14}$$

The resulting discrete system thus takes the form

$$\frac{1}{h}\begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & -1 \\ -1 & & & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_N \end{pmatrix}$$

$$= h \begin{pmatrix} 2/3 & 1/6 & & & & 1/6 \\ 1/6 & 2/3 & 1/6 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & 1/6 \\ 1/6 & & & & 1/6 & 2/3 \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_N \end{pmatrix} \tag{4.15}$$

The similarity of the left-hand-side with the finite difference discretisation is striking. The extra matrix on the right-hand-side, known as the mass-matrix has no equivalence in the finite difference formulation, but is a reminiscence of a compact finite difference formula.

It is worth pondering on the distinctions between spectral methods, which we introduced in the preceding chapter and finite elements. The two formulations are indeed remarkably similar and spectral methods could have been introduced via the weak formulation. The first essential point is that basis functions are local in space in the finite element formulation, whereas they are global in spectral method. The second essential distinction is the lack of orthogonality of the basis functions in the finite element framework (which results in the mass matrix). The last important distinction is that in the finite element formalism, the coefficients on the expansion correspond to nodal values (the u_i), such is obviously no the case with spectral methods.

4.2.1 Finite Elements in Practice

The Finite Element method is not very appealing to program in two dimensions of space or more. First of all, a meshing strategy has to be implemented. Then all the integral of shape functions products and/or their derivatives have to be evaluated. This is usually performed by mapping the element (say a triangle) to a regular domain and using quadrature formula for the integrations. While there is no complexity in the method, the implementation is necessarily complex. For this reason, one usually relies on packages or pre-written program when using the Finite Element method.

The FreeFem++ software offers a nice interface for simple problems. Several examples are given on the course webpage

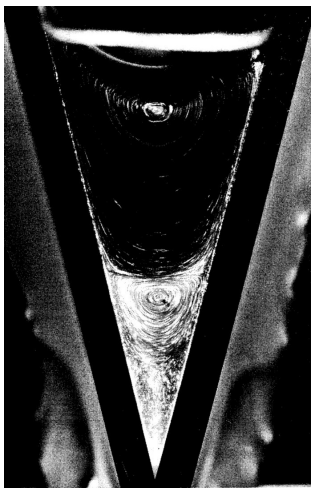


Figure 4.1: The first two Moffatt vortices in the infinite cascade, as visualised by Taneda (1979).

We illustrate here a computation of the bi-harmonic operator for Moffatt vortices (Moffatt, 1964) in Stokes flows. The experimental visualisation (Taneda, 1979) is presented on figure 4.1.

We can construct triangulations of the domain, with various grid size (fig. 4.2) to investigate this problem. The simulation of this problem using finite elements requires higher order elements (suited for the bi-harmonic problem). We use here the so-called Morley elements. The simulations reported on figures 4.3 and 4.4 highlight the need of a large resolution to capture the second Moffatt vortex. One of the strength of Finite Elements, is the possibility to use adaptative mesh refinement (fig. 4.5). It does however not help on the Moffatt vortices problem, because we need resolution where the gradients are weak...

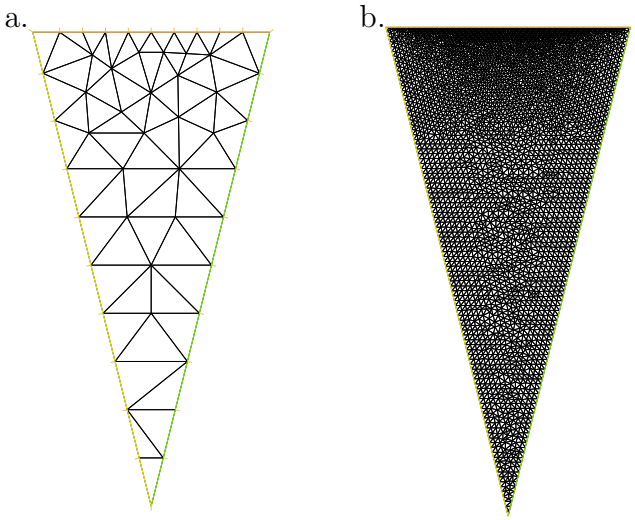


Figure 4.2: Examples of triangular meshes for the investigation of Moffatt vortices. The finer grid (b) has 100 times the number of elements of the coarse grid (a).

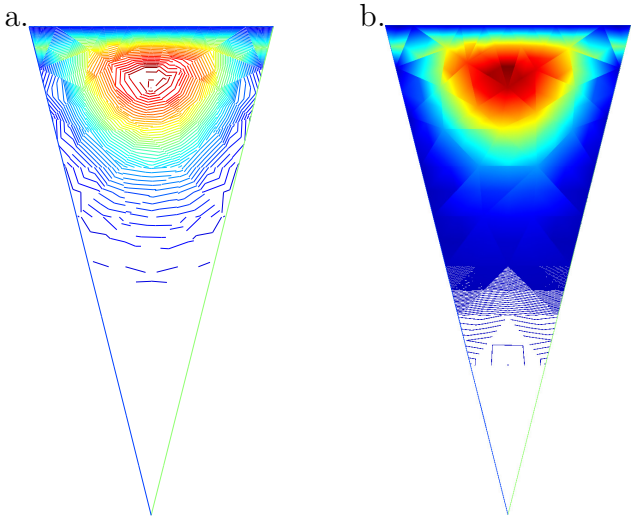


Figure 4.3: Solution obtained with the coarse grid (left with 50 isovaluse, right with 5000 isovalues).

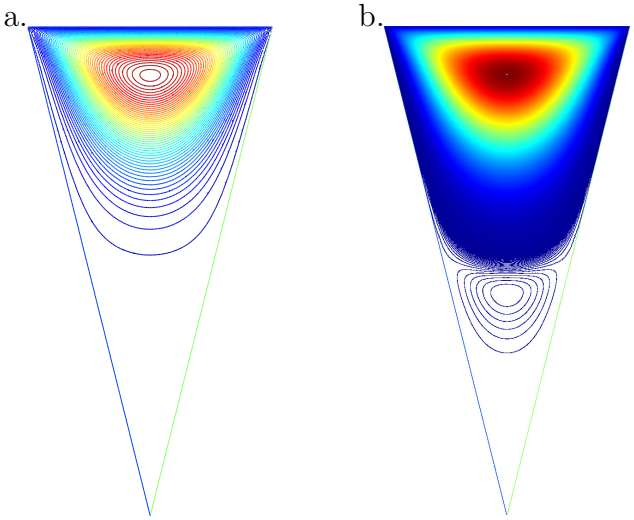


Figure 4.4: Solution obtained with the finer grid (left with 50 isovaluse, right with 5000 isovalues).

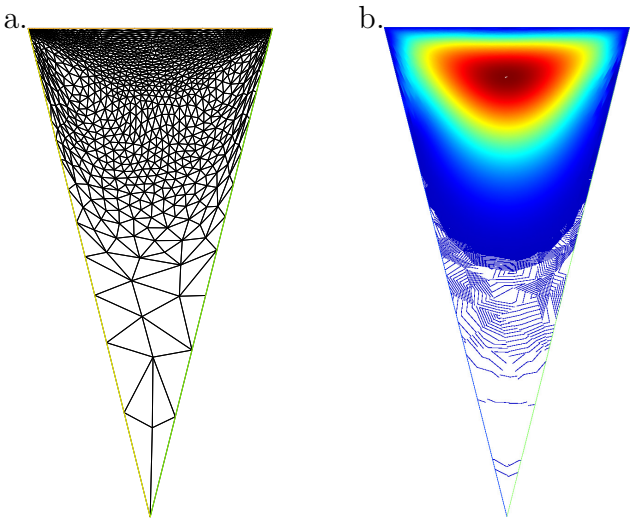


Figure 4.5: Adaptative mesh refinement does not help on the Moffatt vortices problem, because we need resolution where the gradients are weak...

Chapter 5

Spectral methods and Pseudo-spectral methods

5.1 Introduction

In the preceding chapters, the discretisation of the governing equations has always been performed in real space. A finite number of degrees of freedom was obtained after discretising the equations on a finite number of points using finite differences, compact differences or finite volumes. We will introduce in this chapter an alternative method to reduce a partial differential equation to a finite number of degrees of freedom. Instead of discretising in the spatial domain, we will truncate the spectral domain. If, for example, one considers only Fourier modes up to $|k| < k_{\max}$ to represent a periodic function, this will result in a finite number of degrees of freedom to be considered.

5.2 From Fourier transform to discrete Fourier transform

Let us first briefly review some classical results on the continuous Fourier transform. We introduce the basis functions defined by

$$\phi_k(x) = \mathrm{e}^{ikx} \,. \tag{5.1}$$

These functions form an orthogonal set on $[0, 2\pi[$ in the sense that

$$\frac{1}{2\pi}\int_0^{2\pi} \phi_k(x)\phi_l^*(x)\,\mathrm{d}x = \delta_{kl} = \begin{cases} 0 & \text{if } k \neq l\,, \\ 1 & \text{if } k = l\,. \end{cases} \tag{5.2}$$

We denote by $\hat{u}_k$ the Fourier coefficients of u defined as

$$\hat{u}_k = \frac{1}{2\pi}\int_0^{2\pi} u(x)\,\mathrm{e}^{-ikx}\,\mathrm{d}x\,. \tag{5.3}$$

If u is a real function, it follows that $\forall k\,,\ \hat{u}_{-k} = \hat{u}_k^*$.
The inverse Fourier transform of $\hat{u}_k$ is then defined as

$$u(x) = \sum_{k=-\infty}^{+\infty} \hat{u}_k\, \phi_k(x)\,. \tag{5.4}$$

We will now introduce the "Discrete Fourier Transform" of a function u. It assumes that the function u is only evaluated at a finite number of points, $x_j = \frac{2\pi j}{N}$, and it is defined as

$$\hat{u}_k = \frac{1}{N} \sum_{j=0}^{N-1} u(x_j)\, \mathrm{e}^{-ikx_j}\,. \tag{5.5}$$

A remarkable property is that the Fourier functions are still orthogonal at the discrete level

$$\frac{1}{N} \sum_{j=0}^{N-1} \mathrm{e}^{i(k'-k)x_j} = \begin{cases} 1 & \text{if } (k'-k) \in N\mathbb{Z}\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.6}$$

The inverse transform is then

$$u(x_j) = \sum_{k=-N/2}^{N/2-1} \hat{u}_k\, \mathrm{e}^{ikx_j}. \tag{5.7}$$

5.2.1 Shannon theorem and the Nyquist frequency

If the physical domain is dicretised using N points, not all frequencies can, accurately be described. Quite obviously, a signal oscillating with a period shorter that the discretisation step cannot be captured. In fact one can define the maximum wave-number (or frequency if time is being discretised) for a given number of point N. This maximum frequency is often referred to as the Nyquist frequency.

Let us for simplicity first illustrate this on a simple example and consider for instance the two functions F, G defined as

$$F(x) = \cos(k\,x)\,, \qquad G(x) = \cos((N-k)x)\,, \tag{5.8}$$

for some $k < N$. If the spatial discretisation takes the form $x_j = j\Delta x = j\frac{2\pi}{N}$, then

$$F_j = \cos(kj\Delta x)\,, \quad G_j = \cos((N-k)j\Delta x) = \cos(2\pi j - kj\Delta x) = \cos(kj\Delta x) = F_j \tag{5.9}$$

Thus these two Fourier modes cannot be distinguished when evaluated on the discrete grid (sampling) being considered. This phenomenon is illustrated on Figure 5.1.

By using a discretisation with N points, we can only accurately describe (in the sense that the discrete Fourier transform can be reversed without loss) functions whose spectrum lies at most in $[-k_{\max},\, k_{\max}]$, where $N > 2k_{\max}$, i.e. $k_{\max} = N/2$.

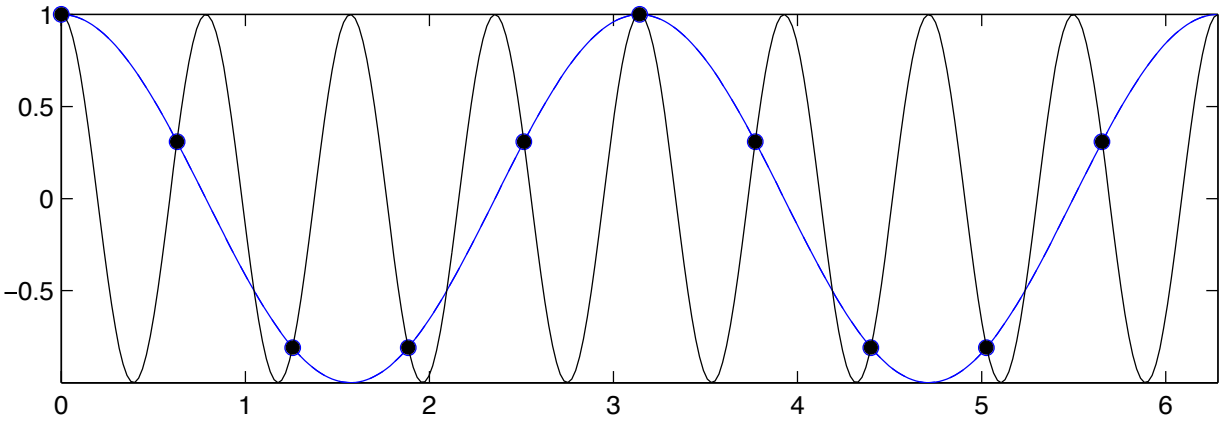


Figure 5.1: Illustration of the Shannon theorem, with 10 sampling points over the interval $[0,\, 2\pi[$. The function $\cos(2\,x)$ and the function $\cos((N-2)\,x) = \cos(8\,x)$ cannot be distinguished.

5.3 Exponential convergence

We may ponder on the order of approximation of spectral methods. How rapidly does the expansion converge? In the case of $\mathcal{C}^\infty$ functions in a $(2\pi$-$)$periodic domain, Fourier coefficients tends to 0 very rapidly with k. If the solution is $\mathcal{C}^\infty$, then for any n positive, we can perform n integrations by parts and write

$$\begin{aligned}\hat{u}_k = \frac{1}{2\pi}\int_0^{2\pi} u(x)\mathrm{e}^{-ikx}\,\mathrm{d}x &= -\frac{1}{2\pi ik}\int_0^{2\pi}\frac{\mathrm{d}u}{\mathrm{d}x}\mathrm{e}^{-ikx}\,\mathrm{d}x + \underbrace{\left[\frac{u(x)\mathrm{e}^{-ikx}}{-2\pi ik}\right]_0^{2\pi}}_{=0} \\ &= \cdots = \frac{1}{2\pi(-ik)^n}\int_0^{2\pi}\frac{\mathrm{d}^n u}{\mathrm{d}x^n}\mathrm{e}^{-ikx}\,\mathrm{d}x \\ &= \mathcal{O}(k^{-n})\,.\end{aligned} \tag{5.10}$$

It follows that the k^{th} Fourier coefficient of a function which is indefinitely differentiable and periodic decays faster than any negative power of k.

In this case the error between a function u and its approximation by its N-th order truncated Fourier series decays faster than algebraically in $1/N$. So assuming that the function is ($\mathcal{C}^\infty$, spectral methods will converge faster than any fixed order finite difference scheme. With additional hypothesis (analytic function), one can show that the convergence is exponential.

Remember that for a finite difference method of order k, the error decays only as N^{-k}. Spectral methods will thus perform much better than any of the finite differences methods seen in Chapter 2, when the above hypothesis ($\mathcal{C}^\infty$ and periodic) are relevant.

5.4 Advection–diffusion in a periodic domain

Differentiation in transform space consists of simply multiplying each Fourier coefficient by the imaginary unit times the corresponding wavenumber. If we consider the advection-diffusion PDE as a first example (with advection at a constant velocity c)

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x} + \kappa\frac{\partial^2 u}{\partial^2 x}\,. \tag{5.11}$$

In spatial Fourier transform, we have

$$\frac{\partial \hat{u}_k}{\partial t} = -cik\hat{u}_k - \kappa k^2\hat{u}_k\,. \tag{5.12}$$

We can solve this last equation by discretisation in time, either in the form

$$\hat{u}_k^{n+1} = \hat{u}_k^n e^{\Delta t(-cik-\kappa k^2)}\,, \tag{5.13}$$

or

$$\frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{\Delta t} = (-cik - \kappa k^2)\hat{u}_k^n\,. \tag{5.14}$$

The first scheme is known as exponential time differencing (ETD) and is exact in this case, the second, the Euler scheme, can be viewed as a Taylor expansion of the exponential in the first scheme. The Fourier method on this equation has the advantage over Finite Differences that we can get rid of numerical diffusion and dispersion problems...

However, problems can occur with more complex equations, for instance with non-linear equations. Let us come back to the Burgers equation, introduced in the previous Chapters

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \varepsilon\frac{\partial^2 u}{\partial^2 x}\,. \tag{5.15}$$

In Fourier space, this equation becomes

$$\frac{\partial \hat{u}_k}{\partial t} = -\hat{u}_k * ik\hat{u}_k - \varepsilon k^2\hat{u}_k\,. \tag{5.16}$$

The product in physical space becomes a convolution product in Fourier space, which couples all the modes and therefore has complexity $O(N^2)$ to compute.

Thankfully Cooley and Tuckey, in 1965, invented a fast algorithm for the discrete Fourier transform. Its complexity is $(O(N\ln(N)))$, and it is called *Fast Fourier Transform* (FFT). The idea behind this algorithm is a "divide and conquer" method: divide the domain (split it in two sub-domains), perform the Fourier Transform in each sub-domain (using a recursive algorithm) and recombine.

This method also permits to reduce the complexity of the convolution, by computing products in real space. This approach is known as a *pseudo-spectral* method. The solution

is transformed at each time-step in the physical space to evaluate products, and then transformed back in Fourier space.
We can also write (5.15) in the form

$$\frac{\partial u}{\partial t} = -\frac{1}{2}\frac{\partial u^2}{\partial x} + \varepsilon \frac{\partial^2 u}{\partial^2 x}\,, \tag{5.17}$$

which leads to

$$\frac{\partial \hat{u}_k}{\partial t} = -\frac{ik}{2}\widehat{(u^2)}_k - \varepsilon k^2 \hat{u}_k\,. \tag{5.18}$$

We can solve numerically this last equation using the two formerly introduced timestepping:

$$\hat{u}_k^{n+1} = \hat{u}_k^n e^{-\Delta t \varepsilon k^2} - \frac{ik}{2}\Delta t \widehat{(u^2)}_k^n\,, \tag{5.19}$$

or

$$\frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{\Delta t} = -\frac{ik}{2}\widehat{(u^2)}_k^n - \varepsilon k^2 \hat{u}_k^n\,. \tag{5.20}$$

$$\tag{5.21}$$

Two important issues remain, which are clearly highlighted on this example.

First the convolution of two functions having their spectra bounded by $k_{\max}$ has a spectrum which externds to $2\,k_{\max}$. In the case of Burgers equation, the non-linear term thus yields non-vanishing coefficients up to $2k_{\max}$ in the Fourier space. We therefore need to evaluate the product in the physical space using twice the number of points needed to accurately represent the original function ($2N$ points) to avoid "aliasing". This procedure is known as "de-aliasing".

Another limit of the spectral method highlighted by this example lies in the formation of discontinuity in the solutions. Discontinuous functions are ill-adapted to spectral methods: functions that have a limited spectrum in Fourier space *are* continuous! (even $\mathcal{C}^\infty$). Fourier series cannot accurately represent a dicontinuous function. A truncated serie will yield oscillating solutions (known as "Gibbs" phenomenon).

Chapter 6

Time evolution and time stepping strategies

6.1 Ordinary Differential Equations (ODE)

As we want to investigate time integration schemes, let us consider the following, possibly nonlinear, ordinary differential equation in time

$$\frac{\mathrm{d}u}{\mathrm{d}t} = f(t,u) \qquad \text{with} \qquad u(0) = u0\,. \tag{6.1}$$

It is then natural to first consider approximations of the time derivative of the form

$$\frac{\mathrm{d}u}{\mathrm{d}t} \simeq \frac{u^{n+1}-u^n}{\Delta t}\,. \tag{6.2}$$

Then depending on the time at which the right-hand-side is evaluated, we get the first ordet explicit Euler formula

$$u^{n+1} = u^n + \Delta t\, f^n\,, \tag{6.3}$$

the first order implicit Euler formula

$$u^{n+1} = u^n + \Delta t\, f^{n+1}\,, \tag{6.4}$$

or the second order Crank-Nicolson scheme

$$u^{n+1} = u^n + \Delta t\, \frac{f^{n+1}+f^n}{2}\,. \tag{6.5}$$

When facing such a choice, it is natural to anticipate that the choice will not be free of consequences on the nature of the numerical integration... Indeed, those first three schemes have very different properties, which we will investigate.

If we consider the simple example of the harmonic oscilator

$$\ddot{X} + \omega^2 X = 0\,. \tag{6.6}$$

It can easily be rewritten in the form of a system of two first order equations

$$\dot{X} = \omega Y\,, \qquad \text{and} \qquad \dot{Y} = -\omega X\,. \tag{6.7a,b}$$

This system can naturally be discretised in time using the explicit Euler formula (6.3)

$$X^{n+1} = X^n + \Delta t\,\omega\, Y^n\,, \qquad \text{and} \qquad Y^{n+1} = Y^n - \Delta t\,\omega\, X^n\,, \tag{6.8a,b}$$

we may alternatively choose to discretise both equation with an implicite Euler scheme (6.4)

$$X^{n+1} = X^n + \Delta t\,\omega\, Y^{n+1}\,, \qquad \text{and} \qquad Y^{n+1} = Y^n - \Delta t\,\omega\, X^{n+1}\,, \tag{6.9a,b}$$

or combining both expressions, in the form of the Crank-Nicholson scheme (6.5)

$$X^{n+1} = X^n + \Delta t\,\omega\, \left(Y^n + Y^{n+1}\right)/2\,, \tag{6.10a}$$

and

$$Y^{n+1} = Y^n - \Delta t\,\omega\, \left(X^n + X^{n+1}\right)/2\,. \tag{6.10b}$$

We will see that these three schemes have very different properties... It can easilly be verified on the continuous system, that the quantity $\mathscr{H} = X^2 + Y^2$ is a conserved quantity ($\dot{\mathscr{H}} = 2\omega XY - 2\omega XY = 0$). In physical terms, for a pendulum say, $\mathscr{H}$ would be the total energy and the terms X^2 and Y^2 would respectively correspond to its potential and the kinetic energy.

The simplest way to apply the three above schemes to this system is to introduce the complex variable $\mathcal{V} = X + iY$, then (6.7a,b) becomes

$$\dot{\mathcal{V}} = -i\omega\mathcal{V}\,. \tag{6.11}$$

We can then write the simple Python code below from which we get figure 6.1. The red curve corresponds to the explicit scheme (6.8a,a), we can see that $\mathscr{H}$ grows in the discrete system (so energy is gained from the discretisation). The blue curve corresponds to the implicit scheme (6.9a,a). $\mathscr{H}$ now decays with time (so energy is lost through the discretisation)... The third scheme (black), however, preserved the "Hamiltonian" $\mathscr{H}$ at the discrete level. Such schemes are known as simplectic integrators.

The Crank-Nicholson is not the only symplectic integrator for this equation. For example we could also have tried to sole one equation explicitely and the other implicitely,

$$X^{n+1} = X^n + \Delta t\,\omega Y^n \qquad \text{and} \qquad Y^{n+1} = Y^n - \Delta t\,\omega X^{n+1}\,. \tag{6.14a,b}$$

This scheme is also symplectic. It is known as the *Verdet* scheme. It is natural to think of it in terms of a staggered mesh in time. Indeed, up to a modification of the initial conditions, it can be rewritten as

$$X^{n+1} = X^n + \Delta t\,\omega Y^{n+1/2} \qquad \text{and} \qquad Y^{n+1/2} = Y^{n-1/2} - \Delta t\,\omega X^n\,. \tag{6.15a,b}$$

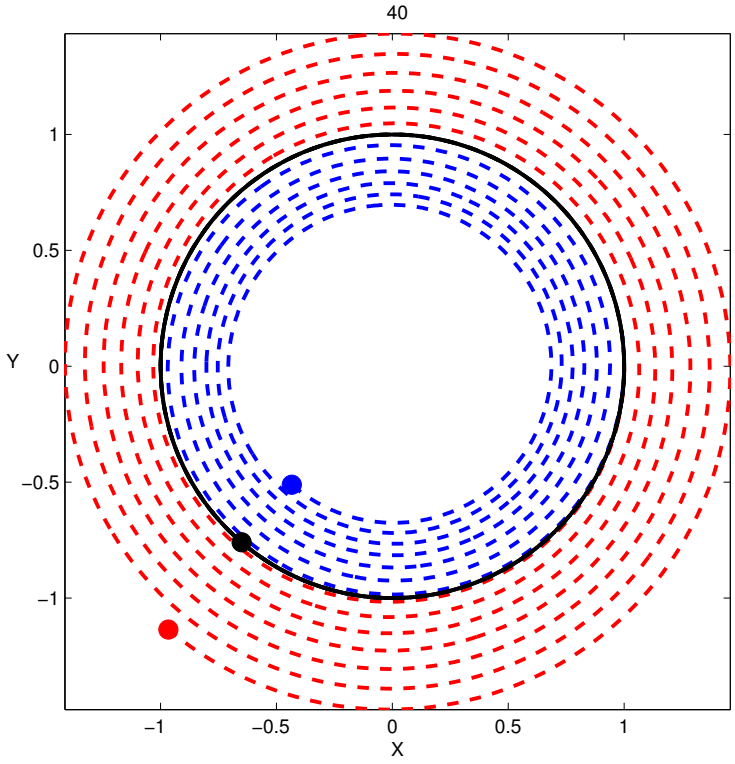


Figure 6.1: Phase protrait of the harmonic oscillator, as numerically integrated using an explicite scheme (red), an implicite scheme (blue) and the Crank-Nicholson scheme (black). Only the third scheme preserves the Hamiltonian structure (symplectic integrator).

If we substitute for Y in the above scheme, we get the natural dicretisation of (6.6)

$$X^{n+1} = 2\,X^n - X^n - \Delta t^2\,\omega^2 X^n\,. \tag{6.16}$$

The different symplectic schemes can be thought f as resulting from different discrete approximation to the Hamiltonian $\mathscr{H}$.

Now that we are motivated to investigate in details the properties of time-stepping schemes, let us first start by generalizing the discretisation strategy to schemes involving wider stencils, as we did for space in Chapter 2.

6.1.1 Linear multistep methods

An s-step method uses at time t^n the known solution values at previous time steps

$$u^{n+1-s}, ..., u^{n-1}, u^n$$

in order to define the next unknown values, u^{n+1}.

Assuming for simplicity a uniform mesh, $t^n = n\Delta t\,,\ n = 0, 1, ...$ where Δt stands for the step size in time, a general linear s-step method for the ODE (6.1) can be written in the form

$$\sum_{j=0}^{s} \alpha_j\, u^{n+1-j} = \Delta t \sum_{j=0}^{s} \beta_j\, f^{n+1-j}\,. \tag{6.17}$$

Here, $f^{n+1-j} = f(t^{n+1-j}, u^{n+1-j})$ and α_j, β_j are coefficients.

Expression (6.17) can obviously be scaled by an arbitrarily factor, we thus set $\alpha_0 = 1$. The method defined by (6.17) is said to be explicit if $\beta_0 = 0$ (i.e. the right hand side f^{n+1} at the new time step does not need to be known in order to advance the unknown u^{n+1}), and implicit otherwise. Note that for an explicit method the unknown u^{n+1} appears only on the left-hand side, so the method can be written as

$$u^{n+1} = \sum_{j=1}^{s} \left(-\alpha_j\, u^{n+1-j} + \Delta t\,\beta_j f_{n+1-j}\right)\,, \tag{6.18}$$

in which the right-hand-side only involves known quantities.

An implicit method, on the other hand, involves the unknown also through $f^{n+1} = f(t^{n+1}, u^{n+1})$.

The most popular families of linear multistep methods are the Adams family and the backward differentiation formula (BDF) family.

In the Adams family, the left-hand-side of (6.1) is always discretised as

$$\frac{u^{n+1} - u^n}{\Delta t}\,,$$

so that the coefficients α_j resume to

$$\alpha_0 = 1\,, \quad \alpha_1 = -1\,, \quad \text{and}\, \alpha_j = 0 \quad \forall j > 1\,.$$

Table 6.1: Coefficients of Adams-Bashforth methods up to order 6.

p	s	j =	1	2	3	4	5	6
1	1	β_j	1					
2	2	$2\beta_j$	3	-1				
3	3	$12\beta_j$	23	-16	5			
4	4	$24\beta_j$	55	-59	37	-9		
5	5	$720\beta_j$	1901	-2774	2616	-1274	251	
6	6	$1440\beta_j$	4277	-7923	9982	-7298	2877	-475

The order of the approximation is then governed by the choice of coefficients β_j.

The s-step explicit Adams method, also called the Adams-Bashforth method, is obtained by interpolating the right-hand-side f through the previous points $t = t^n, t^{n-1}, ..., t^{n+1-s}$, giving an explicit method of accuracy $p = s$. Table 6.1 displays the coefficients of the Adams-Bashforth methods for s up to 6. Using $s = 1$ yields the standard forward Euler method

$$u^{n+1} = u^n + \Delta t\, f^n\,. \tag{6.19}$$

The two-step Adams-Bashforth method is

$$u^{n+1} = u^n + \Delta t\, \left(\frac{3}{2}\, f_n - \frac{1}{2}\, f_{n-1}\right). \tag{6.20}$$

The s-step implicit Adams method, also called the Adams-Moulton method, is obtained by using an interpolation of f through the previous points plus the new one, $t = t^{n+1}, t^n, t^{n-1}, ..., t^{n+1-s}$, giving an implicit method of accuracy $p = s + 1$. Table 6.2 displays the coefficients of the Adams-Moulton methods for s up to 5. Note that there are two one-step methods here: the backward Euler formula ($s = p = 1$),

$$u^{n+1} = u^n + \Delta t\, f^{n+1}\,, \tag{6.21}$$

and the Crank-Nicolson formula ($s = 1, p = 2$),

$$u^{n+1} = u^n + \Delta t\, \frac{(f^n + f^{n+1})}{2}\,. \tag{6.22}$$

The s-step backward differentiation formula (BDF) method is obtained by evaluating f only at the right end of the current step, (t^{n+1}, u^{n+1}), while driving an interpolating polynomial of u through the previous points plus the next one, $t = t^{n+1}, t^n, t^{n-1}, ..., t^{n+1-s}$, and differentiating the resulting interpolate. This gives an implicit method of accuracy order $p = s$. Table 6.3 displays the coefficients of the BDF methods for s up to 6. For $s = 1$ we again obtain the backward Euler method. These methods are particularly useful for stiff problems.

Table 6.2: Coefficients of Adams-Moulton methods up to order 6.

p	s	j =	0	1	2	3	4	5
1	1	β_j	1					
2	1	$2\beta_j$	1	1				
3	2	$12\beta_j$	5	8	-1			
4	3	$24\beta_j$	9	19	-5	1		
5	4	$720\beta_j$	251	646	-264	106	-19	
6	5	$1440\beta_j$	475	1427	-798	482	-173	27

Table 6.3: Coefficients for the s-step backward differentiation formula up to order 6. The method is implicit and of order $p = s$.

p	s	β_0	α_0	α_1	α_2	α_3	α_4	α_5	α_6
1	1		1	1	-1				
2	2	$\frac{2}{3}$	1	$\frac{-4}{3}$	$\frac{1}{3}$				
3	3	$\frac{6}{11}$	1	$\frac{-18}{11}$	$\frac{9}{11}$	$\frac{-2}{11}$			
4	4	$\frac{12}{25}$	1	$\frac{-48}{25}$	$\frac{36}{25}$	$\frac{-16}{25}$	$\frac{3}{25}$		
5	5	$\frac{60}{137}$	1	$\frac{-300}{137}$	$\frac{300}{137}$	$\frac{-200}{137}$	$\frac{75}{137}$	$\frac{-12}{137}$	
6	6	$\frac{60}{147}$	1	$\frac{-360}{147}$	$\frac{450}{147}$	$\frac{-400}{147}$	$\frac{225}{147}$	$\frac{-72}{147}$	$\frac{10}{147}$

6.1.2 Multistage methods

Linear multistep formulas represent one extreme: one function evaluation per time step. Multistage methods represent another extreme: the function $f(u,t)$ is evaluated at any number of intermediate stages in the process of getting from u^n to u^{n+1}, but those stages are never used again. As a result, multistage methods are often more stable than linear multistep formulas, but at the price of more work per time step.

Multistage methods require multiple evaluations of the right-hand-side per time step. The most famous class of multistage methods are Runge-Kutta methods. Runge-Kutta are one-step methods in which f is repeatedly evaluated within one mesh interval to obtain a higher order method.

Runge-Kutta formulas tend to be "non unique" in the sense that if we define the number of stages s and the order of accuracy p, then for most values of s there are infinitely many Runge-Kutta formulas with the maximal order of accuracy p.

For example, we can use forward Euler to take a trial step to the midpoint of the interval,

$$U_1 = u^n\,, \quad U_2 = u^n + \Delta t\, f(t^n, U_1)\,. \tag{6.23}$$

This can be followed by setting

$$u^{n+1} = u^n + \frac{\Delta t}{2}\left(f(t^n, U_1) + f(t^{n+1}, U_2)\right)\,. \tag{6.24}$$

This corresponds to a two-stage Runge-Kutta method of order 2 (or RK2).

In general, an s-stage Runge-Kutta method for the ordinary differential equation (6.1) can be written in the form

$$U_i = u^n + \Delta t \sum_{j=1}^{s} a_{ij}\, f(t^n + c_j \Delta t, U_j)\,, \qquad \text{for} \quad 1 \le i \le s\,, \tag{6.25}$$

$$u^{n+1} = u^n + \Delta t \sum_{j=1}^{s} b_i\, f\left(t^n + c_j \Delta t, U_j\right)\,, \tag{6.26}$$

where the intermediate values U_i are intermediate approximations to the solution at times $t^n + c_i \Delta t$, which may be correct to a lower order of accuracy than the order for the solution u^{n+1} at the end of the time step.

The coefficients of the Runge-Kutta methods are chosen to make error terms cancel in such a way that u^{n+1} is more accurate. Thus, whereas in an s-step multistep method the solution must be smooth over the several subintervals forming $[t^{n+1-s}, t^{n+1}]$ to realize its high order, here the exact solution may be merely continuous at t^n.

The most often used Runge-Kutta formula is the fourth order formula (or RK4)

$$U_1 = \Delta t\, f(t^n, u^n)\,, \tag{6.27}$$

$$U_2 = \Delta t\, f(t^n + \frac{\Delta t}{2}, u^n + \frac{U_1}{2})\,, \tag{6.28}$$

$$U_3 = \Delta t\, f(t^n + \frac{\Delta t}{2}, u^n + \frac{U_2}{2})\,, \tag{6.29}$$

$$U_4 = \Delta t\, f(t^n + \Delta t, u^n + U_3)\,, \tag{6.30}$$

$$u^{n+1} = u^n + \frac{U_1}{6} + \frac{U_2}{3} + \frac{U_3}{3} + \frac{U_4}{6}\,. \tag{6.31}$$

The fourth-order Runge-Kutta method requires four evaluations of the right-hand side per step Δt.

6.1.3 Stability

The concept of stability of linear multistep formula is very central from a theoretical and practical view point. We will assume a bounded continuous problem and investigate two different stability concepts:

- **Zero-Stability:** If $t > 0$ is held constant, do the computed values $u(t)$ remain bounded as $\Delta t \to 0$?
- **Absolute Stability:** If Δt is kept fixed, do the computed values $u(t)$ remain bounded as $t \to \infty$?

The Zero-Stability of a scheme is concerned with the limit of vanishing time step $\Delta t \to 0$. Although this limit constitutes an important test of linear multistep schemes, a stability analysis for finite Δt is indispensable. For finite Δt we need to take the terms β_j into consideration. Nevertheless, we do not want to perform a separate stability analysis of a linear multistep formula for each function f. Instead, let us now consider the test equation

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \lambda\, u \qquad \text{with} \qquad u(0) = u0\,. \tag{6.32}$$

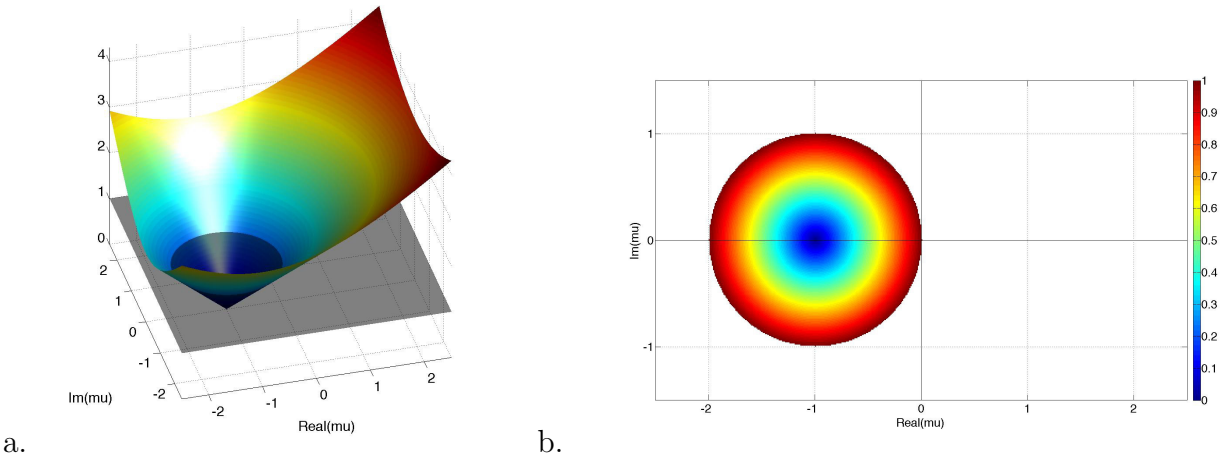
Here λ is a complex number. As we will soon get back to partial differential equations, it can be useful to think of λ as some eigenvalue of a linear operator.

The solution of this test equation, $u(t) = \mathrm{e}^{\lambda t}\, u(0)$ is obviously decaying if $Re(\lambda) \leq 0$, thus

$$\mid u(t) \mid \leq \mid u(0) \mid, \quad \forall t > 0 \quad \leftrightarrow \quad Re(\lambda) \leq 0\,. \tag{6.33}$$

Correspondingly, for a numerical method we define the region of absolute stability as that region of the complex z-plane containing the origin where

$$\mid u^{n+1} \mid \leq \mid u^n \mid, \quad n = 0, 1, 2, ... \tag{6.34}$$



a. b.

Figure 6.2: Domain of absolute stability for the explicit Euler time stepping scheme.

It is then convenient to introduce $\mu = \lambda \Delta t$, and study the above property for varying μ in the complex plane. Note that 0-stability for the test equation corresponds to having the origin $\mu = 0$ belonging to the absolute stability region.

For simplicity, let us start by introducing $\mathcal{Z}$ such that

$$u^{n+1} = \mathcal{Z} u^n \,. \tag{6.35}$$

For the explicit Euler scheme (6.3), we get

$$\mathcal{Z} = 1 + \mu \,. \tag{6.36}$$

The region of the complex plane in which $\mid \mathcal{Z} \mid \leq 1$ defines the region of absolute stability (see Figure 6.2).

Similarly the implicit Euler scheme (6.4), we get

$$\mathcal{Z} = \frac{1}{1-\mu} \tag{6.37}$$

(see Figure 6.3). Clearly in both cases the discretisation in time modifies the stability diagram compared to the continuous case described by (6.33). Then for the Crank-Nicholson formula (6.5), we get

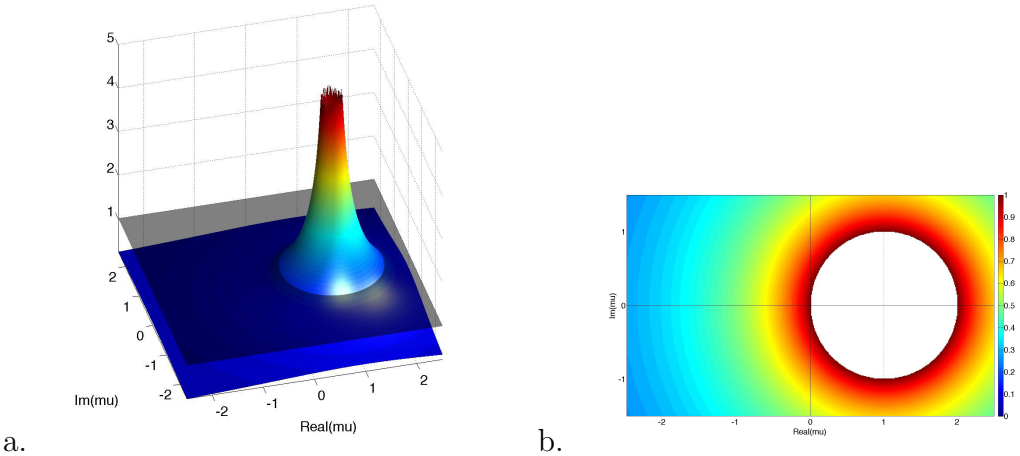
$$\mathcal{Z} = \frac{1+\mu/2}{1-\mu/2} \tag{6.38}$$

(see the corresponding absolute stability region in Figure 6.4).

The later two schemes (Implicit Euler and Crank-Nicholson) contain the entire left half-plane in their absolute stability region. Such methods are called "A-stable".

Formally, if a linear multistep formula is applied to this model equation, it reduces to the recurrence relation

$$\sum_{j=0}^{s} \alpha_j \, u^{n+1-j} - \lambda \Delta t \sum_{j=0}^{s} \beta_j \, u^{n+1-j} = 0 \,. \tag{6.39}$$



a. b.

Figure 6.3: Domain of absolute stability for the implicit Euler time stepping scheme.

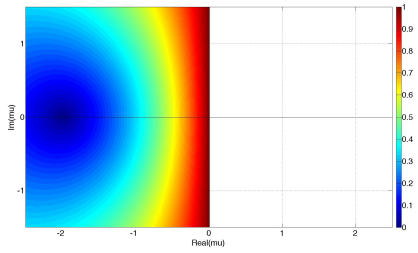


Figure 6.4: Domain of absolute stability for the Crank-Nicholson time stepping scheme.

Figure 6.5 displays the stability regions for a few Adams-Bashforth schemes. It is worth noting that the stability regions of the explicit Adams-Bashforth methods shrink rapidly as s increases. High-order methods of this sort typically yield very restricting stability criteria.

Figure 6.6 displays similar regions for the implicit Adams-Moulton formula. The first two Adams-Moulton methods are not represented because they respectively correspond to the implicit Euler scheme and the Crank-Nicholson scheme presented above.

The absolute stability regions of the first few backward differentiation formula are represented on Figure 6.7. Although the BDF schemes are not A-stable, their region of absolute stability includes the whole of the negative real axis. BDF methods of order larger than 6 can be shown to be zero-unstable.

The stability regions for some explicit Runge-Kutta methods are depicted in Figure 6.8. Unlike for the Adams families, the absolute stability regions here grow with s. We can note on this figure, that whereas the regions for the Runge-Kutta methods of order 2 has no intersection with the imaginary axis, the classical four-stage Runge-Kutta method of order 4 does. This property regarding the intersection of its absolute stability region with the imaginary axis is a major reason for the popularity of this method in time discretisations of some hyperbolic partial differential equations.

6.2 Partial Differential Equations (PDE)

In this section, we will revisit our study of time-dependent partial differential equations (PDEs), whose solutions vary both in space and time. The first example treated in section 1 of this chapter has exhibited what we can now identify as conditional stability.

We can interpret this instability in the light of our study of ordinary differential equations.

For the sake of simplicity, let us now consider an advection-diffusion problem

$$\frac{\partial u}{\partial t} = -c\,\frac{\partial u}{\partial x} + \nu\,\frac{\partial^2 u}{\partial x^2} \qquad \text{with} \quad u(t=0,x) = u_0(x)\,, \tag{6.40}$$

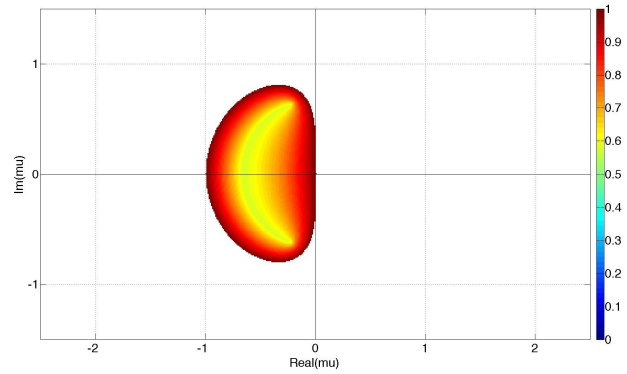
in a 2π periodic domain in x. We will now discretise this equation in space only using N points and Finite Differences as

$$\frac{\partial u}{\partial t} = -c\,\frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} + \nu\,\frac{u(x-\Delta x) - 2u(x) + u(x+\Delta x)}{\Delta x^2}\,. \tag{6.41}$$

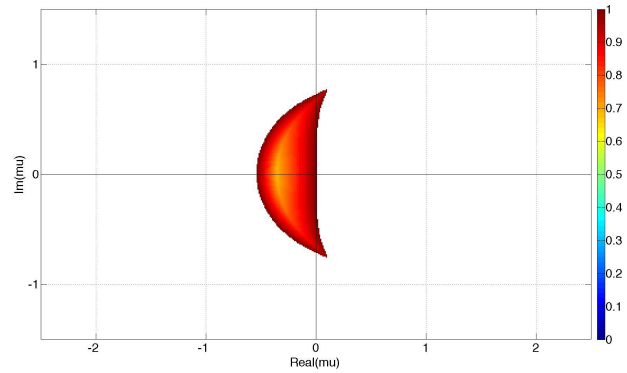
The case $c = 0$ corresponds to the example treated in the first section of this chapter.

The discretisation in space is associated, as we saw in Chapter 2, to a finite number of degrees of freedom and a modified wave number.

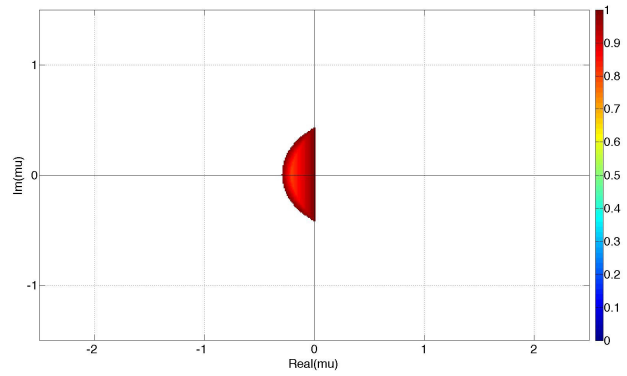
In fact whereas the spectrum of (6.40) is represented by a curve in the complex plane, the one of the discrete equation contains only a finite number of localised eigenvalues. These correspond to the eigenvalue of the matrix A representing the discrete operator on the right-hand-side of (6.41). This matrix is a Toeplitz matrix, i.e. it has constant



a.

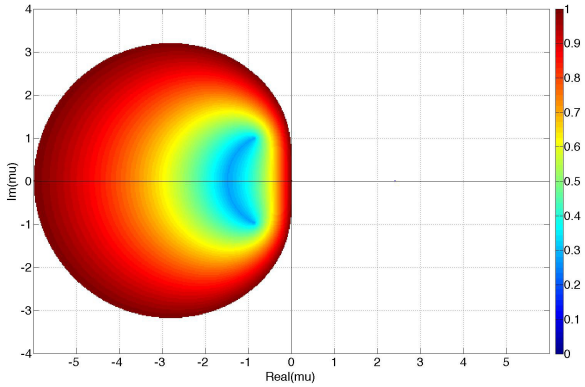


b.

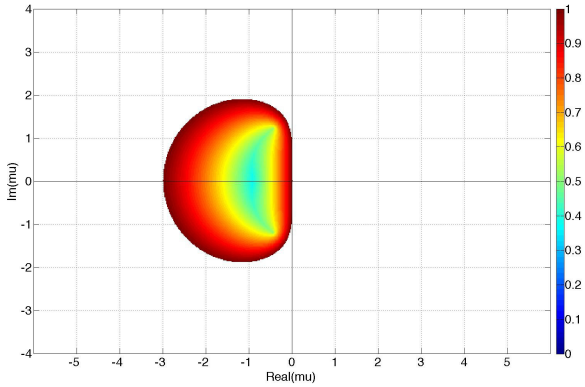


c.

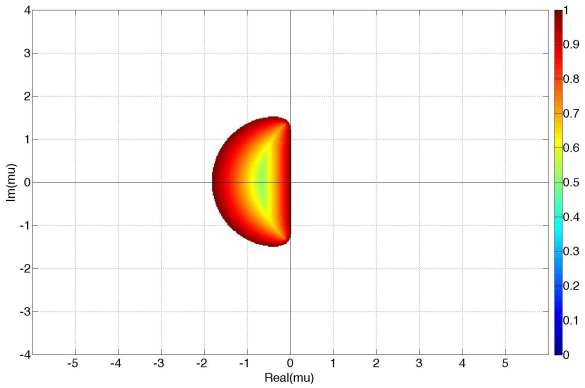
Figure 6.5: Adams-Bashforth absolute stability regions for expressions of order 2, 3, and 4.



a.



b.



c.

Figure 6.6: Adams-Moulton absolute stability regions for expressions of order 3, 4, and 5.

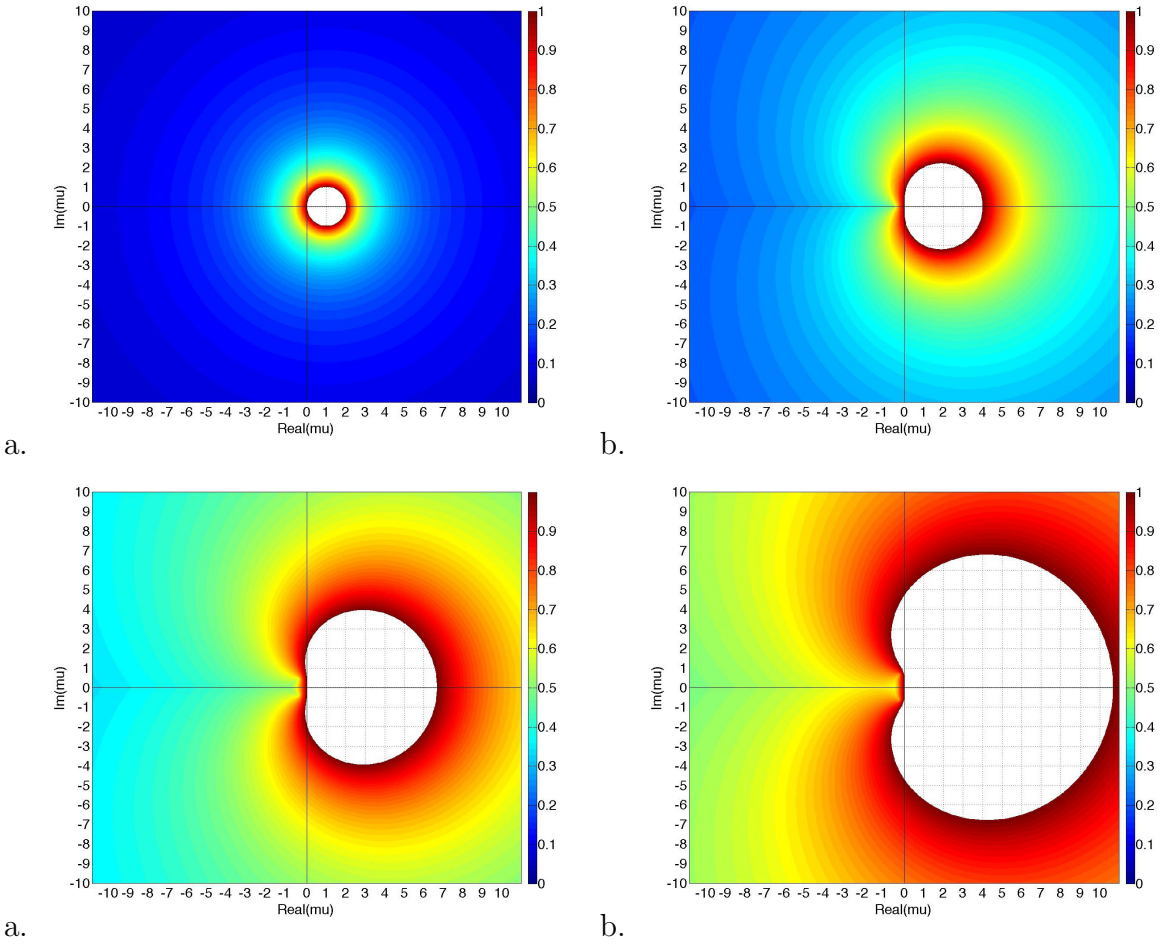
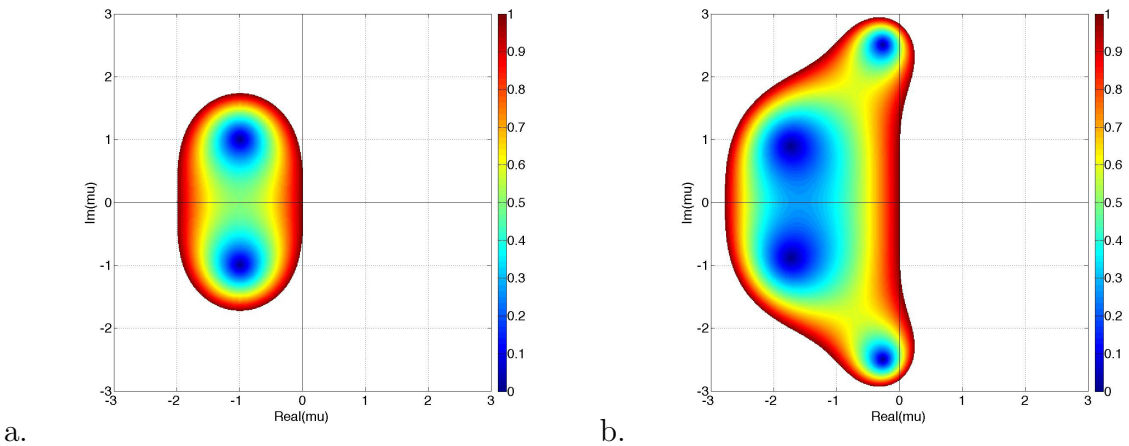


Figure 6.7: Absolute stability regions for backward differentiation formula of order 1, 2, 3 and 4.



a. b.

Figure 6.8: Absolute stability regions for Runge Kutta of second (RK2) and fourth (RK4) order.

coefficients along its diagonals (running from upper left to lower right). This is illustrated for our equation in Figure 6.9.

The spectrum of the matrix A can also be determined analytically, taking advantage of the special structure of its coefficients

$$\lambda_j = -c\,i\,\sin(k_j\Delta x)/\Delta x + 2\,\nu\,\frac{\cos(k_j\Delta x)-1}{\Delta x^2}\,, \quad \text{with} \quad k_j = j - \frac{N}{2} \quad j = 1...N\,. \quad (6.42)$$

For a given discretisation in time (i.e. a given absolute stability diagram), it is now needed that all the discrete eigenvalues lie in the region of absolute stability for the scheme to be stable. This is usually achieved (if the scheme in conditionally stable) by tuning Δt as a scaling factor, which acts has a homotetie on the spectrum in the complex plane (as $\mu = \lambda\Delta t$).

This procedure is illustrated in Figure 6.10.

The Von Neumann stability analysis can thus be understod as a joint study of the discrete eigenvalues and the A-stability property of the time-stepping scheme.

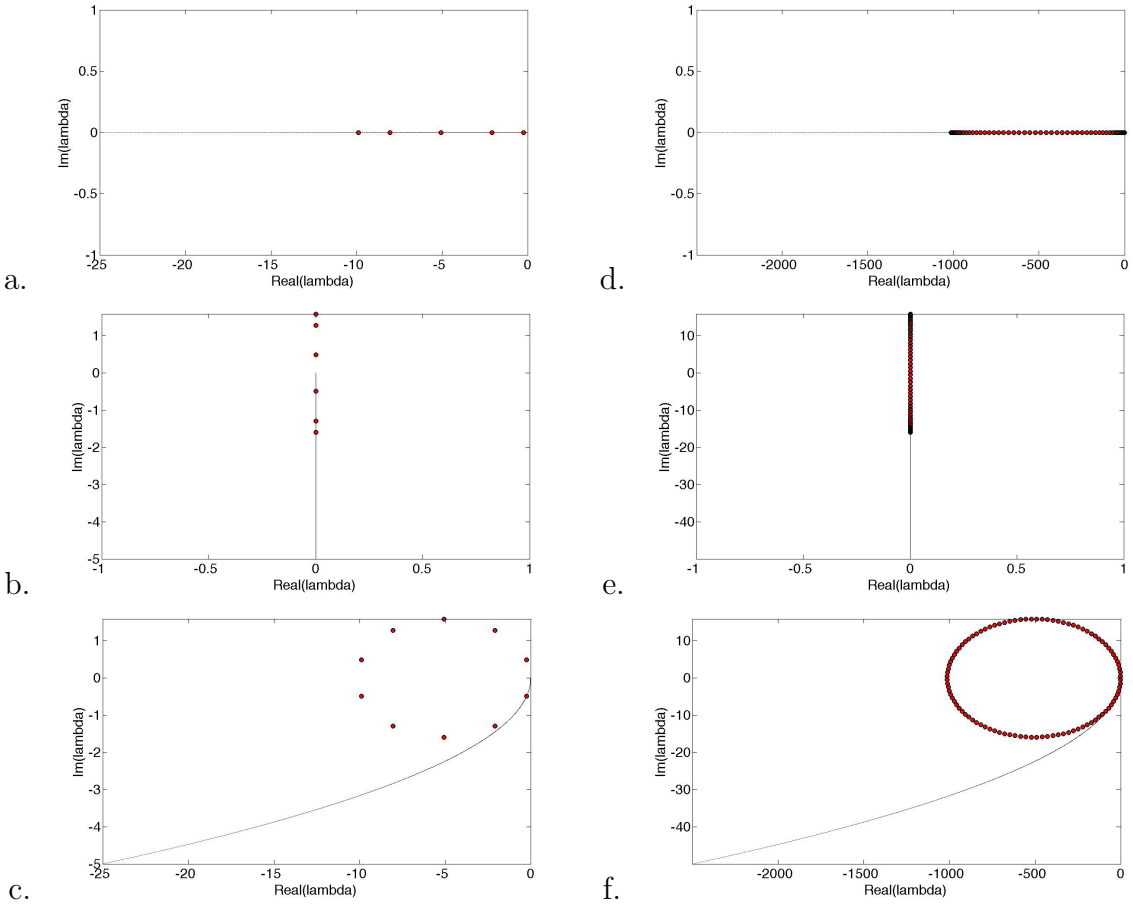
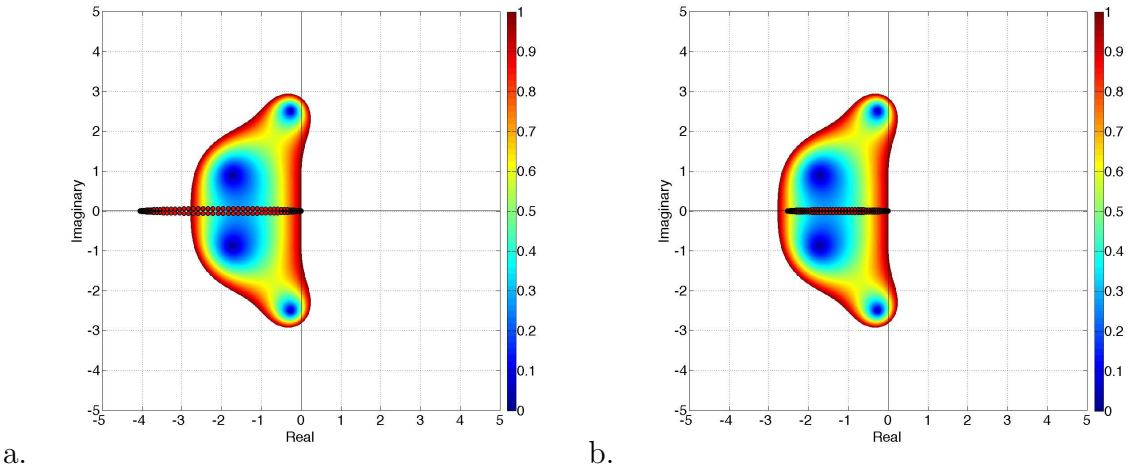


Figure 6.9: Red dots correspond to the discrete spectrum for the advection diffusion problem (top pure diffusion, middle pure advection, bottom advection-diffusion) for $N = 10$ (left) and $N = 100$ (right). The black line indicates the position of the eigenvalues for the continuous problem.



a. b.

Figure 6.10: Unstable and stable schemes. Only Δt is changed between both graphs. The absolute stability region corresponds here to RK4 and the eigenvalues to the semi-discretisation of the advection-diffusion problem introduced in (6.41).

Chapter 7

How to choose? What are the consequences?

7.1 How to chose the numerical strategy

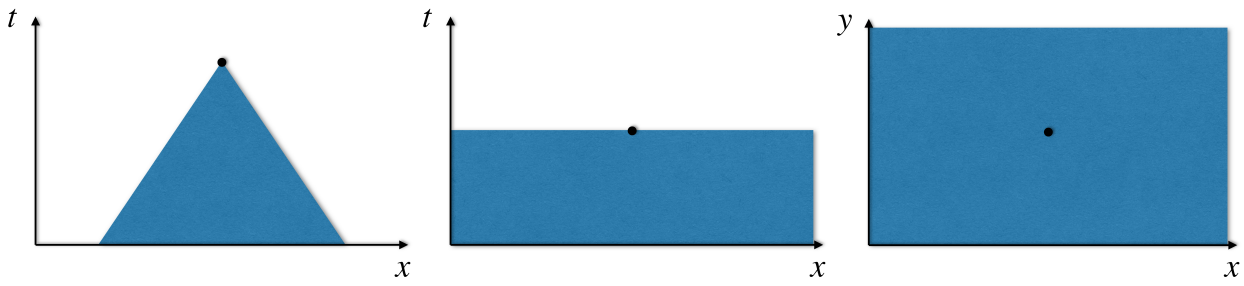
In the preceding chapters, we introduces four different classes of methods. We introduced these on a variety of problems stemming from fluid dynamics. We saw that the choice of numerical method was not free of consequences on the quality of the numerical approximation. An essential aspect of computational fluid dynamics, will therefore be to properly select the numerical method to be used. We provide below elements of thought in doing so...

7.1.1 Different types of problems

The first guide toward choosing the most appropriate method is usually physical insight. When numerically studying a problem derived from a physical application, or a real world experiment, one usually anticipate the expected behavior of the solutions, or the difficulties which could occur. This includes for example the occurrence of localised singularities.

In addition to this physical insight, one should first consider the type of partial differential equation being investigated. By this we mean the classification of partial differential equations.

Partial differential equations are usually classified into three categories: hyperbolic, parabolic and elliptic. This classification provides important information on the domain of dependence of a given value localized in space and time. In an hyperbolic system the information propagates at a finite velocity along characteristics. For this reason, in a time-space diagram, the domain of dependence takes the form of a cone, expanding backward in time. Wave propagation or transport phenomena are typical example of phenomena governed by hyperbolic equations. The parabolic case corresponds to the limit of a propagation at infinite velocity. Each point in space then depends on the entire solution history. Heat or viscous diffusion are governed by this type of equations. Finally



Hyperbolic Parabolic Elliptic

Figure 7.1: Schematic representation of the classification of partial differential equations. The domain of dependence of a given point in the computational domain is indicated with the shaded area.

the elliptic case is the particular case for which the value at one point depends of all the other values. Because of causality, these problems do not involve time. Potential problems (governed by a Laplace equation) are elliptic problems.

The names resemble those of conic sections, this is because these notions are usually introduced on a generic second order linear partial differential equation of the form

$$a\frac{\partial^2 u}{\partial x^2} + 2b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + f\,u = g\,, \tag{7.1}$$

with real coefficients $a \dots g$. The characteristic polynomial associated to this partial differential equation (which is related to the notion of characteristic curves) is then

$$ax^2 + 2bxy + cy^2 + dx + ey + f = 0\,. \tag{7.2}$$

The higher order terms will determine the nature of the equation. One thus studies the sign of the reduced determinant

$$\Delta = b^2 - ac\,. \tag{7.3}$$

If $\Delta < 0$, the equation is said to be elliptic ; if $\Delta = 0$ it is parabolic, if $\Delta > 0$ it is hyperbolic.

This classification has important consequences as it will guide us in the choice of numerical methods. Each type of problem has specific properties, as well as associated difficulties.

Parabolic problems are regularizing. Even a non-smooth initial condition will yield a smooth solution. These problems can be discretised by any of the numerical methods we introduced. The guiding criterion may thus be the geometry of the domain or the measurements required on the solution.

Elliptic problems will always require the resolution of a linear system. A method with a mass matrix (such as compact finite differences or finite volumes) is often selected.

Hyperbolic problems are the most puzzling ones. They are usually associated with conservation laws. In this case the Finite Volume formalism is useful to preserve the conservativity property at the discrete level.

Particular care must be taken with non-linear partial differential equations. In particular non-linear hyperbolic problems, which even with smooth initial data can provide discontinuities in finite time. The again Finite Volumes are usually the method of choice.

7.1.2 Stiff problems

As we have seen in this course, the natural approach before discretising the equations is to write them in non-dimensional form. One is then left with the minimal number of relevant controlling parameters to be investigated. A key question to ask before choosing a numerical method, is the typical orders of magnitude anticipated for these parameters.

It is essential to understand that these numbers being non-dimensional, they can be interpreted as ratios of forces in the equations, ratios of lenghtscale occurring in the solution of these equations, or ratios of typical timescales for the solution. Therefore, if these numbers are extreme (either extremely small or extremely large), one is bound to face difficulties. Such problems are known as "stiff problems". Very small length scale are to be anticipated. In hydrodynamics, this is often associated with a very small prefactor for the viscous term. The result will be extremely small viscous boundary layers or free shear layers. It is often wise to use a stretch grid (to ensure a good resolution in the regions of sharp variations), if the regions of shear are known before hand (usually, linear or weakly non-linear problems). If these regions are not known a priori, one usually implements an adaptative grid strategy.

Care must also be taken in choosing the time stepping strategy. Stiff problems are associated with eigenvalues of very different norm, which correspond to the wide span of time scales. One therefore often recommends using time stepping schemes with wide A-stable domains, such as the Runge-Kutta-4 method see in Chapter 3.

Let us illustrate how stiff problems can behave on a very simple ordinary differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\mathrm{d}f}{\mathrm{d}t} - \varepsilon^{-1}(y - f)\,, \qquad y(t=0) = y_0\,, \tag{7.4}$$

where f is a known function (and $f_0 = f(t=0)$).

Evidently if the initial condition y_0 is chosen such that $y_0 = f_0$, the solution to this equation will trivially be $y \equiv f$.

If, however, one chooses $y_0 \neq f_0$, then the solution takes the form

$$y = f + (y_0 - f_0)\,\mathrm{e}^{-t/\varepsilon}\,. \tag{7.5}$$

This is equivalent to a "boundary layer in time", the discrepancy between the initial condition on y_0 and the value of f_0 will be damped over a very short time scale ε, short compared with the natural variations of f (assumed to be of order one if the system was non-dimensionalised using this natural timescale.

The resolution of such system with an explicit scheme will thus necessarily imply severe restrictions on the time step.

7.1.3 Different types of measurements

Surprisingly, the measurements needed on the solution may also guide the choice of numerical algorithm. For example, many studies in turbulence investigate energy cascades, or the slope of the time-averaged kinetic energy spectrum. In all such cases it is natural to use a pseudo-spectral approach, for which time averaged spectra are computed at no extra cost.

7.1.4 Computational resources

Last, but not least, the discretised system will finally be implemented and solved on a computer. Numerical systems stemming from computational fluid dynamics are usually quite large (in particular for applications in three dimensions of space) and require to be solved using state of the art parallel computers. In such computers, the memory is usually distributed between the computing units (processors, cores). As a result the resolution is usually performed using a domain decomposition strategy (similar to that outlined in Chapter 10). Not all algorithm are well suited for this type of computers. In particular pseudo-spectral, methods require the knowledge of all modes to compute the value at the collocation points. As such, they are not ideally suited for such computers. Parallelisation can still be achieved using a transpose strategy, yet when one aims at large scale parallel computing, it is often wise to advise the use of a grid based method.

7.2 Other methods

7.2.1 Combining these methods

Before we introduce methods which do not fall the the standard scope covered in the previous chapters, it is worth pointing that the above mentioned methods can be combined for the resolution of a given problem. For example, a problem between two infinite plate could be modeled using a periodic domain in one direction and a bounded domain in the orthogonal direction. One can then discretise this problem using a spectral expansion (say Fourier) in the periodic direction and a grid based method (finite difference, finite volumes, or finite elements) in the orthogonal direction.

In the same manner, different time stepping schemes can be combined. For example, in a partial differential equation involving a linear term and a non-linear term as right hand side, it is customary to use an implicit scheme for the linear term and an explicit scheme of the same order for the non-linear term.

7.2.2 Spectral elements

Further advanced methods, not covered in this course can be quotes, for example spectral elements. While not covered in the course, the reader has the knowledge to rapidly learn such methods and see to which class of method they belong...

The spectral element method for example relies on the same formalism as the finite element method, except that high degree piecewise polynomials are being used as basis functions (often Legendre polynomials). This approach thus combines the strength of finite elements with that of spectral methods. Thanks to the domain decomposition in element, the method is flexible to irregularly shaped domains.

This approach therefore allows two independent parameters to improve the quality of the numerical approximation. Either an increase of the number of elements by decreasing the element size h, or an increase of the order of the spectral approximation p on each element. The strategy to increase the quality of the numerical approximation by using these two degrees of freedom is known as hp-adaptivity. h-adaptivity can be understood as splitting elements in space while keeping their polynomial degree fixed, while p-adaptivity consists in increasing the polynomial degree. The hp-adaptivity is significantly different from both h- and p-adaptivity, since the hp-refinement of an element can be done in many different ways. Typically, p-adaptivity will be used in smooth regions of the flow, to take advantage of the exponential convergence associated with a spectral method, whereas h-adaptivity will be used in cases of abrupt changes in the solution.