Computational plasma physics

S. Possanner and E. Sonnendrücker

Max-Planck-Institut für Plasmaphysik
und
Zentrum Mathematik, TU München

Lecture notes
Sommersemester 2020

June 11, 2020
## Contents

1 Introduction .................................................................................. 3
   1.1 Scientific computing .......................................................... 3
   1.2 Plasmas ................................................................................. 5
   1.3 Model equations .................................................................... 6
      1.3.1 The N-body model ....................................................... 8
      1.3.2 Kinetic models ............................................................ 9
      1.3.3 Fluid models .............................................................. 11
   1.4 Outlook ................................................................................. 14

2 Basic numerical tools for the Poisson equation .............................. 15
   2.1 The 1D Poisson equation and boundary conditions ................. 15
   2.2 The finite difference method ................................................ 17
      2.2.1 Higher-order finite differences ................................ .... 20
      2.2.2 Convergence of finite difference schemes ....................... 20
      2.2.3 Finite difference methods in higher dimensions ............... 24
   2.3 The method of manufactured solutions ................................. 26
   2.4 The Finite Element Method .................................................. 26
      2.4.1 Principle of the method .............................................. 26
      2.4.2 The weak form of a boundary value problem ................. 30
      2.4.3 Lagrange Finite Elements ........................................... 34
      2.4.4 B-spline Finite Elements ............................................ 36
      2.4.5 Convergence of the Finite Element method ................. 37
   2.5 Spectral methods I ................................................................. 41
      2.5.1 Orthogonal Legendre polynomials ................................ 42
   2.6 Spectral methods II: Fourier methods ..................................... 44
      2.6.1 Fourier series ............................................................. 45
      2.6.2 The Fourier transform ............................................... 46
      2.6.3 The Discrete Fourier Transform ................................... 46
      2.6.4 Circulant matrices ...................................................... 48
      2.6.5 Stability of the discrete Laplacian with periodic boundary conditions ............................................. 50
      2.6.6 The Fourier pseudospectral method ............................. 51
### 3 Numerical schemes for the Vlasov-Poisson system

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>The Vlasov-Poisson model for electrons</td>
<td>54</td>
</tr>
<tr>
<td>3.1.1</td>
<td>The Vlasov equation in a given potential</td>
<td>55</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Conservation properties</td>
<td>57</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Linearisation of the 1D Vlasov-Poisson equation</td>
<td>58</td>
</tr>
<tr>
<td>3.2</td>
<td>Operator splitting</td>
<td>60</td>
</tr>
<tr>
<td>3.3</td>
<td>The particle in cell (PIC) method</td>
<td>63</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Time scheme for the particles</td>
<td>63</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Particle mesh coupling for Finite Elements</td>
<td>64</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Time loop</td>
<td>64</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Conservation properties at the semi-discrete level</td>
<td>65</td>
</tr>
<tr>
<td>3.4</td>
<td>Monte Carlo simulation</td>
<td>66</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Basics of probability theory</td>
<td>67</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Estimators</td>
<td>68</td>
</tr>
<tr>
<td>3.4.3</td>
<td>PIC Monte Carlo</td>
<td>70</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Control variates (“δf-method”)</td>
<td>70</td>
</tr>
</tbody>
</table>

### 4 Finite volume methods for plasma fluid models

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Hyperbolic systems of conservation laws</td>
<td>72</td>
</tr>
<tr>
<td>4.2</td>
<td>Euler-Poisson equations</td>
<td>73</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Velocity moments of Vlasov</td>
<td>73</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Euler Equations</td>
<td>75</td>
</tr>
<tr>
<td>4.3</td>
<td>The Finite Volume method</td>
<td>76</td>
</tr>
<tr>
<td>4.3.1</td>
<td>The first order Finite Volume schemes</td>
<td>76</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Higher order schemes</td>
<td>78</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Example: FV method for 1D Euler equations</td>
<td>79</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Scientific computing

Understanding an experiment in physics relies on a model which is generally a differential equation or a partial differential equation or a system involving many of these. In sufficiently simple cases analytical solutions of these models exist and then this can be used to predict the behaviour of a similar experiment. However in many cases, especially when the model is based on first principles, it is so complex that there is no analytical solution available. Then there are two options: the first is to simplify the model until it can be analytically solved, the second is to compute an approximate solution using a computer. In practice both are usually done, the simplified models being used to verify that the code is working properly. Due to the enormous development of computer resources in the last 50 years, quite realistic simulations of physical problems become now possible. A large amount of theoretical work in physics and related disciplines, in particular in plasma physics, now relies quite heavily on numerical simulation.

Computational sciences have emerged next to theory and experiments as a third pillar in physics and engineering. Designing efficient, robust and accurate simulation codes is a challenging task that is at the interface of the application domain, plasma physics in our case, applied mathematics and computer science. The main difficulties are to make sure that the implemented algorithms provide a good approximation of the physics model and also that the algorithms use efficiently the available computer resources which are costly. Even though many basic simulations can be performed nowadays on a laptop, state of the art computations require huge super-computers that consists of many computing units and nowadays often heterogeneous computing elements (CPU, GPU, MIC, ...). These require parallel algorithms and often, to achieve optimal efficiency, a very good knowledge of the computer architecture.

This lecture will provide an introduction to the main numerical methods used for plasma physics simulations. Its aim is to introduce the process of developing a simulation code and some standard methods and numerical issues. The models we will consider come from plasma physics, but the models and even more so the techniques and ideas will be useful for many other applications. Specific skills and
methodologies for high performance computing that are also very important in computational physics are beyond the scope of this lecture. We refer to [4] for an introduction.

The first step is to **find an appropriate model**, which is often a set of coupled differential or partial differential equations. If the model is continuous the solution lives in an infinite dimensional space which cannot be represented on a computer. The second step then will be to **discretise it**, i.e. represent the unknowns by a large but finite number of values, typically its values on a finite grid or the coefficients of its expression on the basis of a a finite dimensional linear space. Then from the equations of the starting models relations between the values representing the discrete unknowns should be found. This yields a **finite number of linear or non linear equations** that can be solved on a computer. This will require methods of numerical linear algebra, which are introduced in [15] or iterative methods for linear or nonlinear equations, a good introduction of which is available in [7]. We won’t focus on these either. They are generally taught in Numerics Bachelor classes. They are generally available in numerical software tools like Matlab or Numpy and those programming in a low level language like Fortran, C or C++ can use efficient libraries, like LAPACK, ScALAPACK, PETSc, Trilinos, to name a few, that are freely available.

There are many possible ways to discretise a differential or partial differential equation. They are not all equal and many things need to be considered, when choosing an appropriate one. The most important is naturally that the **discrete model converges towards the initial model** when the number of discrete values goes to infinity. Because computer arithmetics is not exact, as real numbers cannot be represented exactly on a computer, sensitivity to round-off errors is very important. Then some algorithms need more operations than other. Optimally an algorithm dealing with an unknown vector of $N$ points can use $O(N)$ operations, but some can use $O(N \log N)$ or $O(N^d)$ or even more, which will make a huge
difference for large values of $N$. Then again some algorithms will be easier to parallelise than others, which is an important issue when developing a simulation code on a massively parallel computer.

Also, before choosing a discretisation, it is important to understand the structure of the equations that are being discretised. Analytical theory plays an essential role in this aspect. What are the conserved quantities? Are there analytical solution in some special cases? What is the evolution of the solution or some quantities depending on the solution? And so on. The more information is available from analytical theory, the easier it will be to check whether the code is correctly approximating the analytical model. The process of verification of a computer code, consists precisely in checking that the code can reproduce as expected information available from the theory. Verification of a computer code is essential to gain confidence in its correctness. Only once the computer code has been appropriately verified, one can proceed with the validation process, which consists in comparing the code to actual experiments and checking that those can be reproduced within the available error bars. If this is not the case, one needs to check the initial model, including initial and boundary conditions and all external parameters that could have an influence on the results. Possibly one also needs to develop more verification tests to check that there is no error in the implementation. This process of Verification and validation (V & V) is essential for developing a simulation code with reliable predictive capabilities.

In this lecture, starting from a few classical models from plasma physics, we will learn how to write a simulation code for solving them. This includes finding a good discrete model, implementing it and verifying it. This is the scope of applied numerical mathematics. The physics exploitation of the code can start after those steps. We will cover most of the classical discretisation methods, finite differences, finite elements, finite volumes and also spectral methods.

### 1.2 Plasmas

When a gas is brought to a very high temperature ($10^4 K$ or more) electrons leave their orbit around the nuclei of the atom to which they are attached. This gives an overall neutral mixture of charged particles, ions and electrons, which is called plasma. Plasmas are considered beside solids, liquids and gases, as the fourth state of matter.

You can also get what is called a non-neutral plasma, or a beam of charged particles, by imposing a very high potential difference so as to extract either electrons or ions of a metal chosen well. Such a device is usually located in the injector of a particle accelerator.

The use of plasmas in everyday life has become common. This includes, for example, neon tubes and plasma displays. There are also a number industrial applications: amplifiers in telecommunication satellites, plasma etching in micro-electronics, production of X-rays.

We should also mention that while it is almost absent in the natural state on
Figure 1.2: Plasma is the most abundant form of matter in the universe. The bottom-right picture shows the interior of the ASDEX-Upgrade tokamak experiment for fusion energy at the Max-Planck Institute of Plasma Physics in Garching, Germany.

Earth, except the Northern Lights at the poles, the plasma is 99% of the mass of the visible universe. In particular the matter in stars is mainly in the plasma state and the energy they release comes from the process of fusion of light nuclei such as protons. This is the process that is at the base of fusion energy, aiming at extracting a vast amount of energy from a plasma. This is an active subject of research in plasma physics. More information on plasmas and their applications can be found on the web site http://www.plasmas.org.

1.3 Model equations

Since plasma is a gas of charged particles, one expects the equations of gas dynamics (Euler, Navier-Stokes, Boltzmann) to play a role in the modeling of plasma phenomena. In contrast to neutral gases, however, the dominant interaction is governed by the Lorentz force, which is long-range and leads to collective phenomena such as plasma waves. Moreover, external electric and magnetic fields can be used to accelerate and guide the plasma particles - this fact is used for instance in particle accelerators (https://home.cern/) or in magnetic fusion devices (https://www.iter.org/). An adequate description of plasmas is thus obtained by coupling the equations of gas dynamics to equations for the electromagnetic fields, namely Maxwell’s equations. A nice overview of plasma models, and an introduction to Plasma Physics in general, can be found for instance in [2, 9].

Indeed, there exists a whole hierarchy of plasma models adapted to the various scales of observation:
Microscopic (atomistic) scale: $N$-body model. Each of the $N$ plasma particles is modeled by its proper equation of motion (Newton’s equations with Lorentz force).

Mesoscopic scale: kinetic models. Each particle species $s$ is modeled by a density $f_s(x,v,t) > 0$, where $t \in \mathbb{R}$ is time and $(x,v) \in \mathbb{R}^6$ denotes a point in phase space, $x$ being the position and $v$ being the velocity. $f_s(x,v,t) \, d^3x \, d^3v$ is the phase space measure, yielding the number of particles contained in the box $d^3x \, d^3v$. $f_s$ satisfies a transport equation (Boltzmann, Vlasov) and the interaction is modeled via Maxwell’s equations.

Macroscopic scale (1): multi-fluid models Each particle species is modeled by a set of fluid dynamical variables such as particle density $n_s(x,t) > 0$, mean velocity $u_s(x,t) \in \mathbb{R}^3$ and temperature $T_s(x,t) > 0$ (sometimes even more fluid variables such as the heat flux and the pressure tensor are introduced). $(n,u,T)$ satisfy hydrodynamic equations (Euler, Navier-Stokes, Braginskii) and the interaction is modeled via Maxwell’s equations.

Macroscopic scale (2): magneto-hydrodynamic (MHD) models Plasma is modeled as a single, ”quasi-neutral” fluid with mass density $N(x,t) > 0$, flow velocity $U(x,t) \in \mathbb{R}^3$ and pressure $P(x,t) > 0$, coupled to the induction equation for the magnetic field $B(x,t) \in \mathbb{R}^7$.

The $N$-body model is a system of coupled ordinary differential equations (ODEs), where in reality $N$ can be extremely large, e.g. of the order $N \sim 10^{20}$ in a cubic meter of a fusion plasma. All other models are (nonlinear) partial differential equations (PDEs), which describe the evolution of statistical averages of the (exact) $N$-body model over small regions in phase- or configuration space. Fluid models are becoming a good approximation when the particles are close to thermodynamic
equilibrium, to which they return in long time due to the effects of collisions and for which the distribution of particle velocities is a Gaussian.

When choosing a model for a simulation code, one should try to take into account accuracy and computational cost and take the model that will allow us to find a solution that is accurate enough for the problem we are considering in the shortest possible time. In particular because of the very large number of particles in a plasma, kinetic models obtained by statistical arguments are almost always accurate enough. The question will then be if a further model reduction, which could diminish cost, can be performed at least for part of the plasma.

Let us have a brief look at these models to get a flavor of plasma modeling and to know what kind of equations one has to deal with in "Computational Plasma Physics".

### 1.3.1 The N-body model

At the microscopic level, plasma is composed of a number of charged particles that evolve following the laws of classical or relativistic dynamics. So each particle, characterised by its position $\mathbf{x}$, velocity $\mathbf{v}$, as well as its mass $m$ and charge $q$, obeys Newton’s law

$$\frac{d(\gamma m \mathbf{v})}{dt} = \sum F_{\text{ext}},$$

where $\gamma = (1 - \frac{v^2}{c^2})^{-\frac{1}{2}}$ is the Lorentz factor ($c$ being the speed of light). The right hand side $F_{\text{ext}}$ is composed of all the forces applied to the particle, which in our case reduce to the Lorentz force induced by the external and self-consistent electromagnetic fields. Other forces, as the weight of the particles, are in general negligible. Whence we have, labelling the different particles in the plasma,

$$\frac{d(\gamma m_i \mathbf{v}_i)}{dt} = \sum_{j \neq i} q_i (\mathbf{E}_j + \mathbf{v}_i \times \mathbf{B}_j) + q_i (\mathbf{E}_{\text{ext}} + \mathbf{v}_i \times \mathbf{B}_{\text{ext}}).$$

The sum on the right hand side is over all the particles in the plasma and $\mathbf{E}_j, \mathbf{B}_j$ denote the electric and magnetic fields fields generated by particle $j$. Moreover, $\mathbf{E}_{\text{ext}}, \mathbf{B}_{\text{ext}}$ denote the external electric and magnetic fields fields, i.e. those that are not generated by particles of the plasma itself. The latter could be for example coils in an accelerator or in a tokamak. On the other hand the velocity of a particle $\mathbf{v}_i$ is linked to its position $\mathbf{x}_i$ by

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i.$$

Thus, if the positions and velocities of the particles at time $t = 0$ are known as well as the external fields, the evolution of the particles is completely determined.
by the equations
\[
\begin{aligned}
\frac{dx_i}{dt} &= v_i, \\
\frac{d(\gamma_m v_i)}{dt} &= \sum_{j \neq i} q_i (E_j + v_i \times B_j) + q_i (E_{ext} + v_i \times B_{ext}), \quad v_i(0) = v_i^0,
\end{aligned}
\]
where the sum contains the electric and magnetic field generated by each of the other particles as well as the external fields.

### 1.3.2 Kinetic models

In a kinetic model, each species $s$ in the plasma is characterized by a **distribution function** $f_s(x, v, t)$ which corresponds to a statistical mean of the repartition of particles in phase space for a large number of realisations of the considered physical system. Note that phase space consists of the subspace of $\mathbb{R}^6$ containing all possible positions and velocities of the particles. For any volume $V$, $\int_V f_s dx \, dv$ is the number of particles of species $s$, whose position and velocity are in $V$ at time $t$.

Normalising $f_s$ to one, $f_s$ becomes the probability density defining the probability of a particle of species $s$ being at point $(x, v)$ in phase space.

In the limit where the collective effects are dominant on binary collisions between particles, the kinetic equation that is derived, by methods of statistical physics from the $N$-body model is the **Vlasov equation** which reads
\[
\frac{\partial f_s}{\partial t} + v \cdot \nabla_x f_s + \frac{q_s}{m_s} (E + v \times B) \cdot \nabla_v f_s = \sum_{\sigma} Q(f_s, f_{\sigma}), \quad (1.1)
\]
in the non relativistic case. In the relativistic case it becomes
\[
\frac{\partial f_s}{\partial t} + v(p) \cdot \nabla_x f_s + q_s (E + v(p) \times B) \cdot \nabla_p f_s = \sum_{\sigma} Q(f_s, f_{\sigma}). \quad (1.2)
\]

We denote by $\nabla_x f_s$, $\nabla_v f_s$ and $\nabla_p f_s$, the respective gradients of $f_s$ with respect to the three position, velocity and momentum variables. The constants $q_s$ and $m_s$ denote the charge and mass of the particle species. The velocity is linked to the momentum by the relation $v(p) = \frac{p}{m_s \gamma_s}$, where $\gamma$ is the Lorentz factor which can be expressed from the momentum by $\gamma_s = \sqrt{1 + |p|^2/(m^2 c^2)}$. On the right-hand-side $Q(f_s, f_{\sigma})$ represents a bilinear collision operator modeling the collisions of species $s$ with all the other species of particles in the plasma, including $s$. Collisions have the effect of bringing the plasma back to its thermodynamical equilibrium in which the velocity distribution is a Gaussian (also called Maxwellian in plasma and gas dynamics). The Vlasov equation is a generalisation for charged particles of the Boltzmann equation for neutral particles. Note that for many applications on short time scales, the collisions can be neglected and the model becomes then the collisionless Vlasov equation often also called just Vlasov equation. From now on we shall consider the Vlasov equation as collisionless.
Then, for a zero right-hand-side, the Vlasov equation expresses that the distribution function $f_s$ is conserved along the trajectories of the particles which are determined by the mean electric field. We denote by $f_{s,0}(x,v)$ the initial value of the distribution function. The Vlasov equation, when it takes into account the self-consistent electromagnetic field generated by the particles, is coupled to the Maxwell equations which enable to compute this self-consistent electromagnetic field from the particle distribution:

$$\begin{align*}
-\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} + \nabla \times \mathbf{B} &= \mu_0 \mathbf{J}, \\
\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} &= 0,
\end{align*}$$

(Ampère)

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0},$$

(Faraday)

$$\nabla \cdot \mathbf{B} = 0,$$

(Gauss)

$$\nabla \cdot \mathbf{B} = 0,$$

(magnetic Gauss)

where the constants are $c$ the speed of light, $\mu_0$ the permeability of free space and $\varepsilon_0$ the permittivity of free space. The source terms for Maxwell’s equation, the charge density $\rho(x,t)$ and the current density $\mathbf{J}(x,t)$ can be expressed from the distribution functions of the different species of particles $f_s(x,v,t)$ using the relations

$$\begin{align*}
\begin{bmatrix} \rho(x,t) \\ \mathbf{J}(x,t) \end{bmatrix} &= \sum_s q_s \int \begin{bmatrix} 1 \\ v \end{bmatrix} f_s(x,v,t) \, d^3v.
\end{align*}$$

(1.4)

Note that in the relativistic case the distribution function becomes a function of position and momentum (instead of velocity): $f_s \equiv f_s(x,p,t)$ and charge and current densities verify

$$\begin{align*}
\begin{bmatrix} \rho(x,t) \\ \mathbf{J}(x,t) \end{bmatrix} &= \sum_s q_s \int \begin{bmatrix} 1 \\ v(p) \end{bmatrix} f_s(x,p,t) \, d^3p.
\end{align*}$$

(1.5)

In order to admit a unique solution $(f_s, \mathbf{E}, \mathbf{B})$, the Vlasov-Maxwell system (1.1), (1.3), (1.4), or (1.2), (1.3), (1.5) in the relativistic case, needs to be supplemented with suitable initial- and boundary conditions when restricted to a finite domain, as is the case in numerical experiments. A classical boundary condition for the electric field is for example the perfect conductor boundary condition $\mathbf{E} \times \mathbf{n} = 0$, where $\mathbf{n}$ denotes the unit outgoing normal of the $x$-domain. No additional condition on $\mathbf{B}$ is needed in that case. Finding and implementing appropriate boundary conditions is usually among the hardest problems when constructing a numerical code.

An important reduced version of the Vlasov-Maxwell system is the Vlasov-Poisson model, which is electrostatic (in contrast to electromagnetic). One assumes $\mathbf{B} = 0$ such that Maxwell’s equations become

$$\begin{align*}
-\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} &= \mu_0 \mathbf{J}, \\
\nabla \times \mathbf{E} &= 0, \\
\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0}.
\end{align*}$$

(1.6)
Taking the divergence of the first equation, substituting the third equation and using \(c^{-2} = \varepsilon_0\mu_0\) leads to the charge continuity equation

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0.
\] (1.7)

This equation also holds for \(\mathbf{B} \neq 0\) as \(\nabla \cdot \nabla \times = 0\) (or \(\text{div curl} = 0\)). It is a solvability condition on Maxwell’s equations, which is satisfied for \(\rho\) and \(\mathbf{J}\) given by (1.4), as can be verified by direct computation under the assumptions \((??)\) and \((??)\) on the collision operators. Because equation (1.7) holds, the first and the third equation in (1.6) are equivalent and one can neglect one or the other. Moreover, the second equation in (1.6) suggest that the electric field can be written as the gradient of a scalar function, called the electric potential \(\phi\), and one usually writes \(\mathbf{E} = -\nabla \phi\). Therefore, the Vlasov-Poisson (VP) system reads

\[
\begin{cases}
\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{x} f_s - \frac{q_s \nabla \phi}{m_s} \cdot \nabla \mathbf{v} f_s = \sum_{\sigma} Q(f_s, f_{\sigma}), \\
-\nabla \cdot \nabla \phi = \frac{1}{\varepsilon_0} \sum_s q_s \int f_s(\mathbf{x}, \mathbf{v}, t) \, d^3\mathbf{v} \quad \text{(Poisson)},
\end{cases}
\]

supplemented with suitable initial and boundary conditions. This model features some interesting phenomena like plasma waves (Langmuir waves) and Landau damping, the latter being a purely kinetic effect which cannot be captured with fluid models.

### 1.3.3 Fluid models

The idea of fluid models is to describe each plasma species in terms of macroscopic variables, i.e., variables that do not depend on the velocity \(\mathbf{v}\), rather than the full distribution function \(f_s\). The most common such variables are the particle density \(n_s\), the mean velocity \(\mathbf{u}_s\), and the total energy \(w_s\), which correspond to the first three velocity moments of \(f_s\):

\[
\begin{bmatrix}
n_s(\mathbf{x}, t) \\
n_s(\mathbf{x}, t) \mathbf{u}_s(\mathbf{x}, t) \\
\vdots \\
w_s(\mathbf{x}, t)
\end{bmatrix} := \int \begin{bmatrix}
1 \\
\mathbf{v} \\
\vdots \\
\mathbf{v}
\end{bmatrix} \frac{m_s|\mathbf{v}|^2/2}{m_s^2} f_s(\mathbf{x}, \mathbf{v}, t) \, d^3\mathbf{v}. \tag{1.8}
\]

The total energy can be written as the sum of internal energy and kinetic energy, which leads to the definition of temperature \(T_s\):

\[
w_s = \frac{m_s}{2} \int |\mathbf{v}|^2 f_s \, d^3\mathbf{v}
\]

\[
= \frac{m_s}{2} \int |\mathbf{v} - \mathbf{u}_s|^2 f_s \, d^3\mathbf{v} + m_s \int \mathbf{v} \cdot \mathbf{u} f_s \, d^3\mathbf{v} - \frac{m_s}{2} \int |\mathbf{u}|^2 f_s \, d^3\mathbf{v}
\]

\[
= \frac{3}{2} n_s k_B T_s + n_s \frac{m_s|\mathbf{u}_s|^2}{2}. \tag{\text{internal energy, kinetic energy}}
\]

11
Remark that \( \mathbf{v} \cdot \mathbf{u} = \mathbf{v}^T \mathbf{u} \) denotes the inner product. Here, we defined the species temperature \( T_s \) via

\[
3 n_s k_B T_s := m_s \int |\mathbf{v} - \mathbf{u}_s|^2 f_s \, d^3 \mathbf{v},
\]

where \( k_B \) denotes the Boltzmann constant. The pressure of species \( s \) is given by \( p_s = n_s k_B T_s \). By comparing (1.8) to (1.4) we can see that the charge- and current densities can be written as

\[
\begin{bmatrix} n \cr J \end{bmatrix} = \sum_s q_s \begin{bmatrix} n_s \cr n_s \mathbf{u}_s \end{bmatrix}.
\]

Fluid models consist of a closed set of equations for a suitable set of fluid variables, for instance \((n_s, \mathbf{u}_s)\) or \((n_s, n_s \mathbf{u}_s, w_s)\). The model equations are obtained by integrating the Vlasov equation over velocity space. However, the moment hierarchy (1.8) is not closed because the equation for the \( n \)-th moment depends on the \((n + 1)\)-th moment. Therefore, a closure relation is needed. The closure relation can be based on physical or on mathematical arguments, or both. A common closure is the assumption of local thermal equilibrium, which means that \( f_s \) is a Gaussian in velocity \( \mathbf{v} \) at all times:

\[
\text{local thermal equilibrium:} \quad f_s(\cdot,t) = \mathcal{M}_{n_s, \mathbf{u}_s, T_s}(\cdot,t) \quad \forall t.
\]

The equilibrium distribution \( \mathcal{M}_{n_s, \mathbf{u}_s, T_s} \) is called Maxwellian and is defined by

\[
\mathcal{M}_{n_s, \mathbf{u}_s, T_s} := n_s \left( \frac{m_s}{2 \pi k_B T_s} \right)^{3/2} \exp \left( -\frac{m_s |\mathbf{v} - \mathbf{u}_s|^2}{2 k_B T_s} \right).
\]

It can be readily checked by computing the Gaussian integrals that

\[
\int \begin{bmatrix} \mathbf{v} \\ m_s |\mathbf{v}|^2/2 \end{bmatrix} \mathcal{M}_{n_s, \mathbf{u}_s, T_s} \, d^3 \mathbf{v} = \begin{bmatrix} n_s \\ n_s \mathbf{u}_s \\ w_s \end{bmatrix},
\]

such that each Maxwellian is uniquely determined by its first three velocity moments \((n_s, n_s \mathbf{u}_s, w_s)\). In case the particle collisions are neglected, and under the Maxwellian closure (1.10), these moments satisfy the Euler-Lorentz equations

\[
\begin{aligned}
\frac{\partial n_s}{\partial t} + \nabla \cdot (n_s \mathbf{u}_s) &= 0, \\
\frac{\partial}{\partial t} (n_s \mathbf{u}_s) + \nabla \cdot (n_s \mathbf{u}_s \mathbf{u}_s^T) + \frac{\nabla p_s}{m_s} &= \frac{q_s}{m_s} (n_s \mathbf{E} + n_s \mathbf{u}_s \times \mathbf{B}), \\
\frac{\partial w_s}{\partial t} + \nabla \cdot [\mathbf{u}_s (w_s + p_s)] &= q_s n_s \mathbf{u}_s \cdot \mathbf{E}.
\end{aligned}
\]

Here, \((\mathbf{u} \mathbf{u}^T)_{i,j} = u_i u_j\) is the outer product. The Euler-Lorentz equations follow directly by integrating the Vlasov equation over \((1, \mathbf{v}, m_s |\mathbf{v}|^2/2)\) and by inserting the assumption (1.10). The dimensional reduction with respect to the kinetic approach, where \( f_s(\cdot,t) \) is defined on \( \mathbb{R}^6 \) whereas fluid variables are defined on \( \mathbb{R}^3 \), greatly reduces the computational complexity in numerical simulations. However, the fluid description is less accurate than the kinetic one. Only if we would compute
all (infinitely many) velocity moments of \( f_s \) would the fluid solution be equivalent to the kinetic solution.

The fluid system (1.11) together with (1.3) and (1.9) forms a closed set of equations for plasma with self-consistent electromagnetic interaction (Euler-Maxwell model). There are other variants of the fluid equations which take into account small deviations from the Maxwellian (compressible, incompressible Navier-Stokes) or the effects of a very strong magnetic field (Braginskii equations).

Yet another popular fluid model in plasma physics is **magneto-hydrodynamics (MHD)**, a theory that describes a multi-species plasma as a single fluid (see for instance [6] for a nice introduction to MHD). The MHD variables are the total mass \( N \), the total velocity \( U \) and the total pressure \( P \) of the plasma, defined by

\[
\text{MHD variables: } N := \sum_s m_s n_s, \quad U := \frac{1}{N} \sum_s m_s n_s u_s, \quad P := \sum_s p_s.
\]

Under suitable assumptions such as quasi-neutrality \((\rho \to 0)\) and mass-less electrons \((m_e \to 0)\) one can derive from the fluid equations (1.11) the following set of MHD equations:

\[
\begin{align*}
\frac{\partial N}{\partial t} + \nabla \cdot (NU) &= 0, \\
\frac{\partial (NU)}{\partial t} + \nabla \cdot (NUU^\top) + \nabla P &= \frac{1}{\mu_0} (\nabla \times B) \times B, \\
\frac{\partial P}{\partial t} + \nabla \cdot (PU) + (\gamma - 1) P \nabla \cdot U &= 0, \\
\frac{\partial B}{\partial t} - \nabla \times (U \times B) &= 0, \quad \text{(Induction equation)}.
\end{align*}
\]

This is a system for the unknowns \((N, U, p, B)\) which describes plasma as a single fluid. Maxwell’s equations were reduced to the induction equation for \( B \), and the electric field has been eliminated from the system (it actually reads \( E = -U \times B \)).

Fluid models are popular because they are intuitive and deal with quantities that are measurable in experiments. However, the distribution function \( f_s \) contains much more information than a fluid description as it includes information on the distributions of particle velocities at each position \( x \). A kinetic description of a plasma is essential when the distribution function is far away from the Maxwell distribution \( M_{n_s, u_s, T_s} \). This is the case if collisions are not frequent enough, as in rarefied gases, because collisions are the mechanism that drives the distribution function towards local thermal equilibrium (H-theorem). In case of high collision frequency (dense gas or liquid) a fluid description can be sufficient. In practice one often deals with a combination of kinetic and fluid models for different parts of the plasma (hot and cold particles, core and edge regions in a tokamak). Such models are called **kinetic-fluid hybrid**. An understanding of the transition from the kinetic to the fluid description is paramount for these ideas.
1.4 Outlook

We have seen that mesoscopic (kinetic) and macroscopic (fluid) plasma models are nonlinear systems of PDEs. In this lecture we will learn about some basic strategies of how to obtain approximate solutions of these equations on a computer, and how to verify the correct implementation of the codes. Let us give a brief overview of the considered topics. In the following list \( f \) denotes the exact solution of some PDE under consideration and \( (f^n_i)_{i,n} \) stand for discrete numbers obtained from a computer code approximating this PDE. Moreover, let \((t_n, x_i)_{i,n}\) denote discretisation (grid-, or mesh-) points of time and space. The following numerical methods will be covered in this lecture:

1. **Finite difference method** (point-wise approach):
   \[
   f(t_n, x_i) \approx f^n_i, \quad \frac{\partial f}{\partial x} \approx \frac{f_{i+1} - f_i}{h}, \quad \frac{\partial^2 f}{\partial x^2} \approx \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2}
   \]

2. **Finite element method** (FEM - local approach)
   \[
   f(t_n, x) \approx \sum_i f^n_i N_i(x), \quad N_i \text{ is non-zero only close to } x_i \text{ (local support)}
   \]

3. **Spectral methods** (global approach)
   \[
   f(t_n, x) \approx \sum_i f^n_i P_i(x), \quad P_i \text{ is non-zero everywhere (global support)}
   \]

4. **Finite volume methods** for conservation laws (cell-average approach)
   \[
   \int_{x_i}^{x_{i+1}} f(t_n, x) \, dx \approx f^n_i, \quad \text{relies on good approx. of flux function}
   \]

5. **Particle-in-cell method** for kinetic plasma models (Lagrangian approach)
   \[
   f(t_n, x) \approx \sum_i w_i \delta(x - x_i(t_n)), \quad \frac{dx_i}{dt} = \ldots
   \]
Chapter 2

Basic numerical tools for the Poisson equation

2.1 The 1D Poisson equation and boundary conditions

The Poisson equation is a fundamental equation in plasma physics, as it enables the computation of the electric potential $\phi$ for a given charge density $\rho$. It is also the prototypical elliptic equation in many textbooks on PDEs. Consider an open bounded and connected set $\Omega \subset \mathbb{R}^n$ with smooth boundary $\partial \Omega$. The Poisson equation reads

$$-\Delta \phi(x) = \rho(x), \quad x \in \Omega \subset \mathbb{R}^n,$$

(2.1)

where $\rho: \Omega \to \mathbb{R}$ is a given function and $\Delta = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}$ denotes the Laplace operator. To obtain a unique solution, suitable boundary conditions on $\partial \Omega$ must be specified. Equation (2.1) is actually the strong form of the Poisson equation, as it holds point-wise in $\Omega$. $\phi$ is then assumed twice continuously differentiable and $\rho$ is continuous, such that the above expression makes sense. Later, in the context of finite element methods, we will get to know also the weak form of Poisson, where less regularity is demanded from the solution.

In order to get familiar with the basic discretization methods, we consider the 1D Poisson equation ($n = 1$) in an interval $\Omega = (a, b) \subset \mathbb{R}$ and assume $\phi \in C^2(\Omega)$ and $\rho \in C^0(\Omega)$:

$$-\phi''(x) = \rho(x), \quad x \in (a, b).$$

(2.2)

For the problem to be well posed, boundary conditions are needed for $x = a$ and $x = b$. We will consider here three classical types of boundary conditions:

1. **Dirichlet boundary conditions**: $\phi$ is given at $x = a$ and $x = b$,

$$\phi(a) = \alpha, \quad \phi(b) = \beta.$$

(2.3)

2. **Neumann boundary conditions**: $\phi'$ is given at boundary. Note that we can do this only at one end of the interval, otherwise the problem is ill-posed
(see below):

\[ \phi(a) = \alpha, \quad \phi'(b) = \beta. \]  \hfill (2.4)

3. **Periodic boundary conditions.** In this case the problem is actually defined on \( \Omega = \mathbb{R} \) but all functions are assumed to be periodic of period \( L = b - a \):

\[ \phi(x + L) = \phi(x), \quad \forall \ x. \]  \hfill (2.5)

Therefore we can restrict the interval of computation to \([0, L]\). Then, there are mathematically no boundaries and no boundary conditions are needed. The terminology "periodic boundary conditions" is somewhat misleading. Note however in this case that \( \phi \) is only determined up to a constant, which needs to be set for a numerical computation.

Let us examine the existence and uniqueness of solutions to problem (2.2) with the three types of boundary conditions (2.3), (2.4) and (2.5). The existence follows from the regularity of the right-hand-side. Because \( \rho \in C^0(\Omega) \) we can compute anti-derivatives (primitive functions):

\[
\begin{align*}
\phi'(x) &= -\int_a^x \rho(x_1) \, dx_1 + c_1, \\
\phi(x) &= -\int_a^x \left( \int_a^{x_1} \rho(x_0) \, dx_0 \right) \, dx_1 + c_1 x + c_0,
\end{align*}
\]

where \( c_1, c_2 \in \mathbb{R} \) are the constants of integration. The lower bound \( 'a' \) in the integration is not important and could be changed because we added \( c_1 \) and \( c_2 \). The uniqueness can be investigated by assuming that there exist two solutions \( \phi \) and \( \psi \) that satisfy (2.2) plus boundary conditions. The difference \( \eta := \phi - \psi \) then satisfies the Laplace equation \(-\eta'' = 0\) and hence is given by \( \eta(x) = c_0 + c_1 x \), where \( c_0, c_1 \in \mathbb{R} \) have to be determined from the boundary conditions. For the problem to be well-posed one needs \( c_0 = c_1 = 0 \) because then \( \phi = \psi \) and the solution is unique. In case of Dirichlet conditions one obtains

\[
\begin{align*}
\eta(a) = c_0 + c_1 a = 0 \\
\eta(b) = c_0 + c_1 b = 0
\end{align*}
\]

\[ \Rightarrow \quad c_1(a - b) = 0 \quad \Rightarrow \quad c_1 = 0 \quad \Rightarrow \quad c_0 = 0, \]

such that the problem is well-posed. In case of (2.4), with a Neumann condition at \( x = b \) one has

\[
\begin{align*}
\eta(a) = c_0 + c_1 a = 0 \\
\eta'(b) = c_1 = 0
\end{align*}
\]

\[ \Rightarrow \quad c_0 = 0, \]

so that also this problem is well-posed. Had we assumed Neumann conditions at both sides of the domain, the constant \( c_0 \) could be chosen arbitrarily, leading to an infinite amount of solutions. Similarly, for periodic boundary conditions we obtain from (2.5) only one equation,

\[ \eta(x) = c_0 + c_1 x = c_0 + c_1(x + L) = \eta(x + L), \]
such that also in this case the constant $c_0$ remains undetermined, and the problem is ill-posed. Moreover, integrating (2.2) on a period, e.g. $[0, L]$ yields

$$0 = \phi'(L) - \phi'(0) = -\int_0^L \rho(x) \, dx,$$

as $\phi'(L) = \phi'(0)$ because $\phi'$ is $L$-periodic. So a necessary condition for a solution to exist for the periodic case is $\int_0^L \rho(x) \, dx = 0$.

### 2.2 The finite difference method

We now make a first attempt at discretising the 1D Poisson equation (2.2). Let us assume for simplicity that $a = 0$ and $b = L$ and let us consider a uniform mesh of the 1D computational domain $\Omega = [0, L]$. The cell size or space step is defined by $h = \frac{L}{N}$ where $N$ is the number of cells in the mesh. The coordinates of the grid points are then defined by $x_j = x_0 + jh = jh$ as $x_0 = 0$. The solution of the numerical scheme will be defined by its values at $x_j$ for $j = 0, \ldots, N$, hence at $N + 1$ grid points.

The principle of the Finite Difference (FD) method is to approximate derivatives of $\phi$ at grid points $x_j$ by differences of $\phi(x_i)$ at well-chosen (neighbouring) grid points $x_i$. The simplest way to do this is to use Taylor expansions around the considered point $x_j$. We do this for all points on the grid. The Taylor expansion will also enable us to see the order of approximation of the derivative. In particular, assuming $\phi \in C^4(\Omega)$ we have

$$\phi(x_{j+1}) = \phi(x_j) + h \phi'(x_j) + \frac{h^2}{2} \phi''(x_j) + \frac{h^3}{6} \phi^{(3)}(x_j) + \frac{h^4}{24} \phi^{(4)}(x_j + \theta^+_j h),$$

$$\phi(x_{j-1}) = \phi(x_j) - h \phi'(x_j) + \frac{h^2}{2} \phi''(x_j) - \frac{h^3}{6} \phi^{(3)}(x_j) + \frac{h^4}{24} \phi^{(4)}(x_j - \theta^-_j h).$$

We deduce

$$\phi(x_{j+1}) - 2\phi(x_j) + \phi(x_{j-1}) = h^2 \phi''(x_j) + O(h^4),$$

so that

$$\phi''(x_j) = \frac{\phi(x_{j+1}) - 2\phi(x_j) + \phi(x_{j-1})}{h^2} = O(h^2).$$

Plugging this into the Poisson equation at point $x_j$, $-\phi''(x_j) = \rho(x_j)$, we get

$$-\frac{\phi(x_{j+1}) - 2\phi(x_j) + \phi(x_{j-1})}{h^2} = \rho(x_j) + O(h^2).$$

(2.6)

Let us now define $\phi_j \in \mathbb{R}$ such that for $1 \leq j \leq N - 1$ we have

$$-\frac{\phi_{j+1} + 2\phi_j - \phi_{j-1}}{h^2} = \rho(x_j),$$

and let us use the boundary conditions to determine the additional unknowns $\phi_0$ and $\phi_N$. Then $(\phi_j)_{j=0}^N$ will give an approximation of $\phi(x_j)$ for all the points on the
grid $0 \leq j \leq N$.

**Dirichlet:** $\phi_0 = \phi(x_0) = \alpha$, $\phi_N = \phi(x_N) = \beta$. So there remain $N - 1$ unknowns $\phi_1, \ldots, \phi_{N-1}$ determined by the $N - 1$ equations

$$-\frac{\phi_{j+1} + 2\phi_j - \phi_{j-1}}{h^2} = \rho(x_j), \quad 1 \leq j \leq N - 1.$$ 

This can be written as a linear system $A_h \Phi_h = R_h$ with

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & & & & & & \\ -1 & 2 & -1 & 0 & & & & & \\ & 0 & \ddots & \ddots & \ddots & & & & \\ & & \ddots & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \ddots & \ddots & \\ & & & & & 0 & -1 & 2 & -1 \\ & & & & & & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(N-1) \times (N-1)},$$ \quad (2.7)

$$\Phi_h = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{pmatrix} \in \mathbb{R}^{(N-1)}, \quad R_h = \begin{pmatrix} \rho(x_1) + \frac{\alpha}{h^2} \\ \rho(x_2) \\ \vdots \\ \rho(x_{N-1}) + \frac{\beta}{h^2} \end{pmatrix} \in \mathbb{R}^{(N-1)}.$$

**Neumann:** Because we need to set the constant for the potential at one point, we consider now the boundary conditions $\phi(x_0) = 0$ and $\phi'(x_N) = \beta$. In this case the unknown are $\phi_1, \ldots, \phi_N$. So there are $N$ unknowns. We can still use like before the finite difference approximations of $-\phi''(x_j) = \rho(x_j)$ at the $N - 1$ interior points. Then the missing equation needs to be obtained from the Neumann boundary condition $\phi'(L) = \beta$. This needs to be expressed from the point values. For this we can substitute

$$\frac{\phi_{N+1} - \phi_N}{h} = \beta,$$

in the Poisson equation at point $x_N$, to express $\phi_{N+1}$. This is only a first order approximation of the derivative at $x_N = L$, but this is enough to keep the second order approximation on $\phi$ at the end (this is shown in the exercise classes). Now we get the linear system $A_h \Phi_h = R_h$ with

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & & & & & & \\ -1 & 2 & -1 & 0 & & & & & \\ & 0 & \ddots & \ddots & \ddots & & & & \\ & & \ddots & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \ddots & \ddots & \\ & & & & & 0 & -1 & 2 & -1 \\ & & & & & & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{N \times N},$$

$$\Phi_h = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{pmatrix} \in \mathbb{R}^N, \quad R_h = \begin{pmatrix} \rho(x_1) \\ \rho(x_2) \\ \vdots \\ \rho(x_{N-1}) + \frac{\beta}{h} \end{pmatrix} \in \mathbb{R}^N.$$
**Periodic:** This case is the simplest as there is in principle no boundary. All points are interior points and can be used to express $-\phi''(x_j) = \rho(x_j)$. Because of the periodicity $\phi(x_{j+N}) = \phi(x_j)$, only the values of $\phi_j$ for $0 \leq j \leq N - 1$ need to be computed, the others being deduced by periodicity. So there will be $N$ unknowns in our system that are determined by the $N$ approximations to $-\phi''(x_j) = \rho(x_j)$.

At interior points we have

$$\frac{-\phi_{j+1} + 2\phi_j - \phi_{j-1}}{h^2} = \rho(x_j), \quad 1 \leq j \leq N - 2.$$  

Moreover, for $j = 0$ we have $\phi_{j-1} = \phi_{-1} = \phi_{N-1}$ so that the equation reads

$$\frac{-\phi_1 + 2\phi_0 - \phi_{N-1}}{h^2} = \rho(x_0),$$

and for $j = N - 1$ we have $\phi_{j+1} = \phi_N = \phi_0$ so that

$$\frac{-\phi_0 + 2\phi_{N-1} - \phi_{N-2}}{h^2} = \rho(x_{N-1}).$$

In this case the system in Matrix form reads

$$A_h \Phi_h = R_h,$$

with

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N \times N}, \quad (2.8)$$

$$\Phi_h = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{N-1} \end{pmatrix} \in \mathbb{R}^N, \quad R_h = \begin{pmatrix} \rho(x_0) \\ \rho(x_1) \\ \vdots \\ \rho(x_{N-1}) \end{pmatrix} \in \mathbb{R}^N.$$  

We notice that each diagonal of $A_h$ has everywhere the same term. We see also that the vector $(1, \ldots, 1)^T$ is in the kernel of $A_h$ as the sum of all the terms of each line vanishes. This means that the matrix is not invertible. Its kernel has rank one and invertibility can be recovered by assuming zero average, which in the discrete case reads

$$\phi_0 + \cdots + \phi_{N-1} = 0.$$  

In practice, to solve the system, one could replace the last row of the linear system by the condition above. Another option, that we will come back to later, would be to use the fact that the matrix $A_h$ is circulant, and solve the system using a Fast Fourier Transform (FFT), setting the zero mode to zero as is implied by the fact that $\phi$ has zero average.
2.2.1 Higher-order finite differences

A fourth order formula for the second derivative can be obtained by adding Taylor expansions expressing in addition \( \phi(x_j+2) \) and \( \phi(x_j-2) \) with respect to \( \phi(x_j) \) and its derivatives at the point \( x_j \). Taking linear combinations of the four Taylor expansions such that all terms up to \( h^5 \), except of course the function values and the second derivative vanish. We then get the formula

\[
\phi''(x_j) = \frac{-\phi(x_{j+2}) + 16\phi(x_{j+1}) - 30\phi(x_j) + 16\phi(x_{j-1}) - \phi(x_{j-2})}{12h^2} + O(h^4).
\]

This can be used everywhere for periodic boundary conditions. In this case the matrix form of the Finite Difference problem reads

\[
A_h \Phi_h = R_h
\]

with

\[
A_h = \frac{1}{h^2} \begin{pmatrix}
\frac{30}{12} & -\frac{16}{12} & \frac{1}{12} & 0 & 0 & \frac{1}{12} & -\frac{16}{12} \\
-\frac{16}{12} & \frac{30}{12} & -\frac{16}{12} & \frac{1}{12} & 0 & \cdots & 0 \\
\frac{1}{12} & -\frac{16}{12} & -\frac{16}{12} & \frac{1}{12} & 0 & \cdots & 0 \\
0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
0 & \cdots & \cdots & \frac{1}{12} & 0 & \cdots & 0 \\
\frac{1}{12} & \frac{1}{12} & \frac{1}{12} & -\frac{16}{12} & \frac{30}{12} & -\frac{16}{12} & \frac{16}{12} \\
-\frac{16}{12} & \frac{1}{12} & \frac{12}{12} & 0 & \frac{1}{12} & -\frac{16}{12} & \frac{30}{12} \\
\end{pmatrix} \in \mathbb{R}^{N \times N}
\]

\[
\Phi_h = \begin{pmatrix}
\phi_0 \\
\phi_1 \\
\vdots \\
\phi_{N-1}
\end{pmatrix} \in \mathbb{R}^N, \quad R_h = \begin{pmatrix}
\rho(x_0) \\
\rho(x_1) \\
\vdots \\
\rho(x_{N-1})
\end{pmatrix} \in \mathbb{R}^N.
\]

For other types of boundary conditions a non centred formula of the same order needs to be applied at \( x_1 \) and \( x_{N-1} \).

2.2.2 Convergence of finite difference schemes

Some theory on the convergence of finite difference schemes will help us understand what is needed for a good scheme and also provide verification tests for checking that the code implementing the scheme behaves correctly. In particular, checking the order of convergence of a scheme is a very good verification test that should be used whenever some theoretical order exists.

In this lecture, we will use mostly the decomposition in eigenmodes for our proofs. Sometimes easier and more general proofs exist, but understanding the behaviour of the eigenmodes is in general very useful to understand a scheme. Some continuous and discrete norms are needed to define rigorously the convergence.
We will use mostly the $L^1$, $L^2$ and $L^\infty$ (or max) norms defined as follows. In the continuous setting

$$\|f\|_1 = \int_a^b |f(x)| \, dx, \quad \|f\|_2 = \left( \int_a^b |f(x)|^2 \, dx \right)^{1/2}, \quad \|f\|_\infty = \max_{a \leq x \leq b} |f(x)|.$$ 

In the discrete setting

$$\|v\|_1 = \frac{1}{N} \sum_{j=1}^N |v_i|, \quad \|v\|_2 = \left( \frac{1}{N} \sum_{j=1}^N |v_i|^2 \right)^{1/2}, \quad \|v\|_\infty = \max_i |v_i|.$$ 

Recall that in a finite dimensional space, all norms are equivalent. For example, the 2-norm and the max-norm can be compared via

$$\frac{1}{\sqrt{N}} \|v\|_\infty \leq \|v\|_2 \leq \|v\|_\infty, \quad \forall \ v \in \mathbb{R}^N. \quad (2.9)$$

After discretisation with Finite Differences we obtained in each case a linear system of the form $A_h \Phi_h = R_h$. If the matrix $A_h$ is invertible, this enables to compute the solution vector $\Phi_h = (\phi_j)_j$. For this to provide a good approximation of the true solution of the Poisson equation, we must compare $\Phi_h$ to the values of the true solution at the grid points, which we denote by $\Phi := (\phi(x_j))_j$ and which we don’t know in general. Our goal is to verify that for some norm $\| \cdot \|$ we have

$$\|\Phi - \Phi_h\| \leq C h^p$$

for some integer $p \geq 1$ and a constant $C$ independent of $h$ (we then say that $\|\Phi - \Phi_h\| = O(h^p)$ as $h \to 0$). For this we need to show two properties of the scheme, namely consistency and stability:

consistency: $\lim_{h \to 0} A_h(\Phi - \Phi_h) = 0$,

stability: $\|A_h^{-1}\| \leq C_s$,

where $C_s$ is independent of $h$. Consistency can be checked by inserting the true solution into our scheme. For example, because of (2.6), we have that $A_h \Phi = R_h + h^2 S_h$, where $S_h$ is the vector containing the rest term of the Taylor expansions. Then as $A_h \Phi_h = R_h$, it follows that $A_h(\Phi - \Phi_h) = h^2 S_h$. Assuming that the fourth derivative of $\phi$ is bounded, it follows that

$$\|S_h\|_\infty \leq C = (\max_{x \in [0,L]} |\phi^{(4)}(x)|)/12,$$

$$\|A_h(\Phi - \Phi_h)\|_\infty \leq C h^2,$$

where $C$ is independent of $h$. We then say that the numerical scheme defined by the matrix $A_h$ is consistent of order 2 in the max-norm. For the 2-norm, we can use the norm comparison (2.9) and $h = L/N$ to get that

$$\|A_h(\Phi - \Phi_h)\|_2 \leq C_2 h^2,$$
where $C_2$ is independent of $h$.

Consistency gives us convergence of $A_h(\Phi - \Phi_h)$ to zero. This is not yet enough to prove that $\Phi_h$ converges to $\Phi$. We also need stability, namely that the norm of the inverse of $A_h$, $\|A_h^{-1}\|$, is bounded independently of $h$.

**Definition 1.** For a given vector norm $\|\cdot\|$, we define the induced matrix norm by

$$\|A\| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$  

The consistency of a Finite Difference scheme comes directly from its derivation using Taylor expansions. Its stability is often more difficult to verify. One possible way to do it is to check its eigenvalues. This relies on the following proposition:

**Proposition 1.** Let $A \in \mathbb{R}^{N \times N}$ be diagonalisable in a basis of orthonormal eigenvectors and let $\lambda_1, \ldots, \lambda_N$ denote the eigenvalues of $A$. Then we have

$$\|A\|_2 \leq \max |\lambda_i|, \quad \|A\|_\infty \leq \max |\lambda_i|.$$  

**Proof.** Any vector $x \neq 0$ in $\mathbb{R}^N$ can be expressed in the basis of orthonormal eigenvectors of $A$, denote by $e_1, \ldots, e_N$

$$x = x_1 e_1 + \cdots + x_N e_N.$$  

Assuming that $e_i$ is the eigenvector of $A$ associated to the eigenvalue $\lambda_i$, we have

$$Ax = \lambda_1 x_1 e_1 + \cdots + \lambda_N x_N e_N.$$  

(2.10)

Hence for the 2-norm, using the orthonormality of the $e_i$

$$\|Ax\|_2^2 = \lambda_1^2 x_1^2 + \cdots + \lambda_N^2 x_N^2 \leq \max(\lambda_i^2) \|x\|^2_2.$$  

From which it follows that $\|A\|_2 \leq \sqrt{\max(\lambda_i^2)} = \max |\lambda_i|.$

For the max-norm, we get from (2.10) that

$$\|Ax\|_\infty = \max |\lambda_i x_i| \leq \max |\lambda_i| \|x\|_\infty,$$

from which the result follows.  

Hence if $A_h$ is an invertible symmetric matrix, it is diagonalisable in a basis of orthogonal eigenvectors and its eigenvalues are real and different from zero. Denoting by $P$ the matrix whose columns are the eigenvectors of $A_h$, we have $A_h = PA \Lambda P^T$, where $\Lambda$ is the diagonal matrix containing the eigenvalues. Then $A_h^{-1} = P \Lambda^{-1} P^T$, where $\Lambda^{-1}$ contains the inverse of the eigenvalues.

It follows that for the 2-norm and max-norm that we are interested in, we have

$$\|A_h^{-1}\| \leq \frac{1}{\min |\lambda_i|}.$$  

22
This leads us to the sufficient condition for stability that for all the eigenvalues \( \lambda_i \) of \( A_h \) we have

\[ |\lambda_i| \geq C > 0, \quad \text{for some constant } C \text{ independent of } h. \]

**Example:** Let us now check this for the Poisson problem with Dirichlet boundary conditions. In the continuous case for homogeneous Dirichlet boundary conditions the eigenvectors and eigenvalues of the Laplace operator \(-\frac{d^2}{dx^2}\) in 1D verify

\[ -\frac{d^2\phi_k}{dx^2} = \lambda_k \phi_k, \quad \phi(0) = \phi(L) = 0. \]

This is the equation of a harmonic oscillator for which the solutions read:

\[ \lambda_k = \frac{k^2 \pi^2}{L^2}, \quad \phi_k(x) = \sqrt{\frac{2}{L}} \sin \frac{k \pi x}{L}, \quad k \in \mathbb{Z}. \]

In the case of second order finite differences, the corresponding discrete eigenvalue problem reads \( A_h \Phi_k = \lambda_k \Phi_k \), with \( h = L/N \) and the \((N-1) \times (N-1)\) matrix from (2.7),

\[ A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & -1 & \ddots & \ddots & 2 \end{pmatrix}. \]

We can check that the components of the discrete eigenvectors (or eigenmodes), up to normalisation, are the values of the continuous eigenmodes at the grid points

\[ (\Phi_k)_j = \sqrt{\frac{2}{N}} \sin \frac{k j \pi}{N}, \quad 1 \leq k \leq N - 1, \]

and the corresponding eigenvalues are

\[ \lambda_k = \frac{4}{h^2} \sin^2 \frac{k \pi}{2N} = \frac{4}{h^2} \sin^2 \frac{kh \pi}{2L}. \]

As \( 0 < k < N \), we have \( 0 < k \pi/(2N) < \pi/2 \), so that the eigenvalues are positive and in increasing order as sinus is increasing on this interval. It follows that \( \lambda_1 \) is the smallest eigenvalue. Moreover, as \( h \to 0 \), we have

\[ \lambda_1 = \frac{4}{h^2} \sin^2 \frac{h \pi}{2L} \approx \frac{4}{h^2} \frac{\pi^2}{4L^2} \to \frac{\pi^2}{L^2}. \]

This corresponds to the continuous eigenvalue. Then because of the convergence property there exists \( h_0 = L/N_0 \) such that for all \( N \geq N_0 \) we have \( \lambda_1^N \geq (1/2)\pi^2/L^2 \). Thus for any \( N \geq N_0 \), we have

\[ \lambda_1^N \geq C = \min \left( \lambda_1^1, \ldots, \lambda_1^{N_0-1}, \frac{1}{2} \frac{\pi^2}{L^2} \right), \]

where \( C \) is a constant independent on \( N \) which proves the stability.
2.2.3 Finite difference methods in higher dimensions

In practical applications one is often interested in solving the Poisson equation (2.1) in a two-dimensional (2D) or 3D setting, hence with \( n = 2 \) or \( n = 3 \). The finite difference approach allows for the construction \( n \)-dimensional methods from the 1D method via tensor products. For simplicity, here we shall extend the finite difference method to a 2D Cartesian grid \((x_i, y_j)_{i,j=0}^N\), which is a tensor product of 1D grids (the 3D case can be treated analogously). We shall assume that the cell size is uniform and equal in the two directions for simplicity, \( h = h_x = h_y \), but this is not required. We will see that the notion of tensor product enables to construct the linear system directly from the 1D system, enabling easy implementation and also extension of the analysis from the 1D case.

For simplicity we will only consider homogeneous Dirichlet boundary conditions. Other boundary conditions can be adapted from the 1D case in the same manner. Denoting \((x, y) \in \Omega \subset \mathbb{R}^2\), the problem reads

\[-\Delta \phi(x, y) = \rho(x, y) \quad (x, y) \in \Omega, \quad \phi(x, y) = 0 \quad (x, y) \in \partial \Omega,\]

where \( \partial \Omega \) denotes the domain boundary and \( \rho \in C^0(\Omega) \) is given. The 2D Laplace operator is defined by

\[\Delta \phi(x, y) := \frac{\partial^2 \phi}{\partial x^2}(x, y) + \frac{\partial^2 \phi}{\partial y^2}(x, y).\]

Applying Taylor expansions as in (2.6) to each of the two 1D Laplacians yields

\[-\Delta \phi(x_i, y_j) = \frac{-\phi(x_{i+1}, y_j) + 2\phi(x_i, y_j) - \phi(x_{i-1}, y_j)}{h^2} + \frac{-\phi(x_i, y_{j+1}) + 2\phi(x_i, y_j) - \phi(x_i, y_{j-1})}{h^2} + O(h^2).\]

Let us consider the natural numbering of the grid values of the approximate solution \( \Phi_h \in \mathbb{R}^{N-1 \times N-1} \), which is now a matrix with entries \( \phi_{i,j} \approx \phi(x_i, y_j) \) for the interior grid points \( 1 \leq i, j \leq N - 1 \). Then, in the FD scheme there are \((N - 1)^2\) unknowns satisfying the \((N - 1)^2\) equations

\[-\phi_{i+1,j} - \phi_{i,j+1} + 4\phi_{i,j} - \phi_{i-1,j} - \phi_{i,j-1} = \frac{\rho(x_i, y_j)}{h^2}, \quad 1 \leq i, j \leq N - 1. \quad (2.11)\]

Introducing the right hand side matrix \( R_h = (\rho(x_i, y_j))_{1 \leq i,j \leq N-1} \) and recalling the 1D second-order Dirichlet matrix from (2.7),

\[A_h = \frac{1}{h^2} \begin{pmatrix}
2 & -1 & 0 \\
-1 & 2 & -1 \\
0 & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & \ddots & \ddots & \ddots & \ddots \\
-1 & 2 & -1 \\
0 & -1 & 2
\end{pmatrix},\]
we notice that the matrix multiplication \( A_h\Phi_h \) applies the 1D finite difference stencil to the columns of \( \Phi_h \) (with \( j \) fixed), which corresponds to the differentiation in \( x \). Similarly, the left multiplication of \( \Phi_h \) by \( A_h \), \( \Phi_h A_h = (A_h\Phi_h^\top)^\top \), applies the 1D finite difference stencil to the lines of \( \Phi \) (with \( i \) fixed) which corresponds to the differentiation in \( y \):

\[
A_h\Phi_h = \frac{1}{h^2} \begin{pmatrix}
2\phi_{1,1} - \phi_{2,1} & 2\phi_{1,2} - \phi_{2,2} & \cdots \\
-\phi_{1,1} + 2\phi_{2,1} - \phi_{3,1} & -\phi_{1,2} + 2\phi_{2,2} - \phi_{3,2} & \cdots \\
-\phi_{2,1} + 2\phi_{3,1} - \phi_{4,1} & -\phi_{2,2} + 2\phi_{3,2} - \phi_{4,2} & \cdots \\
\vdots & \vdots & \ddots
\end{pmatrix},
\]

\[
\Phi_h A_h = (A_h\Phi_h^\top)^\top = \frac{1}{h^2} \begin{pmatrix}
2\phi_{1,1} - \phi_{1,2} & -\phi_{1,1} + 2\phi_{1,2} - \phi_{1,3} & \cdots \\
2\phi_{2,1} - \phi_{2,2} & -\phi_{2,1} + 2\phi_{2,2} - \phi_{2,3} & \cdots \\
2\phi_{3,1} - \phi_{3,2} & -\phi_{3,1} + 2\phi_{3,2} - \phi_{3,3} & \cdots \\
\vdots & \vdots & \ddots
\end{pmatrix}.
\]

Then adding the two, yields at each matrix entry the 2D Laplacian stencil, so that the equations (2.11) can be written in matrix form

\[
\Phi_h A_h + A_h\Phi_h = R_h.
\]

Denoting by \( I_h \) the identity matrix of the same size as \( A_h \) and \( \Phi_h \) this reads equivalently

\[
I_h\Phi_h A_h + A_h\Phi_h I_h = R_h. \tag{2.12}
\]

In order to solve such a matrix system it needs to be brought into the standard matrix-vector multiplication form. The Kronecker product formalism does that for us. A detailed presentation can be found in the textbooks by Steeb [11, 12]. A nice review article on the properties and applications of the Kronecker product was written by Van Loan [16].

For our application, we first need to replace the matrix unknown \( \Phi_h \) by a column vector, which is called vec(\( \Phi_h \)) in the Kronecker product formalism and is obtained by stacking the columns of \( \Phi_h \) or equivalently numbering the grid points line by line:

\[
\text{vec}(\Phi_h) := (\phi_{1,1}, \phi_{2,1}, \ldots, \phi_{N-1,1}, \phi_{1,2}, \phi_{2,2}, \ldots, \phi_{N-1,2}, \phi_{3,1}, \ldots, \phi_{N-1,N-1})^\top.
\]

Moreover, for two matrices \( B \in \mathbb{R}^{m \times n} \) and \( C \in \mathbb{R}^{p \times q} \) their Kronecker product is defined by

\[
B \otimes C = (b_{m,n} \cdot C) := \begin{pmatrix}
b_{11}C & \cdots & b_{1n}C \\
\vdots & \ddots & \vdots \\
b_{m1}C & \cdots & b_{mn}C
\end{pmatrix} \in \mathbb{R}^{mp \times nq}.
\]

Then, for \( X \in \mathbb{R}^{q \times n} \) and \( Y \in \mathbb{R}^{p \times m} \) we have the important equivalence

\[
CXB^\top = Y \iff (B \otimes C) \text{vec}(X) = \text{vec}(Y).
\]
This is all we need to rewrite our 2D discrete Poisson equation using Kronecker products. As $A_h$ is symmetric, (2.12) is equivalent to

$$(A_h \otimes I_h + I_h \otimes A_h)\text{vec}(\Phi_h) = \text{vec}(R_h).$$

As the Kronecker product is available in numerical computing languages like Matlab or Python (numpy), this can be used directly to assemble the linear system in 2D, which means that only the 1D Finite Difference matrices need to be assembled explicitly.

As the eigenvalues of the Kronecker product of two square matrices is the product of the eigenvalues of each matrix, the stability of the 2D problem can also be studied using the eigenvalues of the 1D problems.

The tensor product ideas generalises to arbitrary dimensions and has the property of separating an $n$D problem into a sequence of 1D problem enabling to obtain some very fast algorithms.

### 2.3 The method of manufactured solutions

A simple and standard way of checking the correctness of the code implementing a numerical method is to use a known analytical solution. This can be done by using a known solution in a specific case or also by picking a solution and constructing the problem around it. This is called the method of manufactured solutions.

For example, for periodic boundary conditions one can pick any periodic function $\phi$ and apply the operator to it, in our case the Laplacian, to find the corresponding right hand side $\rho$, and then solve the problem with this $\rho$ for different mesh sizes and check the convergence order in some given norm.

For non periodic boundary conditions, one can pick a function satisfying homogeneous Dirichlet or Neumann boundary conditions or on can pick any smooth function and determine the boundary conditions according to the function we chose. In any case it is important not to forget the boundary conditions when defining the problem.

### 2.4 The Finite Element Method

#### 2.4.1 Principle of the method

For solving a problem on a computer that can only store a finite amount of information, a discrete form of the problem is needed. In the Finite Difference method one simply computes an approximation of the solution at a finite number of grid points. In the Finite Element method, which is mathematically more involved, the idea is to look for the solution in a finite dimensional vector space, i.e. for some well chosen vector space $V_h$, with basis $(\varphi_i)_{0 \leq i \leq N-1}$, the approximate solution has the form

$$u_h(x) = \sum_{i=0}^{N-1} u_i \varphi_i(x).$$
The basis being given, the approximate solution $u_h$ is fully determined by its coefficients $u_i$ in this basis, which need not be values of $u_h$ at some points in the computational domain, but can be in some cases.

The question now becomes how to choose $V_h$ and determine the coefficients $u_i$ such that $u_h$ is a good approximation of the solution $u$ of the original problem, that we take as a start as the Poisson problem with homogeneous Dirichlet boundary conditions:

$$-\Delta u(x) = f(x), \quad x \in \Omega \subset \mathbb{R}^d, \quad u = 0, \quad \text{on } \partial \Omega.$$ 

The first idea, introduced by Ritz in his thesis in Göttingen in 1902, was to transform the boundary problem into an equivalent minimisation problem. Suppose we search for a solution $u \in V$ is some Hilbert space $V$. We then write the \textbf{weak formulation} of the problem by multiplying the equation with a test function $v \in V$ and integrating over the domain. Integration by parts and application of the boundary condition gives

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} fv \, dx, \quad \forall \ v \in V. \quad (2.13)$$

Remark that we search for $u \in V$ such that this relation holds for all $v \in V$. The space $V$ is chosen such that all of the above integrals make sense. We shall need in particular the following Hilbert spaces:

$$H^1(\Omega) = \{ u \in L^2(\Omega), \nabla u \in (L^2(\Omega))^d \},$$

$$H^1_0(\Omega) = \{ u \in H^1(\Omega), u = 0 \text{ on } \partial \Omega \}.$$ 

The scalar product associated to these Hilbert spaces is

$$(u, v)_{H^1} = \int_{\Omega} \nabla u \cdot \nabla v \, dx + \int_{\Omega} uv \, dx.$$ 

Thus setting $V = H^1_0$ would be a possible choice for our problem. It turns out that the weak problem can be formulated as a \textbf{variational problem} (minimisation problem):

\textbf{Theorem 1.} The problem

$$\begin{align*}
find \ u \in V : \\
\begin{cases}
J(u) = \min_{v \in V} J(v), \\
J : V \to \mathbb{R}, \quad J(v) = \frac{1}{2} \int_{\Omega} |\nabla v|^2 \, dx - \int_{\Omega} fv \, dx,
\end{cases}
\end{align*} \quad (2.14)$$

is equivalent to the problem (2.13) in the sense that $u$ is a solution of (2.13) if and only if it is a solution of (2.14).

Proof. Suppose first that $u \in V$ is a solution of the variational problem (2.14), then

$$J(u) \leq J(u + \varepsilon v) \quad \forall \ v \in V.$$
Therefore, the function \( \psi(\varepsilon) = J(u + \varepsilon v) \) has a minimum at \( \varepsilon = 0 \), hence
\[
\frac{d\psi}{d\varepsilon}|_{\varepsilon=0} = \lim_{\varepsilon \to 0} \frac{J(u + \varepsilon v) - J(u)}{\varepsilon} = \frac{dJ(u + \varepsilon v)}{d\varepsilon}|_{\varepsilon=0} = 0.
\]
This corresponds to the definition of the Gâteaux derivative of the functional \( J \) at \( u \),
\[
J'[u](v) := \lim_{\varepsilon \to 0} \frac{J(u + \varepsilon v) - J(u)}{\varepsilon} = \frac{dJ(u + \varepsilon v)}{d\varepsilon}|_{\varepsilon=0}.
\] (2.15)
For \( J \) give in (2.14) we have
\[
J(u + \varepsilon v) = \frac{1}{2} \int_{\Omega} |\nabla u + \varepsilon \nabla v|^2 \, dx - \int_{\Omega} f(u + \varepsilon v) \, dx
\]
\[
= \frac{1}{2} \left( \int_{\Omega} |\nabla u|^2 \, dx + 2\varepsilon \int_{\Omega} \nabla u \cdot \nabla v \, dx + \varepsilon^2 \int_{\Omega} |\nabla v|^2 \, dx \right) - \int_{\Omega} f u \, dx - \varepsilon \int_{\Omega} f v \, dx.
\] (2.16)
From this we deduce, using the Gâteaux formula (2.15), that
\[
J'[u](v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx - \int_{\Omega} f v \, dx.
\]
Note that \( J'[u] \) being a linear form on \( V \) is defined by applying it to some vector \( v \in V \). Finally the solution of our minimisation problem is a solution of \( J'[u] = 0 \) or equivalently \( J'[u](v) = 0 \) for all \( v \in V \), which reads
\[
\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx \quad \forall \ v \in V.
\]
Conversely, suppose \( u \in V \) solves (2.13). Then for arbitrary \( v \in V \), from (2.16) we deduce
\[
J(u + \varepsilon v) = J(u) + \varepsilon \left[ \int_{\Omega} \nabla u \cdot \nabla v \, dx - \int_{\Omega} f v \, dx \right] + \frac{\varepsilon^2}{2} \int_{\Omega} |\nabla v|^2 \, dx.
\]
As \( |\nabla v|^2 \geq 0 \) we have
\[
J(u + \varepsilon v) \geq J(u) \quad \forall \ v \in V.
\]
Because of the original problem being transformed into a minimisation problem, it becomes quite natural to look for an approximation in a finite dimensional subspace \( V_h \subset V \) of the function space in which the minimisation problem is posed (in our case \( V = H^1_0(\Omega) \)). This means means that the minimisation is performed by considering only minima in a finite dimensional subspace \( V_h \). Then, if the form of the finite dimensional space is chosen such that any function of
the original space can be approximated to any given tolerance, by a function of
the approximation space, we should be able to get a good approximation
The approximate weak formulation thus reads: find $u_h \in V_h \subset V$ such that
$$\int_{\Omega} \nabla u_h \cdot \nabla v_h \, dx = \int_{\Omega} f v_h \, dx \quad \forall \, v_h \in V_h. \quad (2.17)$$
Note that the weak formulation (2.13) expresses in some sense the orthogonality
of the residual $u - u_h$ to the space in which the solution is sought (here $V_h$), since
$$\int_{\Omega} \nabla (u - u_h) \cdot \nabla v_h \, dx = 0 \quad \forall v_h \in V_h.$$  
This is more general than Euler’s equation $J' = 0$ of minimisation problems as
noticed by Galerkin and has a wide range of applications. One can even extend
this concept by making the residual orthogonal to a different function space, than
the one in which the solution lives. Such methods are called Petrov-Galerkin
methods and are beyond the scope of this lecture.

So the principle of the **Galerkin Finite Element method** is to look for a
solution $u_h \in V_h$ in a finite dimensional subspace $V_h \subset V$ of the original space
and to use the same variational formulation (2.13) defining the exact solution, but
with test functions in $V_h$. What remains to be done now is to choose $V_h$ with
good approximation properties. As we will see later, the stability of the Galerkin
method follows directly from the well-posedness of the variational problem (2.13).

The space $V_h$ is characterised by a basis $(\varphi_0, \ldots, \varphi_{N-1})$ so that finding the
solution of the weak problem (2.17) amounts to solving a linear system. Indeed,
express the trial function $u_h$ and the test function $v_h$ on this basis:
$$u_h(x) = \sum_{j=0}^{N-1} u_j \varphi_j(x), \quad v_h(x) = \sum_{i=0}^{N-1} v_i \varphi_i(x),$$
and substitute these expressions in (2.17). This yields
$$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} u_j v_i \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, dx = \sum_{i=0}^{N-1} v_i \int_{\Omega} f \varphi_i \, dx.$$  
This can be expressed in matrix form,
$$v_h^T A_h u_h = v_h^T b_h, \quad \forall \, v_h \in \mathbb{R}^N,$$
where
$$u_h = (u_0, \ldots, u_{N-1})^T \in \mathbb{R}^N,$$
$$v_h = (v_0, \ldots, v_{N-1})^T \in \mathbb{R}^N,$$
$$b_h = \left( \int_{\Omega} f \varphi_0 \, dx, \ldots, \int_{\Omega} f \varphi_{N-1} \, dx \right)^T \in \mathbb{R}^N.$$  

$^1$Ritz who was actually looking at solutions for the bilaplacian equation, chose as basis functions for $V_h$ a finite number of eigenfunctions of his operator.
and the matrix $A_h \in \mathbb{R}^{N \times N}$ whose entries are

$$(A_h)_{i,j} = \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, dx,$$

is called the stiffness matrix. This is equivalent to the linear system $A_h u_h = b_h$ as the previous equality is true for all $v_h \in \mathbb{R}^N$.

The finite dimensional space $V_h$ is in general defined by its basis functions. For those, Ritz used eigenfunctions of the problem. But those are in general cumbersome to compute. Galerkin proposed to use general classes of simple functions, trigonometric functions or polynomials, that are know to be able to approximate any continuous function with a finite number of basis functions. Trigonometric polynomials which are linked to Fourier series are very good in periodic domains, with a few simple extensions. Polynomials enjoy more widespread applications, however to get a good conditioning of the linear system that is obtained at the end, care needs to be taken in the choice of the basis functions. The monomial basis $(1, x, x^2, \ldots)$ has very bad properties. Best approximations are provided by the orthogonal Legendre polynomials or by the Chebychev polynomials which are used in practice. Note that all the basis functions we have mentioned up to now have a global support in the computational domain and thus lead to full matrices in the linear system, which can be computationally expensive. Methods using such bases are actually not known as Finite Element methods but rather as spectral methods. We will come back to those later.

Another ingredient is needed to define what is known as Finite Element methods. This was introduced by Courant in 1943 and consists in using basis functions with a small support in the computational domain, so that its product with other basis functions vanishes for most of the other basis functions leading to a very sparse matrix in the linear system, which can be solved very efficiently on a computer. For this the computational domain is decomposed into small elements, in general triangles or quads in 2D and the basis functions are chosen to be relatively low order polynomials, on each of these elements. Convergence is achieved by taking smaller elements like the cells in the Finite Difference method. In 1D a finite element mesh will look like a finite difference mesh. Some examples of unstructured Finite Element meshes are displayed in Figure 2.1, which shows the great flexibility in particular to handle complicated boundaries with finite elements, which finite differences do not provide. This is a key to its very wide usage.

The article by Gander and Wanner [5] provides a clear and well documented overview of the historical developments of the Finite Element method. For more technical historical developments of the Finite Difference and Finite Element methods one can also consult [13].

2.4.2 The weak form of a boundary value problem

The weak form of a boundary value problem contains all its elements, which are the partial differential equation in the interior of the domain and the boundary
conditions. There are two very distinct ways to handle the boundary conditions depending on how they appear when deriving the variational formulation:

- If they appear on the test function they are called essential boundary conditions and need to be included in the space where the solution is looked for.
- If they appear on the trial function, which will be the approximate solution, they can be handled in a natural way in the weak formulation. Such boundary conditions are called natural boundary conditions.

We will see on the examples of Dirichlet and Neumann boundary conditions how this works in practice. In order to define the weak formulation, we will need the following Green formula: For $u \in H^2(\Omega)$ and $v \in H^1(\Omega)$

$$- \int_\Omega \Delta u v \, dx = \int_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial \Omega} \frac{\partial u}{\partial n} v \, d\sigma.$$  \hspace{1cm} (2.18)

Here $H^2(\Omega)$ denotes the Hilbert space of the functions whose partial derivatives up to second order are in $L^2(\Omega)$ and $\frac{\partial u}{\partial n} = \nabla u \cdot n$, where $n$ is the outbound normal at any point of the boundary.

**Case of Neumann boundary conditions**

Let $f \in L^2(\Omega)$ and $g \in H^1(\partial \Omega)$. We consider the problem

$$\left\{ \begin{array}{l}
-\Delta u + u = f \quad \text{in } \Omega, \\
\frac{\partial u}{\partial n} = g \quad \text{on } \partial \Omega.
\end{array} \right.$$  \hspace{1cm} (S)

Assuming that $u \in H^2(\Omega)$, we multiply by a test function $v \in H^1(\Omega)$ and integrate using the Green formula (2.18), which yields

$$\int_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial \Omega} \frac{\partial u}{\partial n} v \, d\sigma + \int_\Omega uv \, dx = \int_\Omega f v \, dx.$$
Replacing \( \frac{\partial u}{\partial n} \) by its value \( g \) on the boundary, we obtain the weak formulation

\[
(W) \quad \text{Find } u \in H^1(\Omega) \text{ such that }
\int_{\Omega} \nabla u \cdot \nabla v \, dx + \int_{\Omega} uv \, dx = \int_{\Omega} f v \, dx + \int_{\partial\Omega} g v \, d\sigma \quad \forall \ v \in H^1(\Omega).
\] (2.19)

Note that the boundary condition does not appear in the function space \( H^1(\Omega) \), but rather in the equation - hence this is an example of a natural boundary condition. The solutions of \((W)\) are called weak solutions of the original boundary value problem. The solutions which are also in \( H^2(\Omega) \) are called strong solutions. Indeed we can prove the following:

**Proposition 2.** Provided \( u \in H^2(\Omega) \), the problems \((S)\) and \((W)\) are equivalent.

**Proof.** We have already shown in the derivation of the weak formulation that if \( u \) is a solution of \((S)\), it also satisfies \((W)\). On the other hand, suppose \( u \in H^2(\Omega) \subset H^1(\Omega) \) is a solution of \((W)\). As \( H^1_0(\Omega) \subset H^1(\Omega) \) one can first take only test functions \( v \in H^1_0(\Omega) \). Then as in the case of homogeneous Dirichlet conditions it follows from the Green formula (2.18) that

\[
\int_{\Omega} (-\Delta u + u) v \, dx = \int_{\Omega} f v \, dx \quad \forall \ v \in H^1_0(\Omega).
\]

This implies, as \( H^1_0(\Omega) \) is dense in \( L^2(\Omega) \), that \( -\Delta u + u = f \) in \( L^2(\Omega) \) and so almost everywhere. It now remains to verify that we have the boundary condition

\[
\frac{\partial u}{\partial n} = g \quad \text{on } \partial \Omega.
\]

For that we start from (2.19) and apply the Green formula (2.18), which yields

\[
-\int_{\Omega} \Delta u \, v \, dx + \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\sigma + \int_{\Omega} uv \, dx = \int_{\Omega} f v \, dx + \int_{\partial\Omega} g v \, d\sigma \quad \forall \ v \in H^1(\Omega),
\]

and as \( -\Delta u + u = f \), it remains

\[
\int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\sigma = \int_{\partial\Omega} g v \, d\sigma \quad \forall \ v \in H^1(\Omega),
\]

which yields that \( \frac{\partial u}{\partial n} = g \) on \( \partial \Omega \). \( \square \)

**Homogeneous Dirichlet boundary conditions**

Let \( f \in L^2(\Omega) \) and consider the boundary value problem

\[
(S) \quad \begin{cases}
-\Delta u = f & \text{in } \Omega, \\
u = 0 & \text{on } \partial \Omega.
\end{cases}
\]
Assume that $u \in H^2(\Omega)$, we multiply by $v \in H^1(\Omega)$ and integrate using the Green formula (2.18), which yields
\[
\int_{\Omega} \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\sigma = \int_{\Omega} f v \, dx.
\]
Here, only the derivative of $u$ appears in the boundary integral, so we cannot apply the boundary condition directly. But in the end $u$ will be in the same function space as the test function $v$, which appears directly in the boundary integral. Thus we take test functions $v$ vanishing on the boundary and get the following weak formulation:

$$
(W) \begin{cases}
\text{Find } u \in H^1_0(\Omega) \text{ such that } \\
\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx \quad \forall \; v \in H^1_0(\Omega).
\end{cases}
$$

As the boundary condition appears in the function space $H^1_0(\Omega)$, this is the case of an \textbf{essential boundary condition}. Again we can show the equivalence of $(S)$ and $(W)$ if $u \in H^2(\Omega)$. In this case the Green formula (2.18) can be used, and as $v$ vanishes on the boundary it yields
\[
-\int_{\Omega} \Delta u \, v \, dx = \int_{\Omega} f v \, dx \quad \forall v \in H^1_0(\Omega).
\]
This implies, as $H^1_0(\Omega)$ is dense in $L^2(\Omega)$, that $-\Delta u = f$ in $L^2(\Omega)$ and so almost everywhere. On the other hand as $u \in H^1_0(\Omega)$, $u = 0$ on $\partial \Omega$. So $u$ is a strong solution.

**Non-homogeneous Dirichlet boundary conditions (lifting)**

Let $f \in L^2(\Omega)$ and $u_0 \in H^1(\Omega)$. We consider the problem
\[
-\Delta u = f \quad \text{in } \Omega, \\
u = u_0 \quad \text{on } \partial \Omega.
\]
As the value of $u$ on the boundary cannot be directly put in the function space if it is not zero, as else the function space would not be stable by linear combinations, we need to bring the problem back to the homogeneous case. To this aim let $\tilde{u} = u - u_0$. The function $u_0$ is called lifting of the boundary data. We then show as previously that $\tilde{u}$ is a solution of the weak problem

$$
(W) \begin{cases}
\text{Find } \tilde{u} \in H^1_0(\Omega) \text{ such that } \\
\int_{\Omega} \nabla \tilde{u} \cdot \nabla v \, dx = \int_{\Omega} f v - \int_{\Omega} \nabla u_0 \cdot \nabla v \, dx \quad \forall \; v \in H^1_0(\Omega).
\end{cases}
$$

This is the weak problem that needs to be solved for non-homogeneous Dirichlet boundary conditions. As $u_0$ will only have non zero entries on the boundary for standard Finite Elements, the problem can be simplified in different manners in practice.
2.4.3 Lagrange Finite Elements

Finite Elements are used to construct a finite dimensional space $V_h$ with basis functions that have a small support so that the resulting matrix is sparse, i.e. most of its entries vanish. The simplest Finite Elements are determined by points values and related to Lagrange interpolation, hence their name. In order to construct a basis of $V_h$, one starts by decomposing the computational domain into non overlapping intervals in 1D (like the Finite Difference mesh), triangles or quads in 2D, tetrahedra or hexahedra in 3D. This defines a mesh of the computational domain. Then the basis functions are defined locally on each cell (or element).

We also want $V_h \subset V$ to be a subspace of the solution space of the continuous problem, which is usually a Sobolev space $H^k$ with $k \geq 0$. The following property speaks to the continuity of functions in $H^k$:

**Lemma 1.** Let $\Omega \subset \mathbb{R}^n$ open, $n \geq 1$, with sufficiently regular boundary, then

$$H^k(\Omega) \subset C^m(\Omega) \quad \text{if} \quad k > m + \frac{n}{2}.$$  

This means in particular that for $n = 1$ functions of $H^1$ are continuous on $\Omega$, whereas this is not necessarily the case in two or three dimensions. However, functions in $H^2$ are indeed continuous also for $n = 2$ and $n = 3$.

Let us start with a non-uniform 1D mesh of the domain $\Omega = [a,b]$ defined by the grid points $a = x_0 < x_1 < \cdots < x_N = b$ and the solution space being $V = H^1(\Omega)$. The elements of the mesh are here the intervals $\Omega_\nu := [x_\nu, x_{\nu+1}]$ such that $\bigcup \nu \nu \nu = \Omega$. The important point is that functions $v_h \in V_h \subset V$ will be constructed as piece-wise polynomials on the elements $\Omega_\nu$. Denoting the space of polynomials of degree $k$ on $\Omega_\nu$ by $P_k(\Omega_\nu)$, a space $V_h \subset H^1(\Omega)$ can be defined as

$$V_h := \left\{ v_h \in C^0([a,b]) : v_h|_{\Omega_\nu} \in P_k(\Omega_\nu) \right\}.$$  

Because of Lemma 1 we need to choose $V_h$ as a subset of $C^0([a,b])$. In order to make this requirement easy to implement in practice it is convenient to define the basis of $P_k(\Omega_\nu)$ as being defined by its value at $k+1$ points in $\Omega_\nu$, **including the endpoints of the interval** $x_\nu$ and $x_{\nu+1}$. Such a basis is called a nodal basis and the naturally associated basis functions are the Lagrange basis functions.

The basis restricted to each element is defined via a reference element, which is conveniently chosen as the interval $[-1,1]$ and an affine map

$$F_\nu : [-1,1] \to [x_\nu, x_{\nu+1}]$$

$$\hat{x} \mapsto x_\nu + \frac{x_{\nu+1} - x_\nu}{2} \hat{x}.$$  

As an affine mapping maps polynomials of degree $k$ to polynomials of degree $k$, the basis can be defined on the reference element $[-1,1]$. Given $k+1$ interpolation points $-1 = \hat{x}_0 < \hat{x}_1 < \cdots < \hat{x}_k = 1$ the Lagrange shape functions (i.e. the basis
functions on the reference element) of degree $k$ denoted by $l_{k,i}$, $0 \leq i \leq k$, are the unique polynomials of degree $k$ verifying
\[ l_{k,i}(\hat{x}_j) = \delta_{i,j} \quad 0 \leq j \leq k. \tag{2.20} \]
Because of this property, any polynomial $p \in \mathbb{P}_k([-1,1])$ can be expressed as
\[ p(\hat{x}) = \sum_{j=0}^{k} p(\hat{x}_j) l_{k,j}(\hat{x}). \]
Therefore, any polynomial $p \in \mathbb{P}_k([-1,1])$ is uniquely determined by its values at the interpolation points $\hat{x}_j$, $0 \leq j \leq k$. In order to ensure the continuity of the piecewise polynomial at the cell interfaces $x_\nu$ it is enough that the values of the polynomials on both sides of $x_\nu$ have the same value at $x_\nu$. This constraint removes one degree of freedom in each cell, moreover for Dirichlet boundary conditions the two end points are known, which removes two other degrees of freedom so that the total dimension of $V_h$ in this case is
\[ \dim V_h = N(k+1) - (N-1) - 2 = Nk - 1 \quad \text{(Dirichlet)}. \]
The functions of $V_h$ are uniquely defined in each cell by their value at the degrees of freedom (which are the interpolation points) in all the cells. The basis functions of $V_h$ are denoted by $(\varphi_i)_{1 \leq j \leq Nk-1}$ and are computed from (2.20) plus the continuity requirements. Examples of Lagrange shape functions are plotted in Figure 2.4. Note that for $k = 1$, corresponding to $\mathbb{P}_1$ finite elements, the degrees of freedom are just the grid points. For higher order finite elements internal degrees of freedom are needed.
2.4.4 B-spline Finite Elements

B-splines or basis splines are piece-wise polynomial functions of minimal support for given degree and smoothness. They are often used in the context of computer-aided design (CAD) and numerical solution of PDEs, in particular with FEM.

In order to define a family of \( n \) B-spline functions of degree \( k \), we need \( T = \{(x_i)_{0 \leq i \leq n+k}\} \) a non-decreasing sequence of points on the real line called knots in the spline terminology. There can be several knots at the same position. In the case when there are \( m \) knots at the same point, we say that the knot has multiplicity \( m \).

**Definition 2 (B-Spline).** Let \((x_i)_{0 \leq i \leq n+k}\) be a non-decreasing sequence of knots. Then the \( j \)-th B-Spline \((0 \leq j \leq n-1)\) denoted by \( N_j^k \) of degree \( k \) is defined by the recurrence relation:

\[
N_0^0(x) = \chi_{[x_i, x_{i+1}]}(x),
\]

\[
k \geq 1 : \quad N_j^k(x) = w_j^k(x) N_{j-1}^{k-1}(x) + (1 - w_{j+1}^k(x)) N_{j+1}^{k-1}(x),
\]

where

\[
w_j^k(x) = \frac{x - x_j}{x_{j+k} - x_j},
\]

and \( \chi_{[x_i, x_{i+1}]}(x) \) denotes the indicator function on the interval \([x_i, x_{i+1}]\).

![B-splines of different degree](image)

**Figure 2.3:** B-spline functions of different degree \( k \), each obtained from the knot sequence \( \{0, \ldots, k+1\} \).

We note some important properties of a B-splines basis:

- B-splines are piecewise polynomial of degree \( k \),
- B-splines are non negative
• Compact support; the support of $N_j^k$ is contained in $[x_j, \ldots, x_{j+k+1}]$

• Partition of unity: $\sum_{i=0}^{n-1} N_i^k(x) = 1, \forall x \in \mathbb{R}$

• Local linear independence

• If a knot $x_i$ has a multiplicity $m$ then the B-spline is $C^{(k-m)}$ at $x_i$.

A key point for constructing discrete Finite Element spaces for the Maxwell equation comes from the recursion formula for the derivatives:

$$N_i^{k'}(x) = k \left( \frac{N_i^{k-1}(x)}{x_{i+k} - x_i} - \frac{N_{i+1}^{k-1}(x)}{x_{i+k+1} - x_{i+1}} \right).$$

It will be convenient to introduce the notation $D_i^k(x) = k \frac{N_i^{k-1}(x)}{x_{i+k} - x_i}$, sometimes called $D$-splines. Then the recursion formula for derivative simply becomes

$$N_i^{k'}(x) = D_i^k(x) - D_{i+1}^k(x).$$

Let us consider a uniform mesh of the interval $\Omega = [a,b]$ defined by the grid points $x_i = a + ih$, $h = \frac{(b-a)}{n}$, $i = 0, \ldots, n$. The knot vector for a B-spline basis of degree $k$ with maximal smoothness reads as follows:

periodic b.c.: $T_{\sharp} = \{x_0, x_0 - h, \ldots, x_0 - kh, x_0, \ldots, x_n, x_n + h, \ldots, x_n + kh\}$,

Dirichlet b.c.: $T_0 = \{x_0, \ldots, x_0, x_0, \ldots, x_n, x_n, \ldots, x_n\}$.

In both cases the knot vector consists of $n + 1 + 2k$ elements. In the periodic case, every knot has multiplicity 1, and $k$ knots are added to the left and to the right to implement periodicity in a natural way. When accounting for periodicity, there are $n$ B-spline functions created from $T_\sharp$. In the Dirichlet case, every knot has multiplicity 1 except for the boundary knots, which appear $k + 1$ times so that the first and last B-spline become interpolatory on the boundary. There are $n + k$ B-spline functions created from $T_0$. In either case, the set $(N_i^k)$ of B-splines of degree $k$ forms a basis of the spline space defined by

$$S^k = \{v \in C^{k-1}([x_0, x_n]) \mid v|_{[x_i, x_{i+1}]} \in P_k([x_i, x_{i+1}])\}.$$ 

It follows immediately that $(D_i^k)_i$ is a basis of $S^{k-1}$. Note that if the first knot has multiplicity $k + 1$, $D_0^k$ will have a support restricted to one point and be identically 0.

### 2.4.5 Convergence of the Finite Element method

The variational (or weak) formulation of PDEs we consider can be written in the following abstract form:

Find $u \in V$ such that $a(u, v) = l(v) \quad \forall \; v \in V$, \hspace{1cm} (2.21)
Figure 2.4: Piece-wise polynomials of B-splines of different degree in a single cell. Bottom right: the left boundary cell for Dirichlet conditions, where the blue spline is interpolatory.

where $V$ is a Hilbert space, $a : V \times V \to \mathbb{R}$ is a symmetric continuous and coercive bilinear form and $l : V \to \mathbb{R}$ a continuous linear form.

The most convenient tool for proving existence and uniqueness of the solution of a variational problem is the Lax-Milgram theorem that we recall here:

**Theorem 2** (Lax-Milgram). Let $V$ a Hilbert space with the norm $\| \cdot \|_V$. Let $a : V \times V \to \mathbb{R}$ denote a continuous, symmetric and coercive bilinear form on $V \times V$, i.e.

1. (Continuity): there exists $K$ such that for all $u, v \in V$

   $$|a(u, v)| \leq K\|u\|_V\|v\|_V.$$

2. (Coercivity): there exists a constant $\alpha > 0$ such that for all $u \in V$

   $$a(u, u) > \alpha\|u\|_V^2.$$

Moreover, let $l : V \to \mathbb{R}$ be a continuous linear form on $V$, i.e. there exists $C$ such that for all $v \in V$

$$|l(v)| \leq C\|v\|_V.$$

Then there exists a unique $u \in V$ such that

$$a(u, v) = l(v) \quad \forall \ v \in V.$$  \hspace{1cm} (2.22)
The Ritz-Galerkin method consists in finding an approximate solution \( u_h \) in a finite dimensional subspace of \( V \). For convergence studies one needs to consider a sequence of subspaces of \( V \) of larger and larger dimension so that they get closer to \( V \). One then defines a sequence of problems parametrised by \( h \) that read:

\[
\text{Find } u_h \in V_h \text{ such that } a(u_h, v_h) = l(v_h) \quad \forall \ v_h \in V_h,
\]

(2.23)

where \( V_h \subset V \) is a vector space of dimension \( N \).

Let \( B = (\varphi_1, \ldots, \varphi_N) \) be a basis of \( V_h \). An element \( u_h \in V_h \) can then be expanded as \( u_h(x) = \sum_{j=1}^N u_j \varphi_j(x) \). Taking \( v_h = \varphi_i \) and using the linearity of \( a \) the equation (2.23) becomes

\[
\sum_{j=1}^N u_j a(\varphi_j, \varphi_i) = l(\varphi_i) \quad \forall \ \varphi_i \in B.
\]

Then using the symmetry of \( a \), we notice that the discrete variational formulation (2.23) is equivalent to the linear system

\[
AU_h = L,
\]

(2.24)

where \( A = (a(\varphi_i, \varphi_j))_{1 \leq i,j \leq N} \in \mathbb{R}^{N \times N} \), \( L \in \mathbb{R}^N \) is the column vector with components \( l(\varphi_i) \) and \( U_h \in \mathbb{R}^N \) is the column vector with the unknowns \( u_i \) that are the coefficients of \( u_h \) in the basis \( B \).

**Theorem 3.** Assume that \( a : V \times V \to \mathbb{R} \) is a symmetric continuous and coercive bilinear form on a Hilbert space \( V \) and \( l : V \to \mathbb{R} \) is a continuous linear form on \( V \). Then the system (2.24) is equivalent to the discrete variational form (2.23) and admits a unique solution.

**Proof.** For \( v_h \in V_h \), we denote by \( \tilde{V} \) the vector of its components in the basis \( B \).

- Thanks to the bilinearity of \( a \) and the linearity of \( l \) the relation (2.23) can be written equivalently

\[
\tilde{V}^T A U_h = \tilde{V}^T L \quad \forall \ \tilde{V} \in \mathbb{R}^N,
\]

(2.25)

which means that the vector \( AU_h - L \in \mathbb{R}^N \) is orthogonal to all the vectors of \( \mathbb{R}^N \), and so is the zero vector. Conversely it is clear that (2.24) implies (2.25) and so (2.23).

- Let \( v_h \in V_h \). Then, as \( a \) is coercive, there exists \( \alpha > 0 \) such that

\[
\tilde{V}^T A \tilde{V} = a(v_h, v_h) \geq \alpha \| v_h \|^2 \geq 0,
\]

and \( \tilde{V}^T A \tilde{V} = 0 = a(v_h, v_h) \Rightarrow \| v_h \| = 0 \), which implies that \( v_h = 0 \) and so \( \tilde{V} = 0 \). So \( A \) is symmetric, positive definite and therefore invertible. \( \square \)
After making sure the approximate solution exists for some given space \( V_h \), one needs to make sure the approximation converges towards the exact solution. This results from two properties: 1) The Galerkin orthogonality, which comes from the conforming Galerkin approximation, 2) The approximability property, which confirms that for any \( v \in V \) there exist \( v_h \) in some finite dimensional space of the family which is close enough to \( v \).

**Lemma 2 (Céa).** Let \( u \in V \) be the solution of (2.22) and \( u_h \in V_h \) the solution of (2.23), with \( V_h \subset V \). Then

\[
\|u - u_h\| \leq \frac{K}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|.
\]

**Proof.** We have

\[
a(u, v) = l(v) \quad \forall \ v \in V,
\]
\[
a(u_h, v_h) = l(v_h) \quad \forall \ v_h \in V_h,
\]
as \( V_h \subset V \), we can take \( v = v_h \) in the first equality and take the difference which yields

\[
a(u - u_h, v_h) = 0 \quad \forall \ v_h \in V_h.
\]

This is the Galerkin orthogonality, which says that \( u_h \) is the orthogonal projection of \( u \) into \( V_h \). It results that \( a(u - u_h, u - u_h) = a(u - u_h, u - v_h + v_h - u_h) = a(u - u_h, u - v_h), \) as \( v_h - u_h \in V_h \) and so \( a(u - u_h, v_h - u_h) = 0 \). Then there exists \( \alpha > 0 \) and \( \beta \) such that

\[
\alpha \|u - u_h\|^2 \leq a(u - u_h, u - u_h) \quad \text{as } a \text{ is coercive,}
\]
\[
= a(u - u_h, u - v_h) \quad \forall \ v_h \in V_h,
\]
\[
\leq K \|u - u_h\| \|u - v_h\| \quad \text{as } a \text{ is continuous.}
\]

Whence \( \|u - u_h\| \leq \frac{K}{\alpha} \|u - v_h\| \) for all \( v_h \in V_h \). We get the desired results taking the infimum in \( V_h \).

Let us now come to the approximability property. For simplicity we consider the 1D case and assume \( v \in C^0(I) \), with \( I = (a, b) \subset \mathbb{R} \) some interval. Suppose we are given nodes \( a = x_0 < x_1 < \ldots x_N = b \) that are used to define a basis \( B = \{\varphi_0, \ldots, \varphi_N\} \) of \( V_h \subset C^0(I) \). We then define the interpolation operator \( \pi_h : C^0(I) \to V_h \) by

\[
\pi_h v(x_i) = v(x_i) \quad \text{for all nodes } x_i \text{ of the partition } x_0, \ldots, x_N.
\]

The function \( \pi_h v \in V_h \) is called the interpolant of \( v \in C^0(I) \). In case that \( B \) consists of \( k \)-th order Lagrange polynomials \( l_{k,i} \) the interpolant is just

\[
\pi_h v(x) = \sum_{i=0}^{N} v(x_i) l_{k,i}(x).
\]
Theorem 4. Let \( v \in H^{k+1}(I) \subset C^0(I) \) (c.f. Lemma 1), for \( k \geq 1 \) and let \( \pi_h v \in V_h \) stand for its interpolant defined by (2.26), where \( V_h = \text{span}(B_k) \) with the basis \( B_k \) being composed of piece-wise polynomials of degree \( k \). Denote by \( h = \min_i |x_{i+1} - x_i| \) the grid parameter, then there exists a constant \( C \) independent of \( h \) such that
\[
\| v - \pi_h v \|_{H^1} \leq C h^k |v|_{H^{k+1}},
\]
where \( |v|_{H^{k+1}} = \left\| \frac{d^{k+1}}{dx^{k+1}} v \right\|_{L^2} \) denotes the semi-norm in \( H^{k+1} \).

Theorem 5. Let \( u \in V \), \( u : I \to \mathbb{R} \) with \( I = (a,b) \subset \mathbb{R} \), be the solution of (2.22) and denote by \( u_h \in V_h \) the solution of the discrete problem (2.23), \( V_h \subset V \) with a piece-wise polynomial basis of degree \( k \). Moreover, let \( u \in H^{p+1}(I) \) for suitable \( p \) such that \( k \leq p \), then
\[
\| u - u_h \|_{H^1} \leq \frac{K}{\alpha} C h^k |u|_{H^{k+1}},
\]
where \( C \) is independent of \( h \).

Proof. Céa’s lemma and Theorem 4 give
\[
\| u - u_h \|_{H^1} \leq \frac{K}{\alpha} \inf_{v_h \in V_h} \| u - v_h \|_{H^1} \leq \frac{K}{\alpha} \| u - \pi_h u \|_{H^1} \leq \frac{K}{\alpha} C h^k |u|_{H^{k+1}}.
\]

\[\Box\]

2.5 Spectral methods I

Spectral methods are very closely related to finite element (FE) methods. The main difference is that in the weak formulation (2.23),
\[
\text{Find } u_h \in V_h \text{ such that } a(u_h, v_h) = l(v_h) \quad \forall \, v_h \in V_h,
\]
the space \( V_h \) is spanned by functions with global support on the computational domain \( \Omega \), instead of the local piece-wise polynomials in FE methods. This leads to superior convergence rates of spectral methods, depending on the regularity of the exact solution, not on the polynomial degree as in FE methods. For analytic solutions exponential convergence of spectral methods is achieved, such that very good approximations can be obtained with a relatively small number of basis functions. A disadvantage is that spectral methods can usually be formulated only on quadrilaterals in higher dimensions, which makes the implementation of complex domain boundaries (and boundary conditions) impossible, or at least very challenging - see spectral element methods. Moreover, the stiffness matrices are full because of the global support of basis functions, which makes numerical inversion costly.

The two most popular types of basis functions in spectral methods are

1. orthogonal polynomials (e.g. Hermite, Laguerre, Legendre, etc.),
2. trigonometric functions (Fourier method).
We will treat the Fourier method extensively in a separate section 2.6 because of its vast practical importance. Some basic results about orthogonal polynomials are collected here by means of Legendre polynomials. For each positive integer $N$, we denote by $Q_N$ the space of polynomials with real coefficients of degree $\leq N$ in each dimension. Hence, in 1D we have

$$Q_N(I) = \left\{ v(x) = \sum_{k=0}^{N} a_k x^k, \quad a_k \in \mathbb{R}, \quad x \in I \subset \mathbb{R} \right\},$$

while in 2D on the domain $\Omega = I_x \times I_y$ we have

$$Q_N(\Omega) = \left\{ v(x,y) = \sum_{k,m=0}^{N} a_{km} x^k y^m, \quad a_{km} \in \mathbb{R}, \quad x \in I_x, \quad y \in I_y \right\}.$$

In spectral methods with orthogonal polynomials, we denote the solution $u_h \in V_h$ in (2.27) by $u_N \in V_N$ to indicate the polynomial degree, and choose $V_N \subset Q_N(\Omega)$. Because (2.27) falls into the class of Galerkin methods, the Lax-Milgram theorem still holds, usually with $V = L^2$, settling the existence and uniqueness of solutions. Moreover, the following a priori estimate can be proven:

**Theorem 6.** Let $u \in V$ be the exact solution of the variational problem (2.21) and suppose $u \in H^{s+1}(\Omega)$, for some $s \geq 0$. If $u_N \in V_N$ is the approximate spectral solution, it holds that

$$||u - u_N||_{H^1} \leq C_s N^{-s} ||u||_{H^{s+1}},$$

where $N$ is the polynomial degree and the constant $C_s$ may depend on $s$ but not on $N$.

From the theorem it is evident that the convergence rate improves with the regularity of the solution (larger $s$ means more regular solutions $u$), even with the polynomial degree $N$ fixed. This is in contrast to FE methods, see Theorem 5, where the convergence rate is given by the polynomial degree. In particular, for analytic functions ($s = \infty$), the convergence is no more algebraic, but exponential:

$$\exists \gamma > 0 : \quad ||u - u_N||_{H^1} \leq C \exp(-\gamma N).$$

### 2.5.1 Orthogonal Legendre polynomials

Let us consider a function $f \in L^2(-1,1)$, where we recall the space

$$L^2(-1,1) = \left\{ f : (-1,1) \to \mathbb{R} : ||f||_{L^2} = \left( \int_{-1}^{1} |f|^2 \, dx \right)^{1/2} < \infty \right\}.$$

$L^2(-1,1)$ is a Hilbert space with the scalar product

$$(f,g) = \int_{-1}^{1} f \, g \, dx.$$
The orthogonal Legendre polynomials $L_k \in \mathbb{P}_k$ constitute a sequence with the following orthogonality property:

$$(L_k, L_m) = \begin{cases} 0 & \text{if } m \neq n \\ (k + 1/2)^{-1} \text{ im } m = k. \end{cases}$$

They are linearly independent and form a basis of $L^2(-1,1)$. Consequently, any function $f \in L^2(-1,1)$ admits the series expansion

$$f(x) = \sum_{k=0}^{\infty} \hat{f}_k L_k(x).$$

The Legendre coefficients $\hat{f}_k \in \mathbb{R}$ can easily be computed by exploiting the orthogonality property,

$$(f, L_k) = \sum_{m=0}^{\infty} \hat{f}_m (L_m, L_k) = \hat{f}_k (k + 1/2)^{-1} \quad \implies \quad \hat{f}_k = (f, L_k) (k + 1/2).$$

From this immediately derives the Parseval identity,

$$||f||^2_{L^2} = \sum_{k=0}^{\infty} \hat{f}_k^2 \ ||L_k||^2_{L^2} = \sum_{k=0}^{\infty} \frac{\hat{f}_k^2}{k + 1/2}.$$

Legendre polynomials can be computed recursively from

$$L_0 = 1, \quad L_1 = x, \quad L_{k+1} = \frac{2k+1}{k+1} x L_k - \frac{k}{k+1} L_{k-1}, \quad k = 1, 2, \ldots.$$

The first view are depicted in Figure 2.5. The Legendre series converges to $f$ in the $L^2$-norm. Let us denote by

$$f_N(x) = \sum_{k=0}^{N} \hat{f}_k L_k(x),$$

the $N$-th order truncation of the Legendre series, then

$$\lim_{N \to \infty} ||f - f_N||_{L^2} = 0.$$

Moreover, if $f \in H^s(-1,1)$ we can also have the rate of convergence:

$$||f - f_N||_{L^2} \leq C_s N^{-s} ||f^{(s)}||_{L^2}.$$

At this point we can prove that $f_N$ is the orthogonal projection of $f$ on $\mathbb{Q}_N$ with respect to the $L^2$ scalar product, which means

$$(f - f_N, p) = 0 \quad \forall \ p \in \mathbb{Q}_N.$$  (2.28)
Figure 2.5: Orthogonal Legendre polynomials on the reference interval $[-1, 1]$.

To prove this relation, note that $L_k$, $k \leq N$, form a basis of $\mathbb{Q}_N$. Therefore,

$$(f - f_N, L_k) = \left( \sum_{m=N+1}^{\infty} \hat{f}_m L_m, L_k \right) = 0 \quad \text{for } k \leq N,$$

because of the orthogonality property, which proves (2.28). From (2.28) it follows that $f_N$ is the function that minimizes the distance of $f$ from $\mathbb{Q}_N$,

$$||f - f_N||_{L^2} = \inf_{p \in \mathbb{Q}_N} ||f - p||_{L^2}. \quad (2.29)$$

For this purpose we reach back to a trick that we already used in the proof of Céa’s lemma:

$$||f - f_N||_{L^2}^2 = (f - f_N, f - f_N) = (f - f_N, f - p + p - f_N) = (f - f_N, f - p),$$

since $(f - f_N, p - f_N) = 0$ because of (2.28). Then we use the Cauchy-Schwartz inequality to obtain

$$||f - f_N||_{L^2}^2 = (f - f_N, f - p) \leq ||f - f_N||_{L^2} ||f - p||_{L^2},$$

which yields (2.29).

### 2.6 Spectral methods II: Fourier methods

Linear PDEs with constant coefficients can be "diagonalised" on periodic domains using an expansion of the solution in Fourier modes, and using the Fourier transform in infinite domains. Hence Fourier series and transforms are an essential tool for understanding the solution of linear PDEs. They can also be used thanks
to their discrete representation, the discrete Fourier transform, for numerical approximation or analysis of the data. This is often quite efficient thanks to a fast algorithm, called Fast Fourier Transform (FFT), for computing the discrete Fourier transform.

### 2.6.1 Fourier series

Let $f : \mathbb{R} \to \mathbb{C}$ be a periodic function of period $L$, that is integrable on $[0, L]$. Its Fourier coefficients can then be defined by

$$\hat{f}_k := \frac{1}{L} \int_0^L f(x) e^{-\frac{i2\pi}{L} kx} \, dx, \quad k \in \mathbb{Z},$$

and the partial Fourier series associated to $f$ can be defined by

$$S_N(f) := \sum_{k=-N}^{N} \hat{f}_k e^{\frac{i2\pi}{L} kx}.$$  

Under adequate conditions the series $S_N(f)$ converges to the so-called Fourier series of $f$ when $N \to +\infty$:

**Theorem 7** (Dirichlet). Assume $f$ $L$-periodic and piecewise $C^0$. Then its Fourier series converges at any point $x_0$ and

$$\lim_{N \to +\infty} S_N(f)(x_0) = \frac{1}{2}(f(x_0^+) + f(x_0^-)),$$

where $f(x_0^+)$ and $f(x_0^-)$ define respectively the right and left limit of $f$ (which is assumed only piecewise continuous) at $x_0$.

Obviously, if $f$ is continuous at a point $x$, it is equal to its Fourier series at $x$:

$$f(x) = \sum_{k=-\infty}^{+\infty} \hat{f}_k e^{\frac{i2\pi}{L} kx}.$$  

An important identity related to Fourier series is the Parseval identity

$$\frac{1}{L} \int_0^L |f(x)|^2 \, dx = \sum_{N=-\infty}^{+\infty} |\hat{f}_k|^2,$$

which follows from

$$\frac{1}{L} \int_0^L e^{\frac{i2\pi}{L} kx} e^{\frac{i2\pi}{L} k'x} \, dx = \frac{1}{L} \int_0^L e^{i\frac{2\pi}{L}(k-k')x} \, dx$$

$$= \frac{1}{L} \int_0^L \left[ \cos \left( \frac{2\pi}{L}(k-k')x \right) + i \sin \left( \frac{2\pi}{L}(k-k')x \right) \right] \, dx$$

$$= \delta_{k,k'},$$

where $\delta_{k,k'} = 1$ for $k = k'$ and zero otherwise (Kronecker delta).
2.6.2 The Fourier transform

For non-periodic functions defined over the whole \(\mathbb{R}\) (this can be extended to several dimensions), the Fourier transform provides a tool analogous to Fourier series for periodic functions. The space of square integrable functions over \(\mathbb{R}\) taking values in \(\mathbb{C}\),

\[ L^2(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{C} \mid \int_{-\infty}^{+\infty} |f(x)|^2 \, dx < +\infty \right\} , \]

defines a Hilbert space with the scalar product

\[ (f, g) = \int_{-\infty}^{+\infty} f(x) \bar{g}(x) \, dx. \]

For any function \(f \in L^2(\mathbb{R})\) we can define its Fourier transform by

\[ \hat{f}(\xi) := \int_{-\infty}^{+\infty} f(x) e^{-i\xi x} \, dx, \quad \xi \in \mathbb{R}, \]

and its inverse Fourier transform by

\[ f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\xi) e^{i\xi x} \, d\xi. \]

Here, Parseval’s formula reads

\[ (f, g) = \int_{-\infty}^{+\infty} f(x) \bar{g}(x) \, dx = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\xi) \bar{\hat{g}}(\xi) \, d\xi = \frac{1}{2\pi} (\hat{f}, \hat{g}), \]

which follows formally from

\[ \int_{-\infty}^{+\infty} e^{i(\eta-\xi)x} \, dx = 2\pi \delta(\eta - \xi), \]

where \(\delta(\cdot)\) denotes the Dirac delta distribution. In the special case when \(f = g\) we get the Plancherel formula

\[ \|f\|^2 = (f, f) = \int_{-\infty}^{+\infty} |f(x)|^2 \, dx = \frac{1}{2\pi} \|\hat{f}(\xi)\|^2. \]

2.6.3 The Discrete Fourier Transform

Let \(\mathcal{F}_N \in \mathbb{C}^{N \times N}\) be the symmetric matrix formed with the inverse powers of the \(N^{th}\) roots of unity, the coefficients of which are given by

\[ (\mathcal{F}_N)_{j,k} := \left( e^{2\pi i k/N} \right)^{-j} = e^{-i\frac{2\pi}{N} jk}, \quad 0 \leq j, k \leq N - 1. \tag{2.30} \]

Denoting by \(\omega_N = e^{i\frac{2\pi}{N}}\), we have \((\mathcal{F}_N)_{jk} = \omega_N^{-jk}\). The adjoint, or conjugate transpose of \(\mathcal{F}_N\) is the matrix \(\mathcal{F}_N^*\) with coefficients \((\mathcal{F}_N^*)_{jk} = \omega_N^{jk}\). Notice that the columns of \(\mathcal{F}_N\), i.e with \(k\) fixed and \(0 \leq j \leq N - 1\), are the vectors interpolating the functions \(x \mapsto e^{-i(2\pi/L)kx}\) at the grid points \(x_j = jL/N\) of the interval \([0, L]\).
Definition 3. Discrete Fourier Transform.

- The **discrete Fourier transform** of a vector \( U = (u_0, \ldots, u_{N-1})^\top \in \mathbb{C}^N \) is the vector \( \hat{U} = (\hat{u}_0, \ldots, \hat{u}_{N-1})^\top := \mathcal{F}_N U \). This can also be written component by component

\[
\hat{u}_k = \sum_{j=0}^{N-1} u_j e^{-i \frac{2\pi}{N} jk}, \quad k = 0, \ldots, N - 1. \tag{2.31}
\]

- The **inverse discrete Fourier transform** of a vector \( \hat{U} \in \mathbb{C}^N \) is the vector \( U = \mathcal{F}_N^{-1} \hat{U} = \frac{1}{N} \mathcal{F}_N^* \hat{U} \). This becomes component by component

\[
u_j = \frac{1}{N} \sum_{k=0}^{N-1} \hat{u}_k e^{i \frac{2\pi}{N} jk}, \quad j = 0, \ldots, N - 1.
\]

Lemma 3. The matrix \( \mathcal{F}_N \) verifies \( \mathcal{F}_N \mathcal{F}_N^* = N I_N \), where \( I_N \) is the identity matrix of dimension \( N \).

*Proof.* We have

\[
(\mathcal{F}_N \mathcal{F}_N^*)_{j,k} = \sum_{l=0}^{N-1} \omega_N^{-jl} \omega_N^{lk} = \sum_{l=0}^{N-1} e^{i \frac{2\pi}{N} (k-j)} = \sum_{l=0}^{N-1} \left( e^{i \frac{2\pi}{N} (k-j)} \right)^l.
\]

For \( j = k \) the summation is trivial and yields \( N \), while for \( j \neq k \) we have a geometric series such that

\[
\sum_{l=0}^{N-1} \left( e^{i \frac{2\pi}{N} (k-j)} \right)^l = \frac{1 - \left( e^{i \frac{2\pi}{N} (k-j)} \right)^N}{1 - e^{i \frac{2\pi}{N} (k-j)}} = \frac{1 - e^{i2\pi(k-j)}}{1 - e^{i \frac{2\pi}{N} (k-j)}} = 0, \quad j \neq k.
\]

Therefore, \((\mathcal{F}_N \mathcal{F}_N^*)_{j,k} = 0 \) if \( j \neq k \) and \((\mathcal{F}_N \mathcal{F}_N^*)_{j,k} = N \) if \( j = k \). \(\square\)

Corollary 1. Let \( U, V \in \mathbb{C}^N \) and denote by \( \hat{U} = \mathcal{F}_N U \) and \( \hat{V} = \mathcal{F}_N V \) their discrete Fourier transforms. Then we have

- the **discrete Parseval identity**:

\[
(U, V) = U^\top \bar{V} = \frac{1}{N} \hat{U}^\top \bar{\hat{V}} = \frac{1}{N}(\hat{U}, \hat{V}),
\]

- The **discrete Plancherel identity**:

\[
\|U\| = \frac{1}{N} \|\hat{U}\|,
\]

where \((.,.)\) and \(\|\|\) denote the usual euclidian dot product and norm in \( \mathbb{C}^N \).
Proof. The scalar product in \( \mathbb{C}^N \) of \( U = (u_0, \ldots, u_{N-1})^\top \) and \( V = (v_0, \ldots, v_{N-1})^\top \) is defined by
\[
(U, V) = \sum_{i=0}^{N-1} u_i \bar{v}_i = U^\top \bar{V}.
\]

Then using the definition of the inverse discrete Fourier transform, we have \( U = \frac{1}{N} \mathcal{F}_N^* \hat{U}, V = \frac{1}{N} \mathcal{F}_N^* \hat{V} \), we get
\[
U^\top \bar{V} = \frac{1}{N^2} (\mathcal{F}_N^* \hat{U})^\top \mathcal{F}_N^* \bar{\hat{V}} = \frac{1}{N^2} \hat{U}^\top \mathcal{F}_N \mathcal{F}_N^* \hat{V} = \frac{1}{N} \hat{U}^\top \hat{V},
\]
as \( \mathcal{F}_N^\top = \mathcal{F}_N^* \) and \( \mathcal{F}_N^* = N \mathcal{F}_N^{-1} \). The Plancherel identity follows from the Parseval identity by taking \( U = V \).

\( \square \)

Remark 1. The discrete Fourier transform is defined as a matrix-vector multiplication. Its computation hence requires a priori \( N^2 \) multiplications and additions. But because of the specific structure of the matrix there exists a very fast algorithm, called Fast Fourier Transform (FFT) for performing it in \( O(N \log_2 N) \) operations. This makes it particularly interesting for many applications, and many fast PDE solvers make use of it.

2.6.4 Circulant matrices

An important class of matrices that occur frequently in numerical discretizations of PDEs on periodic domains are the so-called circulant matrices. Such matrices are characterised by one coefficient for each diagonal and are of the form
\[
C = \begin{pmatrix}
c_0 & c_1 & c_2 & \cdots & \cdots & c_{N-1} \\
c_{N-1} & c_0 & c_1 & \cdots & \cdots & c_{N-2} \\
c_{N-2} & c_{N-1} & c_0 & \cdots & \cdots & c_{N-3} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
c_3 & \cdots & c_0 & c_1 & c_2 & \cdots \\
c_2 & c_{N-1} & c_0 & c_1 & \cdots & \cdots \\
c_1 & \cdots & \cdots & \cdots & \cdots & c_{N-1} & c_0
\end{pmatrix}
\]
with \( c_0, c_1, \ldots, c_{N-1} \in \mathbb{R} \).

Proposition 3. The eigenvalues of the circulant matrix \( C \in \mathbb{R}^{N \times N} \) are given by
\[
\lambda_k = \sum_{j=0}^{N-1} c_j \left( e^{i2\pi j k} \right)^j = \sum_{j=0}^{N-1} c_j \omega_N^j, \quad k = 0, \ldots, N-1,
\]
where \( \omega_N = e^{i2\pi/N} \).
Proof. Let $J$ be the circulant matrix obtained from $C$ by taking $c_1 = 1$ and $c_j = 0$ for $j \neq 1$, hence with entries $J_{j,k} = \delta_{k, \mod(j+1,N)}$, where $\delta_{k,k'}$ is the Kronecker delta. We note that the powers of $J$ compute to

\[
J^2_{j,k} = \sum_{i=0}^{N-1} J_{i,j} J_{k,i} = \sum_{i=0}^{N-1} \delta_{i,j+1} \delta_{k,i+1} = \delta_{k,j+2},
\]

\[
J^3_{j,k} = \sum_{i=0}^{N-1} J^2_{i,j} J_{k,i} = \sum_{i=0}^{N-1} \delta_{i,j+2} \delta_{k,i+1} = \delta_{k,j+3},
\]

\[
J^n_{j,k} = \delta_{k, \mod(j+n,N)}, \quad n \geq 1.
\]

Therefore, $C$ can be written as a polynomial in $J$

\[
C = \sum_{j=0}^{N-1} c_j J^j.
\]

Moreover, as $J^N = I$, the eigenvalues of $J$ are the $N$-th roots of unity that are given by $\omega_N^k = e^{i(2\pi/N)k}$. Looking for the corresponding eigenvectors $X_k \in \mathbb{C}^N$ of $J$ such that $JX_k = \omega_N^k X_k$, we find that

\[
X_k = \begin{pmatrix}
1 \\
\omega_N^k \\
\omega_N^{2k} \\
\vdots \\
\omega_N^{(N-1)k}
\end{pmatrix}.
\] (2.33)

We then have that

\[
CX_k = \sum_{j=0}^{N-1} c_j J^j X_k = \sum_{j=0}^{N-1} c_j \omega_N^{jk} X_k,
\]

and so the eigenvalues of $C$ associated to the eigenvectors $X_k$ are

\[
\lambda_k = \sum_{j=0}^{N-1} c_j \omega_N^{jk}.
\]

Sidenote: the eigenvectors $X_k$ in (2.33) of any circulant matrix $C$ are the columns of the conjugate transpose Fourier matrix $F_N^*$ defined in (2.30).

Proposition 4. Any circulant matrix $C$ can be written in the form $C = \frac{1}{N} F_N^* \Lambda F_N$ where $F_N$ is the discrete Fourier matrix from (2.30) and $\Lambda = \text{diag}(\lambda_0, \ldots, \lambda_{N-1})$ is the diagonal matrix of the eigenvalues of $C$. In particular, all circulant matrices have the same eigenvectors (which are the columns of $F_N^*$), and any matrix of the form $F_N^* \Lambda F_N$ is circulant.

Corollary 2. We have the following properties:
• The product of two circulant matrix is circulant matrix.

• A circulant matrix whose eigenvalues are all non vanishing is invertible and its inverse is circulant.

Proof. The key point is that all circulant matrices can be diagonalized in the same basis of eigenvectors. If $C_1$ and $C_2$ are two circulant matrices of length $N$, we have $C_1 = \frac{1}{N} \mathbf{F}_N^* \Lambda_1 \mathbf{F}_N$ and $C_2 = \frac{1}{N} \mathbf{F}_N^* \Lambda_2 \mathbf{F}_N$ so that $C_1 C_2 = \frac{1}{N} \mathbf{F}_N^* \Lambda_1 \Lambda_2 \mathbf{F}_N$. Moreover, if all eigenvalues of $C = \frac{1}{N} \mathbf{F}_N^* \Lambda \mathbf{F}_N$ are non vanishing, $\Lambda^{-1}$ is well defined and

$$1/N \mathbf{F}_N^* \Lambda \mathbf{F}_N \Lambda^{-1} \mathbf{F}_N = I_N.$$ 

So the inverse of $C$ is the circulant matrix $1/N \mathbf{F}_N^* \Lambda^{-1} \mathbf{F}_N$. □

2.6.5 Stability of the discrete Laplacian with periodic boundary conditions

For periodic boundary conditions, the matrix of the discrete Laplacian given in (2.8) is circulant, with $c_0 = 2/h^2$, $c_1 = -1/h^2$, $c_{N-1} = -1/h^2$ and all the other terms vanish. Hence formula (2.32) for computing its eigenvalues can be used to verify its stability. It yields

$$\lambda_k = c_0 + c_1 \omega_N^k + c_{N-1} \omega_N^{(N-1)k}$$

$$= \frac{1}{h^2} \left(2 - e^{i \frac{2\pi}{N} k} - e^{i \frac{2\pi}{N} (N-1)k}\right)$$

$$= \frac{2}{h^2} \left(1 - \cos \left(\frac{2\pi k}{N}\right)\right)$$

$$= \frac{4}{h^2} \sin^2 \left(\frac{\pi k}{N}\right), \quad k = 0, \ldots, N - 1.$$

Here we used that $\cos(x + y) = \cos(x) \cos(y) - \sin(x) \sin(y)$. The matrix is not invertible because for $k = 0$ we get $\lambda_0 = 0$. In order to fix the constant, we assume that $\hat{\phi}_0 = \phi_0 + \cdots + \phi_{N-1} = 0$, c.f. formula (2.31). This discards the eigenvalue $\lambda_0 = 0$, so that the rest of the matrix is invertible and the smallest eigenvalue corresponds to

$$\lambda_1 = \lambda_{N-1} = \frac{4}{h^2} \sin^2 \left(\frac{\pi}{N}\right) = \frac{4}{h^2} \sin^2 \left(\frac{\pi h}{L}\right).$$

We can now proceed like in the case of homogeneous Dirichlet boundary conditions. As $\sin x \sim x$ for $x$ close to 0, we find

$$\lim_{h \to 0} \lambda_1 = \frac{4\pi^2}{L^2},$$

which is strictly larger than 0, so that all eigenvalues are bounded from below by the half of that number, after some small enough $h_0$, and the others are a finite number of strictly positive values. This proves that for all $N$ all eigenvalues of $A_h$ are positive and bounded from below by a constant independent of $h$ which proves stability.
2.6.6 The Fourier pseudospectral method

This method is also called the Fourier spectral collocation method. See the books [14, 3] for a detailed description. The principle of spectral methods is to look for an approximation of the solution of some PDE as an \(N\)-term expansion in some well chosen basis verifying the boundary conditions. In order to determine the basis coefficients of this expansion, pseudospectral methods, also called spectral collocation methods, rely on plugging this expansion along with its derivatives into the PDE to be solved and make it be exact at \(N\) well chosen collocation points (which play a similar role as interpolation points), so as to get a square system of equations to be solved for the basis coefficients.

For periodic problems a good choice of basis is the discrete Fourier basis, and the collocation points are uniformly chosen in one period. In this case, the method is the Fourier pseudospectral, or Fourier collocation method. Then for solving in a periodic domain a PDE of the form

\[ \mathcal{L}u = f, \]

where \(\mathcal{L}\) is an arbitrary (nonlinear) differential operator, one approximates the exact solution \(u\) by

\[ u_N(x) = \frac{1}{N} \left( \sum_{k=-N/2+1}^{N/2-1} a_k e^{i\frac{2\pi}{N} k x} + a_{-N/2} \cos \frac{\pi N x}{L} \right) \quad \text{for } x \in [0, L), \quad (2.34) \]

and solves for the \(N\) coefficients \(a_k\), assuming \(N\) even, such that

\[ \mathcal{L}u_N(x_j) = f(x_j), \quad x_j = j \frac{L}{N}, \quad 0 \leq j \leq N - 1. \]

The \((x_j)_{0 \leq j \leq N - 1}\) are the collocation points. It is important for the problem to be well-posed that the number of expansion coefficients \(a_k\) is equal to the number of collocation points \(N\). Expressing \(u_N\) at the collocation points, we get

\[ u_j := u_N(x_j) = \frac{1}{N} \left( \sum_{k=-N/2-1}^{N/2+1} a_k e^{i\frac{2\pi}{N} kj} + a_{-N/2} (-1)^j \right). \]

Let us comment on the representation of the last term of \(u_N\) in (2.34) as a \(\cos\) instead of using the \(k = N/2\) or \(k = -N/2\) modes. First of all we get real-valued functions for \(a_{-k} = a_k\). We also notice that on the grid we have

\[ e^{i\frac{2\pi}{N} j N/2} = e^{i \pi j} = (-1)^j = e^{-i\frac{2\pi}{N} j N/2} = \cos(j\pi). \]

Therefore, the grid representations of the functions

\[ x \mapsto e^{i\frac{2\pi}{N} x}, \quad x \mapsto e^{-i\frac{2\pi}{N} x}, \quad x \mapsto \cos \left( \frac{2\pi N}{L} x \right) \]

are...
are the same and we can use any of the three representations on the grid. On the other hand, the only instance when the continuous form (2.34) is needed and not the grid representation is for computing the derivatives. For this, from the three forms only the cos-representation has a good behavior. Indeed,

\[
\frac{d}{dx} \left( \cos \left( \frac{2\pi N}{L} x \right) \right)_{x=x_j} = -\frac{\pi N}{L} \sin(\pi j) = 0 \quad 0 \leq j \leq N - 1,
\]

\[
\frac{d}{dx} \left( e^{i \frac{2\pi N}{L} x} \right)_{x=x_j} = \frac{i\pi N}{L} e^{i\pi j} = (-1)^j \frac{i\pi N}{L} \quad 0 \leq j \leq N - 1,
\]

\[
\frac{d}{dx} \left( e^{-i \frac{2\pi N}{L} x} \right)_{x=x_j} = -\frac{i\pi N}{L} e^{-i\pi j} = (-1)^j \frac{-i\pi N}{L} \quad 0 \leq j \leq N - 1.
\]

We notice that the three possibilities yield three different grid approximations of the derivatives. The problem with the exponentials lies in the fact that the grid derivative of these modes is not real (it is purely imaginary). And for this reason the derivative of a real grid function would not be a real grid function, which is not acceptable.

**Remark 2.** Note that this problem appears for all odd derivatives but not for even derivatives which always have the same grid representation for the three choices.

Finally we can summarise the computation of the derivatives as follows: For \( m \) even the \( m \)th grid derivative is given by

\[
u^{(m)}_N(x_j) = \sum_{k=-N/2}^{N/2-1} \left( i \frac{2\pi}{L} k \right)^m a_k e^{i \frac{2\pi}{L} kj}, \quad 0 \leq j \leq N - 1,
\]

For \( m \) odd, the contribution of the \( N/2 \) mode vanishes and the \( m \)th grid derivative is given by

\[
u^{(m)}_N(x_j) = \sum_{k=-N/2+1}^{N/2-1} \left( i \frac{2\pi}{L} k \right)^m a_k e^{i \frac{2\pi}{L} kj}, \quad 0 \leq j \leq N - 1.
\]

Let us now apply this pseudospectral method for the \( L \)-periodic 1D Poisson problem

\[-u''(x) = f(x), \quad \int_0^L u(x) \, dx = \alpha \in \mathbb{R}.\]

For this problem to be well posed the average of \( u \) has been set \( \alpha \).

The pseudospectral method consists in looking for an approximation of the periodic solution \( u \) of the form (2.34) and writing that

\[-u''_N(x_j) = f(x_j), \quad x_j = jL/N, \quad 0 \leq j \leq N - 1. \quad (2.35)\]

Denoting the right-hand side by \( f_j = f(x_j) \) and \( R_h = (f_0, \ldots, f_{N-1})^\top \), with its discrete Fourier transform given by \( \mathcal{F}_N R_h = \hat{R}_h = (\hat{f}_0, \ldots, \hat{f}_{N-1})^\top \), we have from
Lemma 3 that $R_h = \frac{1}{N} \mathcal{F}^*_N \mathcal{F}_N R_h$ and can hence write

$$f_j = \frac{1}{N} \sum_{k=0}^{N-1} \hat{f}_k e^{i \frac{2\pi}{N} kj}.$$ 

Then (2.35) becomes

$$\frac{1}{N} \sum_{k=-N/2}^{N/2-1} \left( \frac{2\pi k}{L} \right)^2 a_k e^{i \frac{2\pi}{N} kj} = \frac{1}{N} \sum_{k=0}^{N-1} \hat{f}_k e^{i \frac{2\pi}{N} kj}$$

$$= \frac{1}{N} \sum_{k=0}^{N/2-1} \hat{f}_k e^{i \frac{2\pi}{N} kj} + \frac{1}{N} \sum_{k=-N/2}^{-1} \hat{f}_{k+N} e^{i \frac{2\pi}{N} kj}.$$ 

Then, the index transformation $k = k' + N$ in the last leads to

$$\frac{1}{N} \sum_{k=-N/2}^{N/2-1} \left( \frac{2\pi k}{L} \right)^2 a_k e^{i \frac{2\pi}{N} kj} = \frac{1}{N} \sum_{k=0}^{N/2-1} \hat{f}_k e^{i \frac{2\pi}{N} kj} + \frac{1}{N} \sum_{k'=-N/2}^{-1} \hat{f}_{k'+N} e^{i \frac{2\pi}{N} k'j}.$$ 

Due to the orthogonality of the Fourier basis functions, we then get the solution $a_k$ by indentifying the coefficients in the two partial Fourier series:

$$\left( \frac{2k\pi}{L} \right)^2 a_k = \hat{f}_k, \quad 0 < k \leq N/2 - 1,$$

$$\left( \frac{2k\pi}{L} \right)^2 a_k = \hat{f}_{k+N}, \quad -N/2 \leq k \leq -1.$$ 

This gives directly an expression for $a_k$ except for $k = 0$ for which we need to use the known average value of $u$:

$$a_0 = \alpha.$$ 

The change of summation needs very often to be performed in computations involving the Discrete Fourier Transform. It consists in shifting the switching the first N/2 modes with the others. This is done in Matlab using the function `fftshift`. The corresponding algorithm then reads, given the grid function $R_h = (f_0, \ldots, f_{N-1})$ and the average value $\alpha$ of $u$:

1. Perform FFT of $R_h$ followed by `fftshift`
2. Compute $a_k$ from $\hat{f}_k$ for $k \neq 0$, and set $a_0 = \alpha$
3. Perform an `fftshift` of the vector containing the $a_k$ and then an inverse FFT gives the result $u$. 

53
Chapter 3

Numerical schemes for the Vlasov-Poisson system

3.1 The Vlasov-Poisson model for electrons

Suppose we want to describe non-magnetized plasma ($B = 0$) on time scales short enough such that the ions can be considered stationary due to their larger mass (and thus inertia) compared to the electrons. Setting the physical constants to one and making sure to keep the right sign for the charge of electrons and ions, the appropriate model reads

\[
\begin{align*}
\frac{\partial f}{\partial t} + v \cdot \nabla_x f - E \cdot \nabla_v f &= 0, \\
f(x, v, 0) &= f_0(x, v), \\
- \Delta \phi &= 1 - \int f \, dv, \quad E = -\nabla \phi,
\end{align*}
\]  

(3.1)

where $f : \mathbb{R}^6 \times \mathbb{R} \to \mathbb{R}^+$ is the electron distribution function with initial condition $f_0$, $\phi : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}$ denotes the electrostatic potential and $E = -\nabla \phi$ stands for the self-consistent electric field. The ion density has been set to one.

Denoting by $A = (v, -E) \uparrow$ the advection field in $(x, v)$-space, called the phase space, the Vlasov equation can be written as an advection equation in phase space of the form

\[
\frac{\partial f}{\partial t} + A \cdot \nabla_{x,v} f = 0.
\]

(3.2)

Moreover, as $\nabla_{x,v} \cdot A = 0$, the Vlasov equation can also be written in the conservative form

\[
\frac{\partial f}{\partial t} + \nabla_{x,v} \cdot (Af) = 0.
\]
3.1.1 The Vlasov equation in a given potential

First verification tests for a numerical Vlasov-Poisson solver consist of considering the Vlasov equation in given potentials where the solution can be computed exactly with the method of characteristics. Let us denote by \( z = (x, v) \) the phase space variable and consider the Vlasov equation in advective form (3.2) with given potential \( \phi : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R} \). Hence the advection field \( A : \mathbb{R}^6 \times \mathbb{R} \to \mathbb{R}^3 \) is given; the Vlasov equation is also called the free transport equation (FTE) in this case. We then call the solution \( Z \) of the system

\[
\begin{cases}
\frac{dZ}{ds} = A(Z, s), \\
Z(t) = z,
\end{cases}
\tag{3.3}
\]

the “characteristics” of the FTE. Supposing that \( A \) is sufficiently smooth, let us say \( A \in C^2 \) with bounded derivatives up to second order, the Cauchy-Lipschitz theorem guarantees a solution \( Z = Z(s; t, z) \). The application

\[
\Phi_t^s : z \mapsto Z(s; t, z)
\]

defines the flow of the characteristic equations. It is now easy to show that the FTE has a unique solution of the form

\[
f(t, z) = f_0(Z(0; t, z)). \tag{3.4}
\]

This follows from the fact that \( f \) is constant along the characteristics,

\[
\frac{d}{ds} f(s, Z(s)) = 0. \tag{3.5}
\]

Indeed,

\[
\frac{d}{ds} f(s, Z(s)) = \frac{\partial f}{\partial s} + \frac{dZ}{ds} \cdot \nabla_z f = \frac{\partial f}{\partial s} + A(Z, s) \cdot \nabla_z f \overset{(3.2)}{=} 0.
\]

Integration of (3.5) between 0 and \( t \) then yields

\[
f(t, Z(t; t, z)) - f(0, Z(0; t, z)) = 0,
\]

\[
\Leftrightarrow f(t, z) = f(0, Z(0; t, z))
\]

The flow \( \Phi_t^s \) has the group property

\[
\Phi_{t_1}^{t_3} = \Phi_{t_2}^{t_3} \circ \Phi_{t_1}^{t_2},
\]

and the inverse of \( \Phi_t^s \) is given by \( \Phi_t^{-s} \). Hence the dynamics of the FTE are reversible. Moreover, the flow \( \Phi_t^s \) conserves phase space volume. This means that for any volume \( \Omega(t) \) that is advected by the flow,

\[
\Omega(t) = \{ z \in \mathbb{R}^6 : z = \Phi_0^t(z_0) \text{ with } z_0 \in \Omega_0 \subset \mathbb{R}^6 \},
\]

55
we have
\[ \frac{d}{dt} \int_{\Omega(t)} d^6z = \int_{\Omega_0} \frac{d}{dt} \left| \det \frac{\partial \Phi_t^0}{\partial z} \right| d^6z_0 = 0. \]
This follows Liouville's theorem which says
\[ \nabla \cdot A = 0 \Rightarrow \frac{d}{dt} \det \frac{\partial \Phi_t^0}{\partial z} = 0, \]
for the divergence-free free vector field A.

**Examples of exact solutions**

1. Consider the free streaming equation
\[ \frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} = 0. \]
In this case the characteristics satisfy
\[ \frac{dX}{ds} = V(s) \quad X(t) = x, \quad \frac{dV}{ds} = 0 \quad V(t) = v. \]
Thus we have \( V(s; t, x, v) = v \) and \( X(s; t, x, v) = x + (s - t)v \) which gives us the solution
\[ f(t, x, v) = f_0(x - vt, v). \]

2. Uniform focusing in a particle accelerator (1D model). We then have \( E(x, t) = x \) and the Vlasov equation writes
\[ \frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - x \frac{\partial f}{\partial v} = 0, \]
with the characteristics satisfying
\[ \frac{dX}{ds} = V(s) \quad X(t) = x, \quad \frac{dV}{ds} = -X(s) \quad V(t) = v. \]
Hence we get
\[ X(s; t, x, v) = x \cos(s - t) + v \sin(s - t), \]
\[ V(s; t, x, v) = -x \sin(s - t) + v \cos(s - t), \]
form which we compute the solution
\[ f(t, x, v) = f_0(x \cos t - v \sin t, x \sin t + v \cos t). \]
3.1.2 Conservation properties

The Vlasov-Poisson system has a number of conservation properties that need special attention when developing numerical methods. In principle it is beneficial to retain the exact invariants in numerical methods and when it is not possible to keep them all as is the case here, they can be used to monitor the validity of the simulation by checking that they are approximately conserved with good accuracy.

Proposition 5. The Vlasov-Poisson system verifies the following conservation properties:

- **Maximum principle:**
  \[ 0 \leq f(x,v,t) \leq \max_{(x,v)}(f_0(x,v)). \]

- **Conservation of \(L^p\), norms for \(p\) integer, \(1 \leq p \leq \infty\),**
  \[ \frac{d}{dt} \left( \int f^p \, dx \, dv \right) = 0. \]

- **Conservation of total momentum:**
  \[ \frac{d}{dt} \int v \, f \, dx \, dv = -\frac{d}{dt} \int J \, dx = 0. \]

- **Conservation of total energy:**
  \[ \frac{d}{dt} \left[ \frac{1}{2} \int \|v\|^2 f \, dx \, dv + \frac{1}{2} \int \|E\|^2 \, dx \right] = 0. \]

**Proof.** The maximum principle follows directly from the exact solution (3.4), provided that \(f_0 > 0\) in \(\mathbb{R}^6\).

Multiplying the Vlasov equation (3.1) by \(f^{p-1}\) and integrating on the whole phase-space we obtain
\[ \frac{d}{dt} \left( \int f^p \, dx \, dv \right) = 0, \]
so that the \(L^p\) norms of \(f\) are conserved for all \(p \in \mathbb{N}^+\). Let us notice that the \(L^\infty\)-norm is also conserved thanks to the maximum principle (5).

Let us now proceed to the conservation of momentum. We multiply the Vlasov equation (3.1) by \(v\) and integrate in \(x\) and in \(v\)
\[ \frac{d}{dt} \int v \, f \, dx \, dv + \int \nabla_x \cdot (v \otimes v \, f) \, dx \, dv - \int v \nabla_v \cdot (E \, f) \, dx \, dv = 0. \]
The second integral vanishes as the domain is periodic in \(x\) and in the last integral we integrate by parts to obtain
\[ -\int v \nabla_v \cdot (E \, f) \, dx \, dv = \int E \, f \, dx \, dv = \int E_n \, dx. \]
Inserting in the last expression the Poisson equation \( n = 1 - \nabla \cdot E \) and using
\[
\int E \, dx = - \int \nabla \phi \, dx = 0,
\]
\[
\int E (\nabla \cdot E) \, dx = \int \left[ \nabla \cdot (E \otimes E) + E \times (\nabla \times E) - \frac{1}{2} \nabla |E|^2 \right] \, dx = 0,
\]
where we applied the periodic boundary conditions as well as \( \nabla \times E = -\nabla \nabla \phi = 0 \), it finally follows that
\[
\frac{d}{dt} \int v f \, dx \, dv = -\frac{d}{dt} \int J \, dx = 0.
\]

In order to obtain the energy conservation property, we start by multiplying the Vlasov equation by \( |v|^2 \) and we integrate over phase space to obtain
\[
\frac{d}{dt} \int |v|^2 f \, dx \, dv + \int \nabla_x \cdot (|v|^2 v f) \, dx \, dv - \int |v|^2 \nabla_v \cdot (E f) \, dx \, dv = 0.
\]
As \( f \) is periodic in \( x \), by integrating in \( x \) we get
\[
\int \nabla_x \cdot (|v|^2 v f) \, dx \, dv = 0
\]
and integration by parts yields
\[
\int |v|^2 \nabla_v \cdot E \, dx \, dv = -2 \int v \cdot (E f) \, dx \, dv = 2 \int E \cdot J \, dx.
\]
Therefore,
\[
\frac{d}{dt} \int |v|^2 f \, dx \, dv = 2 \int E \cdot J \, dx = -2 \int \nabla \phi \cdot J \, dx. \tag{3.6}
\]
On the other hand, integrating the Vlasov equation (3.1) with respect to \( v \), yields the charge conservation equation \( \frac{\partial \rho}{\partial t} + \nabla \cdot J = 0 \), where \( \rho = 1 - n \). Then, again integrating by parts leads to
\[
- \int \nabla \phi \cdot J \, dx = \int \phi \nabla \cdot J \, dx = - \int \phi \frac{\partial \rho}{\partial t} \, dx = \int \phi \frac{\partial \Delta \phi}{\partial t} \, dx = -\frac{1}{2} \frac{d}{dt} \int |\nabla \phi|^2 \, dx.
\]
Plugging this equation in (3.6) and using that \( E = -\nabla \phi \), we get the conservation of energy.

\[ \square \]

### 3.1.3 Linearisation of the 1D Vlasov-Poisson equation

In order to verify the correct implementation of a Vlasov-Poisson (VP) solver, we consider the problem linearised around an equilibrium solution. For the VP system, let us first realise that any constant homogeneous distribution function, \( i.e. \) a distribution function which depends neither on \( t \) nor on \( x \) but only on \( v \), is an equilibrium solution of Vlasov-Poisson. Indeed, in this case the partial
derivatives with respect to \( t \) and \( x \) are obviously zero and the third term in the Vlasov equation vanishes because for a homogeneous \( f \), the electric field vanishes as the charge density is uniform and equal to the background density.

Let us now consider the simplest and most important case of thermodynamic equilibrium in 1D. For this the equilibrium distribution, denoted by \( f^0 \), is the homogeneous Maxwellian

\[
f^0(v) = \frac{n_0}{2\pi} e^{-\frac{v^2}{2}}.
\]

We can now linearise VP around this equilibrium state by expanding the distribution function and the electric field in the form of the equilibrium solution plus a small perturbation:

\[
f(x,v,t) = f^0(v) + \varepsilon f^1(x,v,t), \quad E(x,t) = E^0 + \varepsilon E^1(x,t), \quad \text{with } E^0 = 0.
\]

The distribution function \( f \) verifies the VP equations

\[
\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - E(x,t) \frac{\partial f}{\partial v} = 0,
\]

\[
\frac{dE}{dx} = (1 - \int_{-\infty}^{+\infty} f(x,v,t) \, dv),
\]

with an initial condition that we assume of the form

\[
f_0(x,v) = f^0(v) + \varepsilon f^1_0(x,v).
\]

Plugging the expansions of \( f \) and \( E \) in this equation

\[
\varepsilon \left( \frac{\partial f^1}{\partial t} + v \frac{\partial f^1}{\partial x} \right) - (E^0 + \varepsilon E^1)(\frac{df^0}{dv} + \varepsilon \frac{\partial f^1}{\partial v}) = 0,
\]

\[
\varepsilon \frac{dE^1}{dx} = (1 - \int_{-\infty}^{+\infty} (f^0(v) + vf^1(x,v,t)) \, dv).
\]

Neglecting the terms in \( \varepsilon^2 \), we obtain, knowing that \( E^0 = 0 \),

\[
\frac{\partial f^1}{\partial t} + v \frac{\partial f^1}{\partial x} - E^1(x) \frac{df^0}{dv} = 0,
\]

\[
\frac{dE^1}{dx} = -\int_{-\infty}^{+\infty} f^1(x,v,t) \, dv,
\]

with the initial condition \( f^1(x,v,0) = f^1_0(x,v) \). As \( f^0 \) is a known function of \( v \), this equation, the unknowns of which are \( f^1 \) and \( E^1 \), is linear and displays derivatives in \( x \) and \( t \). We can thus compute an analytic solution using, as \( f^1 \) is periodic in \( x \), a Fourier series in \( x \) and a Laplace transform in \( t \).

After a long a quite involved computation due to the Laplace transform and a singularity in the velocity integral, one can obtain a dispersion relation and explicit solution of the linearised problem in form of a series. The dispersion
relation, which gives the frequency $\omega(k)$ of a plane wave in the system as a function of the wave vector $k$, can be expressed simply by

$$D(k, \omega) = 1 + \frac{1}{k^2} \left[ 1 + \frac{\omega}{\sqrt{2k}} Z\left( \frac{\omega}{\sqrt{2k}} \right) \right] = 0.$$  \hspace{1cm} (3.7)

using the so-called plasma dispersion function $Z$ defined by

$$Z(\xi) = \sqrt{\pi} e^{-\xi^2} \left[ -i - \text{erfi}(\xi) \right],$$

where $\text{erfi}(\zeta) = \frac{2}{\sqrt{\pi}} \int_0^\zeta e^{t^2} \, dt$ is the complex error function. It remains to solve numerically the dispersion relation (3.7) for fixed $k$ values to obtain those $\omega(k)$ for which $D(k, \omega)$ vanishes. This amounts to a minimisation problem. The simplest way to solve this is to use the Newton method with a good initial guess. We obtain the following values $\omega$ for different $k$:

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1.4156 - 0.1533i</td>
</tr>
<tr>
<td>0.4</td>
<td>1.2850 - 0.0661i</td>
</tr>
<tr>
<td>0.3</td>
<td>1.1598 - 0.0126i</td>
</tr>
<tr>
<td>0.2</td>
<td>1.0640 - 5.510 \times 10^{-5}i</td>
</tr>
</tbody>
</table>

These values can be used to verify your VP implementation by initialising $f_0$ with one or several plane waves in $x$ with wave vector $k$ and then checking whether the correct frequency is obtained from the simulation. The amplitudes of the initial perturbation must be small such that at the linear regime assumed in the derivation of the dispersion relation is valid.

### 3.2 Operator splitting

In the Vlasov equation (3.2) the advection field in $x$, which is $v$, does not depend on $x$ and the advection field in $v$, which is $-E(x, t)$, does not depend on $v$. Therefore it is often convenient to decompose these two parts, using the technique called operator splitting.

Let us consider the Vlasov-Poisson system (3.1). We shall split the Vlasov equation into the following two pieces:

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = 0, \quad \text{with } v \text{ fixed},$$  \hspace{1cm} (3.8)

and

$$\frac{\partial f}{\partial t} - E(x, t) \cdot \nabla_v f = 0, \quad \text{with } x \text{ fixed}.$$  \hspace{1cm} (3.9)

We then get two constant coefficient advection equations that can be easily solved with high-order methods (in particular Fourier methods). This is obvious for (3.8) as $v$ does not depend on $t$ and $x$. On the other hand, integrating (3.9)
with respect to \( v \), we get that \( \partial_t n = 0 \), so that the charge density \( \rho = 1 - n \) and consequently \( E \) do not change when this equation is advanced in time. Therefore, \( E(t, x) \) needs to be computed once from the initial \( f \) for this equation and depends neither on \( t \) nor on \( v \).

**Remark 3.** When the starting equation has some features which are important for the quality of the numerical solution, it is essential not to remove them when doing operator splitting. In particular, if the initial equation is conservative, it is generally a good idea to split such that each of the split equation is conservative.

In order to analyze the error resulting from operator splitting, let us consider equations of the form

\[
\frac{\partial f}{\partial t} = (A + B)f , \tag{3.10}
\]

where \( f = f(t, x) \) \( A \) and \( B \) are any two differential operators in \( x \) that are assumed constant in time. The formal solution of this equation after time \( \Delta t \) reads

\[
f(t + \Delta t) = e^{\Delta t(A+B)}f(t).
\]

Let us split the equation (3.10) into

\[
\frac{\partial f}{\partial t} = Af , \tag{3.11}
\]

\[
\frac{\partial f}{\partial t} = Bf . \tag{3.12}
\]

The formal solutions of these equations taken separately are

\[
f(t + \Delta t) = e^{\Delta tA}f(t) \quad \text{and} \quad f(t + \Delta t) = e^{\Delta tB}f(t).
\]

The **Lie-Trotter splitting** method consists in solving successively on one time step first (3.11) and then (3.12). Then after one time step we get

\[
\tilde{f}(t + \Delta t) = e^{\Delta tB}e^{\Delta tA}f(t).
\]

If the operators \( A \) and \( B \) commute, i.e. \( AB = BA \), \( e^{\Delta tB}e^{\Delta tA} = e^{\Delta t(A+B)} \) and the splitting is exact. This is the case in particular when \( A, B \) do not depend on \( x \), as for instance in

\[
\frac{\partial f}{\partial t} + c_1 \frac{\partial f}{\partial x} + c_2 \frac{\partial f}{\partial y} = 0,
\]

with \( c_1 = \text{const.} \) and \( c_2 = \text{const.} \). Note that such an equation is also a good first test case to validate a Vlasov code. In the Vlasov equation we have

\[
v \cdot \nabla_x (E \cdot \nabla_v)f \neq E \cdot \nabla_v (v \cdot \nabla_x)f ,
\]

hence the operators do not commute and one introduces a splitting error.

In the case when \( A \) and \( B \) do not commute, the splitting error can be decreased by solving first (3.11) on a half time step, and then (3.12) on a full time step and
again (3.11) on a half time step. This method is known as the **Strang splitting method**. It corresponds to the formal solution

\[
\hat{f}(t + \Delta t) = e^{\frac{\Delta t}{2}A} e^{\Delta t B} e^{\frac{\Delta t}{2}A} f(t).
\]

The error committed at each time step by the operator splitting method when the operators do not commute is given by

**Proposition 6.** The Lie-Trotter splitting method has a global error in time of \(O(\Delta t)\) and the Strang splitting method has a global error in time of \(O(\Delta t^2)\).

**Proof.** In order to find the error we need to expand the matrix exponential. On the one hand we have

\[
e^{\Delta t(A + B)} = I + \Delta t(A + B) + \frac{\Delta t^2}{2} (A + B)^2 + O(\Delta t^3),
\]

and on the other hand

\[
e^{\Delta tB} e^{\Delta tA} = \left[ I + \Delta tB + \frac{\Delta t^2}{2} B^2 + O(\Delta t^3) \right] \left[ I + \Delta tA + \frac{\Delta t^2}{2} A^2 + O(\Delta t^3) \right]
\]

\[= I + \Delta t(A + B) + \frac{\Delta t^2}{2} (A^2 + B^2 + 2BA) + O(\Delta t^3).
\]

But as \(A\) and \(B\) do not commute, we have \((A + B)^2 = A^2 + AB + BA + B^2\). It follows that the local error is \(e^{\Delta t(A+B)} - e^{\Delta tB} e^{\Delta tA} = O(\Delta t^2)\), which leads to a global error of order \(O(\Delta t)\) after summation of the local error over all time steps.

For the Strang splitting method, we have

\[
e^{\frac{\Delta t}{2}A} e^{\Delta tB} e^{\frac{\Delta t}{2}A} = \left[ I + \frac{\Delta t}{2}A + \frac{\Delta t^2}{8} A^2 + O(\Delta t^3) \right] \left[ I + \Delta tB + \frac{\Delta t^2}{2} B^2 + O(\Delta t^3) \right]
\]

\[= I + \Delta t(A + B) + \frac{\Delta t^2}{2} (A^2 + B^2 + 2BA + AB) + O(\Delta t^3).
\]

We thus obtain a local error of \(O(\Delta t^3)\) and thus a global error of \(O(\Delta t^2)\) for the method of Strang.

**Remark 4.** It is possible to obtain splitting methods of order as high as desired by taking adequate compositions of the two operators. Details on high order splitting methods can be found in [17].

**Remark 5.** The Strang splitting method can also be generalized to more than two operators. If \(A = A_1 + \cdots + A_n\), the following decomposition will be of global order \(O(\Delta t^2)\):

\[
e^{\frac{\Delta t}{2}A_1} \cdots e^{\frac{\Delta t}{2}A_{n-1}} e^{\Delta t A_n} e^{\frac{\Delta t}{2}A_{n-1}} \cdots e^{\frac{\Delta t}{2}A_1}.
\]
3.3 The particle in cell (PIC) method

Due to its simplicity and its efficiency in high dimensions, a popular method for kinetic plasma models is the Particle In Cell (PIC) method. It consists of drawing randomly a finite number \( K \gg 1 \) of initial conditions for the characteristic equations (3.3) and follow them in time by solving the equations by some numerical method. The distribution function is then approximated by a sum of Dirac masses

\[
f(t, x, v) \approx f_h(t, x, v) = \sum_{k=0}^{K-1} w_k \delta(x - x_k(t)) \delta(v - v_k(t)). 
\]  

(3.13)

Here, \( w_k = \text{const.} \) stand for time-independent ”weights” associated to each ”particle” (or marker) indexed by \( k \). The meaning of the weights will become clear in the section “Monte-Carlo Simulation”. The difficult part of the PIC method is the coupling between the particles and the fields. In the Vlasov-Poisson system for instance, the charge density \( \rho = 1 - n \), source of the Poisson equation, needs to be computed from the particle positions \( x_k(t) \). On the other hand, in the characteristic equations the electric field \( E(x_k(t)) \) must be evaluated at the particle positions. In Finite Element methods, the basis function provide a natural way to express the electric fields everywhere in space and, moreover, the weak formulation of Poisson is compatible with the expression of the distribution function as a sum of Dirac masses.

3.3.1 Time scheme for the particles

PIC solvers are initialized by randomly drawing a set of \( K \) initial conditions, or origins, \((x_{k0}, v_{k0}) \in \mathbb{R}^6, k = 0, \ldots , K - 1\), for the characteristics of the Vlasov equation. The so-called ”markers” \((x_k(t), v_k(t))\) then satisfy

\[
\begin{align*}
\frac{dx_k}{dt} &= v_k(t), & x_k(0) &= x_{k0}, \\
\frac{dv_k}{dt} &= -E(t, x_k(t)), & v_k(0) &= v_{k0}.
\end{align*}
\]  

(3.14)

A vital part for a PIC scheme is a good integrator for this system of ODEs. The system being Hamiltonian, it should be solved using a symplectic time scheme in order to enjoy long time conservation properties. The scheme which is used most of the time is the Verlet scheme, which is defined as follows: assume the marker positions \( x^n_k \), velocities \( v^n_k \) and the electric \( E^n \) at time \( t_n = n\Delta t \) to be known, where \( \Delta t \) denotes the time step size, then we advance in time as

\[
\begin{align*}
v^{n+\frac{1}{2}}_k &= v^n_k - \frac{\Delta t}{2} E^n(x^n_k), \\
x^{n+1}_k &= x^n_k + \Delta t v^{n+\frac{1}{2}}_k, \\
v^{n+1}_k &= v^{n+\frac{1}{2}}_k - \frac{\Delta t}{2} E^{n+1}(x^{n+1}_k).
\end{align*}
\]  

(3.15)
We notice that step (3.16) needs the electric field at time $t_{n+1}$. It can be computed after step (3.15) by solving the Poisson equation which uses as input the density $n_{n+1}$ that needs only $x_{k}^{n+1}$ (and not the velocities $v_{k}^{n+1}$).

3.3.2 Particle mesh coupling for Finite Elements

The coupling between mesh and particles is obtained in a natural way in the Finite Element method. Suppose that $(Λ_j)_{j=0}^{N-1}$ is a basis of the FEM space $V_h$ in which the approximation $φ_h$ of the electrostatic potential $φ$ is searched, then

$$φ(t, x) \approx φ_h(t, x) = \sum_{j=0}^{N-1} φ_j(t) Λ_j(x) \in V_h.$$  

At least locally on each cell the gradient of $φ_h$ is well defined and so the electric field at a particle position is directly approximated by

$$E(t, x_k) \approx E_h(t, x_k) = - \sum_{j=0}^{N-1} φ_j(t) \nabla Λ_j(x_k). \quad (3.17)$$

On the other hand, using the approximation (3.13) for $f$ as a sum of Dirac masses, the weak form of the Poisson equation reads

$$\int \nabla φ_h \cdot \nabla ψ_h \, dx = \int \left[ 1 - \int f_h(t, x, v) \, dv \right] ψ_h \, dx$$

$$= \int ψ_h \, dx - \sum_{k=0}^{K-1} w_k ψ_h(x_k), \quad \forall ψ_h \in V_h. \quad (3.18)$$

3.3.3 Time loop

Let us now summarize the main stages to go from time $t_n$ to time $t_{n+1}$ in a FE-based PIC solver:

1. Perform the first two steps of the Verlet scheme (3.14)-(3.15) to obtain $x_{k}^{n+1}$.
2. From the new marker positions $x_{k}^{n+1}$, compute the charge density, that is the right-hand side of (3.18).
3. Solve (3.18) with the FE method (inversion of the stiffness matrix) to obtain $φ_{n+1}$ and thus $E_{n+1}$ via (3.17).
4. Evaluate the new electric field field at the particle positions and complete the Verlet scheme by updating (3.16) to obtain $v_{k}^{n+1}$.  

64
3.3.4 Conservation properties at the semi-discrete level

Conservation of mass. The discrete mass is defined as

\[ N_h(t) := \int f_h(x, v, t) \, dx \, dv = \sum_k w_k. \]

This is obviously conserved if no particle gets in or out of the domain, as the weight \( w_k \) is conserved for each particle when the particles move.

Momentum. The total momentum of the system of particles is defined as

\[ P_h(t) := \int v f_h(x, v, t) \, dx \, dv = \sum_k w_k v_k(t). \]

So

\[ \frac{dP_h}{dt} = \sum_k w_k \frac{dv_k}{dt} = - \sum_k w_k E_h(x_k, t). \]

In FE-based PIC methods as the one above, the total momentum is NOT conserved. However, it is conserved for point-based PIC methods, where the Poisson equation is solved with a finite differences or Fourier collocation method. By contrast, such methods do not conserve the total energy, but the FE-based method does.

Remark 6. Note that the conservation of momentum is linked to the self-force problem that is often mentioned in the PIC literature. Indeed if the system is reduced to one particle. The conservation of momentum is equivalent to the fact that a particle does not apply a force on itself.

Conservation of energy. Classical point-based solvers based on finite difference or spectral methods do not conserve total energy, but the semi-discrete Finite Element solver does. Indeed, consider the equations of motion for the particles

\[ \frac{dx_k}{dt} = v_k, \quad \frac{dv_k}{dt} = \nabla \phi_h(t, x_k), \]

coupled with the finite element discretisation of the Poisson equation (3.18),

\[ \int \nabla \phi_h \cdot \nabla \psi_h \, dx = \int \psi_h \, dx - \sum_k w_k \psi_h(x_k) \quad \forall \psi_h \in V_h. \]

Then the following semi-discrete energy is exactly conserved:

\[ E_h(t) := \sum_k \frac{w_k}{2} |v_k|^2 + \frac{1}{2} \int |\nabla \phi_h|^2 \, dx. \]
Let us verify this by direct computation. First taking $\psi_h = \phi_h$ as a test function, using that $\phi_h$ has zero mean, and taking the time derivative, the weak Poisson equation yields

$$\frac{d}{dt} \int |\nabla \phi_h|^2 \, dx = - \sum_k w_k \left( \frac{\partial \phi_h}{\partial t}(t, x_k) + \frac{dx_k}{dt} \cdot \nabla \phi_h(t, x_k) \right). \tag{3.19}$$

On the other hand taking $\psi_h = \partial_t \phi_h(t, x_k)$ in the weak Poisson equation, we also have that

$$- \sum_k w_k \frac{\partial \phi_h}{\partial t}(t, x_k) = \int \nabla \phi_h \cdot \nabla \frac{\partial \phi_h}{\partial t} \, dx = \frac{1}{2} \frac{d}{dt} \int |\nabla \phi_h|^2 \, dx,$$

so that equation (3.19) becomes

$$\frac{1}{2} \frac{d}{dt} \int |\nabla \phi_h|^2 \, dx = - \sum_k w_k \frac{dx_k}{dt} \cdot \nabla \phi_h(t, x_k) = - \sum_k w_k v_k \cdot \nabla \phi_h(t, x_k).$$

Now using this, we find

$$\frac{dE_h(t)}{dt} = \sum_k \left( w_k v_k \cdot \frac{dv_k}{dt} - w_k v_k \cdot \nabla \phi_h(t, x_k) \right) = 0$$

as

$$\frac{dv_k}{dt} = - \nabla \phi_h(t, x_k).$$

### 3.4 Monte Carlo simulation

There are two numerical errors committed in a PIC method:

1. The error from the discretisation of the field equations (Poisson or Maxwell).

2. The **statistical error** coming from the approximation of the Vlasov-distribution $f$ by a finite number of point particles.

We already analysed the first error. In order to get an idea of the second, it has been very fruitful to interpret the FE source terms

$$\int f \underbrace{\psi_i(x)}_{\text{test func.}} \, dx \, dv \quad \text{(density)}, \quad \int v f \underbrace{\psi_i(x)}_{\text{test func.}} \, dx \, dv \quad \text{(current)} \tag{3.20}$$

as Monte Carlo (MC) integrals [1]. The basic idea of MC simulation is to use concepts from probability theory to evaluate integrals. An integral is viewed as the expected value of a random variable distributed according to a certain probability law, which is then approximated by a large number of random events. In this section we introduce the basic notions needed in MC simulations.
3.4.1 Basics of probability theory

The proper mathematical framework for probability theory starts with the construction of a probability space as a measurable set using \( \sigma \)-algebras. We refrain from using these abstract objects and consider only probabilities in \( \mathbb{R}^n \).

The main objects we consider are random variables. A random variable will be denoted with capital letters; it can be viewed as a pair \( \mathbf{X} = (\mathbb{R}^n, g) \), where \( g : \mathbb{R}^n \to \mathbb{R}_+ \) is a positive function called the probability distribution (PDF) of \( \mathbf{X} \). The PDF is normalized

\[
\int_{\mathbb{R}^n} g(x) \, d^n x = 1, \tag{3.21}
\]

and one says \( \mathbf{X} \) is distributed according to \( g \). A point \( x \in \mathbb{R}^n \) is called a sample point and open subsets \( A \subset \mathbb{R}^n \) are called events. The probability \( P_X(A) \) of an event \( A \) to occur is defined by

\[
P_X(A) = P(\mathbf{X} \in A) := \int_A g(x) \, d^n x.
\]

Clearly, one has \( P_X(\mathbb{R}^n) = 1 \) due to (3.21). To each random variable one can associate its expected value \( E(\mathbf{X}) \), defined by

\[
E(\mathbf{X}) := \int_{\mathbb{R}^n} x \, g(x) \, d^n x,
\]

provided the integral exists (we shall assume that all integrals we write in the following exist). Moreover, for (possibly vector-valued) functions \( h : \mathbb{R}^n \to \mathbb{R}^d \) with \( d \geq 1 \) one has

\[
E(h(\mathbf{X})) = \int_{\mathbb{R}^n} h(x) g(x) \, d^n x. \tag{3.22}
\]

Other important parameters of a random variable \( \mathbf{X} \) are its variance \( \mathbb{V}(\mathbf{X}) \) and its standard deviation \( \sigma(\mathbf{X}) \), defined by

\[
\mathbb{V}(\mathbf{X}) := E(|\mathbf{X} - E(\mathbf{X})|^2),
\]

\[
\sigma(\mathbf{X}) := \sqrt{\mathbb{V}(\mathbf{X})}.
\]

Using (3.22) we can rewrite the variance as

\[
\mathbb{V}(\mathbf{X}) = E(|\mathbf{X} - E(\mathbf{X})|^2)
\]

\[
= \int_{\mathbb{R}^n} |\mathbf{x} - E(\mathbf{X})|^2 g(x) \, d^n x
\]

\[
= \int_{\mathbb{R}^n} (|\mathbf{x}|^2 - 2\mathbf{x} \cdot E(\mathbf{X}) + |E(\mathbf{X})|^2) \, g(x) \, d^n x
\]

\[
= E(|\mathbf{X}|^2) - |E(\mathbf{X})|^2.
\]

From this we deduce that \( \mathbb{V}(h(\mathbf{X})) \) is also well-defined.
We shall frequently use the term independent random variables. Roughly speaking, this means that the probability for an event of a random variable $X_1$ is unrelated to events of a second random variable $X_2$. More precisely, let $X_1 = (R^l, g_1)$ and $X_2 = (R^m, g_2)$, then we denote by $X = (R^{l+m}, g)$ a new random variable with probability

$$P_X(A) = P(X_1 \in A_1, X_2 \in A_2) = \int_A g(x_1, x_2) \, d^{l+m}x,$$

where $A = A_1 \times A_2$. The random variables $X_1$ and $X_2$ are said to be independent if

$$P(X_1 \in A_1, X_2 \in A_2) = P(X_1 \in A_1) \, P(X_2 \in A_2),$$

or, in terms of PDFs, $g(x_1, x_2) = g_1(x_1)g_2(x_2)$. For independent real-valued random variables $X_1, \ldots, X_n$ one has

$$\mathbb{E}(X_1 \ldots X_n) = \mathbb{E}(X_1) \ldots \mathbb{E}(X_n),$$

$$\mathbb{V}(X_1 + \ldots + X_n) = \mathbb{V}(X_1) + \ldots + \mathbb{V}(X_n) \quad \text{(Bienaymé’s theorem)} \quad (3.23)$$

### 3.4.2 Estimators

In practice sometimes the PDF of a random variable of interest is not known exactly; observables which are functions of this random variable must then be estimated from sample data. The correct way to do this in statistics is via estimators, which are themselves functions of random variables.

Suppose that $X = (R^n, g)$ is a random variable of which $g$ is not known. Further, let $h : R^n \to R^d$, $d \geq 1$ stand for a (possibly vector-valued) function and suppose an observable of interest $\theta$ is given by

$$\theta_E = \mathbb{E}(h(X)). \quad (3.24)$$

In order to approximate $\theta_E$, let $(X_i)_{i=0 \ldots N-1}$ stand for a vector of $N$ independent random variables $X_i = (R^n, g)$, thus distributed according to $g$, and define the sample mean

$$\bar{\theta}_E := \frac{1}{N} \sum_{i=0}^{N-1} h(X_i). \quad (3.25)$$

This is a function of the random variables $(X_i)_i$ and an estimator for the expected value $(3.24)$. The bias of an estimator $\bar{\theta}$ is defined as $\text{bias}(\bar{\theta}) := \theta - \mathbb{E}(\bar{\theta})$. An estimator is called unbiased if its bias is zero, which is clearly the case for the sample mean,

$$\mathbb{E}(\bar{\theta}_E) = \frac{1}{N} \sum_{i=0}^{N-1} \mathbb{E}(h(X_i)) = \frac{1}{N} \sum_{i=0}^{N-1} \mathbb{E}(h(X)) = \mathbb{E}(h(X)) = \theta_E,$$
since the $X_i$ are distributed like $X$. Using Bienaymé’s theorem (3.23) one can show that an unbiased estimator for the variance

$$\theta_V = \mathbb{V}(h(X))$$

is given by

$$\bar{\theta}_V := \frac{1}{N-1} \sum_{i=0}^{N-1} |h(X) - \bar{\theta}_E|^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} \left| h(X) - \frac{1}{N} \sum_{i=0}^{N-1} h(X_i) \right|^2.$$  

Remark that using $1/N$ in front of the sum would also yield an estimator for the variance, but this estimator would be biased.

The mean square error (MSE) of an estimator $\bar{\theta}$ approximating an observable $\theta$ reads

$$MSE(\bar{\theta}) := E(\bar{\theta} - \theta)^2.$$  

A straightforward computation yields

$$MSE(\bar{\theta}) = E(|\bar{\theta} - \theta|^2) = E(|\bar{\theta}|^2) - 2E(\bar{\theta}) \cdot \theta + |\theta|^2
= E(|\bar{\theta}|^2) - |E(\bar{\theta})|^2 + |E(\bar{\theta})|^2 - 2E(\bar{\theta}) \cdot \theta + |\theta|^2
= \mathbb{V}(\bar{\theta}) + |\theta - E(\bar{\theta})|^2
= \mathbb{V}(\bar{\theta}) + |bias(\bar{\theta})|^2.$$  

The root mean square error (RMS) of an estimator is the square root of the MSE. For an unbiased estimator, it follows that the RMS is the standard deviation $RMS(\bar{\theta}) = \sigma(\bar{\theta})$. For the sample mean (3.25) we can use Bienaymé’s theorem (3.23) to compute

$$\mathbb{V}(\bar{\theta}_E) = \mathbb{V}\left( \frac{1}{N} \sum_{i=0}^{N-1} h(X_i) \right) = \frac{1}{N^2} \left( \sum_{i=0}^{N-1} \mathbb{V}(h(X_i)) \right) = \frac{1}{N} \mathbb{V}(\theta_V),$$

since the $(X_i)_i$ are distributed like $X$. We conclude that

$$RMS(\bar{\theta}_E) = \frac{\sigma(\theta_V)}{\sqrt{N}}.$$  

This is the statistical error found in MC simulations when working with the sample mean as an estimator for observables. Two things are noteworthy: the convergence rate with $1/\sqrt{N}$ and the fact that the error is proportional to the standard deviation of the observable $\theta_V$. Choosing thus in a clever way the observable $\theta_V$, which amounts to choosing the function $h$ in order to minimize $\sigma(\theta_V)$, can lead to a reduced error in the simulations - this is the purpose of the control variates (“$\delta f$-methods”).

69
3.4.3 PIC Monte Carlo

Let us now come back to the computation of the integrals (3.20) relevant for FE-based PIC methods in plasma physics. Let us define a time-dependent PDF $s : \mathbb{R} \times \mathbb{R}^6 \to \mathbb{R}_+$ on the phase space $\mathbb{R}^6$ which solves the Vlasov equation (just as $f$, which is however not a PDF):

$$\frac{\partial s}{\partial t} + v \cdot \nabla_x s - E \cdot \nabla_v s = 0, \quad s(t=0,x,v) = s_0(x,v).$$

(3.26)

The function $s$ is called the **marker distribution** and its initial condition $s_0$ will be referred to as **sampling distribution**. This is for good reason, as in the PIC method, the markers (or particles) will be drawn according to $s_0$. Therefore, to initialise a PIC solver we draw $K$ samples $(x_k, v_k)_{k=0}^{N-1}$ according to the PDF $s_0(x,v)$. In the Poisson equation, the particle density is then re-written as the expectation value of the following random variable:

$$\int f \psi_i \, dx \, dv = \int f \psi_i \, s \, dx \, dv = \mathbb{E} \left[ \frac{f}{s} \psi_i \right].$$

(3.27)

We do not know $s$ exactly at each point in time, but we do know $K$ samples $(x_k(t), v_k(t))_{k=0}^{N-1}$ in time because we solve the characteristic equations of (3.26) for each marker. Thus, we can write an estimator for the expectation value,

$$\mathbb{E} \left[ \frac{f}{s} \psi_i \right] \approx \frac{1}{K} \sum_{k=0}^{K-1} f(t, x_k(t), v_k(t)) \frac{s(t, x_k(t), v_k(t)) \psi_i(x_k(t))}{w_k} =: \theta_i,$$

for each test function $\psi_i$. Moreover, since both $f$ and $s$ are constant along the characteristics, we have

$$\frac{f(t, x_k(t), v_k(t))}{s(t, x_k(t), v_k(t))} = \frac{f_0(x_{k_0}, v_{k_0})}{s_0(x_{k_0}, v_{k_0})} =: w_k = const.,$$

which explains the constant weights $w_k$ and leads us back to (3.18). Note here that $f_0$ is the ”physical” initial condition of the PDE problem at hand, whereas $s_0$ is completely arbitrary. All the information about the initial state of the system is thus encoded in the weights. The proper choice of $s_0$ can lead to more efficient simulations, using less particles; this technique is called **importance sampling** (more samples drawn in the important regions of phase space).

Likewise, for the current density in Ampère’s law we would approximate

$$\int v \, f \psi_i \, dx \, dv = \mathbb{E} \left[ \frac{f}{s} V \psi_i \right] \approx \frac{1}{K} \sum_{k=0}^{K-1} w_k v_k(t) \psi_i(x_k(t)).$$

3.4.4 Control variates (”$\delta f$-method”)

A very popular and fruitful method for **noise reduction** in PIC methods is the control variate method, also referred to as the ”$\delta f$-method”. This method works
well if \( f \) is close to an "equilibrium state" denoted by \( \mathcal{M} \), which is known a priori. Since we know \( \mathcal{M} \) we can compute the following integrals exactly (without statistical error):

\[
n_{\mathcal{M}} := \int \mathcal{M} \psi_i \, dx \, dv, \quad j_{\mathcal{M}} := \int \mathbf{v} \mathcal{M} \psi_i \, dx \, dv.
\]

The control variate is a distribution function introduced as \( \delta f := f - \mathcal{M} \). The statistical noise of \( \delta f \) is greatly reduced compared to "full"-\( f \) as long as \( f \) is close to \( \mathcal{M} \) (we leave the statement without proof). The price to pay is that the weights become time-dependent, which has implications on certain conservation properties of PIC schemes (exact energy conservation is destroyed for example). The expectation value (3.27) is altered in the following way when using \( \delta f \):

\[
\int f \psi_i \, dx \, dv = \int (f - \mathcal{M}) \psi_i \, dx \, dv + \int \mathcal{M} \psi_i \, dx \, dv
\]

\[
= \int \frac{\delta f}{s} \psi_i \, dx \, dv + n_{\mathcal{M}} \quad \text{(3.28)}
\]

\[
= \mathbb{E} \left[ \frac{\delta f}{s} \psi_i \right] + n_{\mathcal{M}}.
\]

As \( \delta f \) is not constant along the characteristics, the weights \( \delta f/s \) are not constant in time. Rather, we have

\[
\mathbb{E} \left[ \frac{\delta f}{s} \psi_i \right] \approx \frac{1}{K} \sum_{k=0}^{K-1} \left[ w_k - \frac{\mathcal{M}(t, x_k(t), v_k(t))}{s_0(x_{k0}, v_{k0})} \right] \psi_i(x_k(t)) =: \bar{\theta}_{i}^{CV}.
\]

As most of the information of the integral (3.28) is in the exact value \( n_{\mathcal{M}} \), the variance of the estimator \( \bar{\theta}_{i}^{CV} \) is greatly reduced compared to \( \bar{\theta}_{i} \) without control variate. The same can be applied to the current density and yields

\[
\int \mathbf{v} f \psi_i \, dx \, dv = \mathbb{E} \left[ \frac{\delta f}{s} \mathbf{V} \psi_i \right] + j_{\mathcal{M}}
\]

\[
\approx \frac{1}{K} \sum_{k=0}^{K-1} \left[ w_k - \frac{\mathcal{M}(t, x_k(t), v_k(t))}{s_0(x_{k0}, v_{k0})} \right] \mathbf{v}_k(t) \psi_i(x_k(t)) + j_{\mathcal{M}}.
\]
Chapter 4

Finite volume methods for plasma fluid models

4.1 Hyperbolic systems of conservation laws

Numerical methods are designed for and adapted to the different classes of PDE’s:

- Elliptic PDE’s are in general steady state equations which satisfy some coercivity property, the prototype of which is the Poisson equation

\[-\Delta \phi = f\]

- Parabolic equation are first order in time, with an elliptic differential operator in space, the prototype is the heat equation

\[\frac{\partial u}{\partial t} - \Delta u = 0.\]

- Hyperbolic equation are either first order transport equation, with advection as a prototype

\[\frac{\partial u}{\partial t} + a \cdot \nabla u = 0,\]

or second order PDEs with the wave equation as a prototype

\[\frac{\partial^2 u}{\partial t^2} - \Delta u = 0.\]

The Euler system is a first order non linear transport equation, which falls into the important category of **hyperbolic systems of conservation laws**, which on a 1D domain have the abstract form

\[\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0,\]
where $u(t) : [0, L] \to \mathbb{R}^n$ is a vector of unknown values and $F : \mathbb{R}^n \to \mathbb{R}^n$ is the flux function. Supposing that nothing flows in or out at the boundaries, we have

$$\partial_t \int_0^L u(t, x) \, dx = F(u(t, L)) - F(u(t, 0)) = 0,$$

hence the term "conservation law". The conservation law can be written also

$$\frac{\partial u}{\partial t} + A(u) \frac{\partial u}{\partial x} = 0,$$

where $A(u)$ is the Jacobian matrix with components $((\frac{\partial F}{\partial u})_{i,j})$. The system is called hyperbolic if for all $u$ the matrix $A$ has only real eigenvalues and is diagonalisable. It is called strictly hyperbolic if all eigenvalues are distinct. Some of the most important plasma models fall into the class of hyperbolic systems, such as the Euler equations and the ideal MHD equations.

### 4.2 Euler-Poisson equations

In $n$ spatial dimensions, the Vlasov equation is a $2n$-dimensional problem ($2n + 1$ if one includes time), which gives the detailed velocity distribution of the particles at each point $x \in \mathbb{R}^n$, making it computationally very expensive. To reduce the complexities, one can model the plasma by fluid equations which give the information about some macroscopic quantities. The underlying assumption for such models is that due to collisions, the system is close to thermodynamic equilibrium at all times. The fluid model is an $n$-dimensional model ($n + 1$ if one includes time), hence computationally less expensive.

#### 4.2.1 Velocity moments of Vlasov

Fluid equations can be derived from the Vlasov equation by taking velocity moments (=integration in $v$). Let us consider the 1D electron Vlasov equation

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - E \frac{\partial f}{\partial v} = 0,$$ \hspace{1cm} (4.1)

and use the following definitions:

$$n(t, x) := \int_{\mathbb{R}} f(t, x, v) \, dv,$$ \hspace{1cm} (4.2a)

$$n(t, x) u(t, x) := \int_{\mathbb{R}} v f(t, x, v) \, dv,$$ \hspace{1cm} (4.2b)

$$w(t, x) := \frac{1}{2} \int_{\mathbb{R}} v^2 f(t, x, v) \, dv,$$

$$p(t, x) := \gamma - \frac{1}{2} \int_{\mathbb{R}} f(v - u)^2 \, dv = (\gamma - 1)(w - nu^2/2).$$ \hspace{1cm} (4.2c)
Here, \( n \) defines the density, \( u \) the plasma mean velocity, \( w \) the energy, \( p \) the pressure, and \( \gamma \) stands for the heat capacity ratio of the plasma. The heat capacity ratio is related to the degrees of freedom \( m \) of one particle via \( \gamma = 1 + 2/m \). Hence for a mono-atomic, ideal gas in one dimension (no rotation, one direction of motion) we have \( m = 1 \) and the heat capacity ratio is \( \gamma = 3 \). Then taking the zeroth moment of the Vlasov equation (4.1) gives:

\[
\int_{\mathbb{R}} \partial_t f \, dv + \int_{\mathbb{R}} v \partial_x f \, dv - \int_{\mathbb{R}} E \partial_v f \, dv = 0.
\]

Using that the last term vanishes due to the decay condition on \( f \),

\[
\int_{\mathbb{R}} E \partial_v f \, dv = E f(v)|^{\infty}_{-\infty} = E (f(\infty) - f(-\infty)) = 0,
\]

and inserting the definitions (4.2a) and (4.2b), one obtains

\[ \partial_t n + \partial_x (n u) = 0, \]

which is the conservation of mass equation. For the momentum equation, multiplying the Vlasov equation with \( v \) and integrating yields

\[
\partial_t \int_{\mathbb{R}} v f \, dv + \partial_x \int_{\mathbb{R}} v^2 f \, dv - E \int_{\mathbb{R}} v \partial_v f \, dv = 0. \tag{4.3}
\]

Since

\[
E \int_{\mathbb{R}} v \partial_v f \, dv = E \left( v f \big|^{\infty}_{-\infty} - \int_{\mathbb{R}} f \partial_v v \, dv \right) = -nE,
\]

by substituting (4.2b)-(4.2c) in equation (4.3) we obtain

\[ \partial_t (n u) + \partial_x (n u^2 + p) = -nE, \]

which is the conservation of momentum equation. For the energy equation we multiply the Vlasov equation by \( v^2/2 \) and integrate to obtain

\[
\partial_t \frac{1}{2} \int_{\mathbb{R}} v^2 f \, dv + \partial_x \frac{1}{2} \int_{\mathbb{R}} v^3 f \, dv - \frac{1}{2} E \int_{\mathbb{R}} v^2 \partial_v f \, dv = 0. \tag{4.4}
\]

At this point we define the heat flux \( q \) as

\[ q(t, x) := \frac{1}{2} \int_{\mathbb{R}} (v - u(t, x))^3 f(t, x, v) \, dv. \]

We make the following observation:

\[
q = \frac{1}{2} \left[ \int_{\mathbb{R}} (v^3 - u^3 - 3v^2 u + 3vu^2) \, f \, dv \right]
= \frac{1}{2} \left[ \int_{\mathbb{R}} v^3 f \, dv - u^3 \int_{\mathbb{R}} f \, dv - 3u \int_{\mathbb{R}} v^2 f \, dv + 3u^2 \int_{\mathbb{R}} v f \, dv \right]
= \frac{1}{2} \left[ \int_{\mathbb{R}} v^3 f \, dv - u^3 n - 6u w + 3u^2 nu \right]
= \frac{1}{2} \int_{\mathbb{R}} v^3 f \, dv - u(3w - n u^2). \tag{4.5}
\]
Moreover,
\[
\frac{1}{2} E \int_\mathbb{R} v^2 \partial_v f \, dv = \frac{1}{2} E \left( v^2 f \big|_\infty^- - \int_\mathbb{R} f \partial_v v^2 \, dv \right) = -nuE. \quad (4.6)
\]

Using (4.5) and (4.6), equation (4.4) becomes:
\[
\partial_t w + \partial_x \left[ u(3w - nu^2) \right] + \partial_x q = -nuE.
\]

With \( w = \frac{1}{2}p + \frac{1}{2}nu^2 \) this can also be written as:
\[
\partial_t w + \partial_x \left[ u(w + p) \right] + \partial_x q = -nuE,
\]

which is the conservation of energy equation. To summarize, the three lowest velocity moments of the Vlasov equation in (4.1) read
\[
\begin{cases}
\partial_t n + \partial_x (nu) = 0, \\
\partial_t (nu) + \partial_x (nu^2 + p) = -nE, \\
\partial_t w + \partial_x [u(w + p)] + \partial_x q = -nuE.
\end{cases} \quad (4.7)
\]

Remark that this system is not closed, because the heat flux \( q \) still depends on the full distribution function \( f \). However, if we assume \( f \) to be an even function of \( v - u \) at all times, then \( q = 0 \) and the Euler-Poisson system is closed. This is the case in thermal equilibrium, where \( f \) is a Gaussian function of \( v - u \), which is attained when particle collisions happen at a sufficient rate (high plasma density). For \( q = 0 \) the system (4.7), coupled to the Poisson equation for \( E \), is called the Euler-Poisson system.

### 4.2.2 Euler Equations

Setting \( E = 0 \) and \( q = 0 \) in (4.7), the velocity moments of the Vlasov equations reduce to the well-known Euler equations,
\[
\begin{align*}
\partial_t n + \partial_x (nu) &= 0, \\
\partial_t (nu) + \partial_x (nu^2 + p) &= 0, \\
\partial_t w + \partial_x [u(w + p)] &= 0.
\end{align*}
\]

This system has the form of a conservation law,
\[
\partial_t \mathbf{u} + \partial_x \mathbf{F}(\mathbf{u}) = 0, \quad (4.9)
\]

where we introduce the electron flux \( \Gamma := nu \) and write the flux function \( \mathbf{F} \) by using \( p = 2w - \Gamma^2/n \),
\[
\mathbf{u} = \begin{pmatrix} n \\ \Gamma \\ w \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \Gamma \\ 2w \\ 2w/n(3w - \Gamma^2/n) \end{pmatrix}.
\]
Let us now prove that (4.9) is indeed a hyperbolic system. By the chain rule one obtains
\[ \partial_t u + A(u) \partial_x u = 0, \]
where \( A := \frac{\partial F}{\partial u} = \left( \frac{\partial F_i}{\partial u_j} \right)_{i,j} \) is the Jacobian matrix which reads
\[
A(u) = \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 2 \\
-3w\Gamma/n^2 + 2\Gamma^3/n^3 & 3w/n - 3\Gamma^2/n^2 & 3\Gamma/n
\end{pmatrix}
\]
To compute the eigenvalues \( \lambda \) of \( A \), we need to solve the characteristic polynomial
\[
\det(\lambda I - A) = 0,
\]
\[
\begin{vmatrix}
\lambda & -1 & 0 \\
0 & \lambda & -2 \\
3uw/n - 2u^3 & -3w/n + 3u^2 & \lambda - 3u
\end{vmatrix}
= \lambda(\lambda^2 - 3\lambda - 6w/n + 6u^2) + 6uw/n - 4u^3
= \lambda(\lambda^2 - 2u\lambda + u^2 - u\lambda + u^2 - 6w/n + 4u^2) + u(6w/n - 4u^2)
= \lambda[(\lambda - u)^2 - u(\lambda - u)] - (\lambda - u)(6w/n - 4u^2)
= (\lambda - u)[\lambda(\lambda - u) - \lambda u - 6w/n + 4u^2]
= (\lambda - u)[\lambda(\lambda - u) - \lambda u - 3p/n + u^2]
= (\lambda - u)[(\lambda - u)^2 - 3p/n] = 0.
\]
Therefore, the three eigenvalues (characteristic velocities) of the Euler system read
\[ \lambda_{1,2,3} = (u, u + v_{th}, u - v_{th}), \]
where \( v_{th} = \sqrt{\gamma p/n} \) (with \( \gamma = 3 \)) is the thermal velocity. The eigenvalues are real because both the pressure \( p \) and the density \( n \) are positive.

4.3 The Finite Volume method

4.3.1 The first order Finite Volume schemes
Let us introduce the Finite Volume method on the generic scalar conservation law of the form
\[ \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0. \] (4.10)
In the case of our linear advection equation, we have \( f(u) = au \).
In the Finite Volume (FV) method, we start out with a grid \((x_i)\) on the simulation domain, just as for Finite Differences. Recall that for Finite Differences the unknowns were the point values of \(u\) at the grid points. In FV the unknown quantity that is numerically computed is the cell average of \(u\) on each cell \([x_i, x_{i+1}]\). In order to avoid confusion with the values of \(u\) at the grid points \(x_i\), we assign half integers to the cell averages of \(u\):

\[
u_{i+\frac{1}{2}}(t) = \frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} u(t, x) \, dx.
\]

The notation is depicted here:

![Diagram of cell averages and grid points]

The Finite Volume numerical scheme is then obtained by integrating the original equation (4.10) on each cell of the domain, which yields

\[
\frac{du_{i+\frac{1}{2}}(t)}{dt} + \frac{1}{\Delta x_{i+\frac{1}{2}}} (f(u(t, x_{i+1})) - f(u(t, x_i))) = 0. \tag{4.11}
\]

Here we see that a second ingredient is needed in order to define the algorithm. We only compute the cell averages \(u_{i+\frac{1}{2}}\) of \(u\), how do we define the values \(u_i\) of \(u\) at the cell interfaces, at which we evaluate the flux?

The simplest scheme, which is first order accurate in space, consists in assuming that \(u\) is constant on each cell and thus equal to its cell average. However, usually \(u_{i-\frac{1}{2}} \neq u_{i+\frac{1}{2}}\) for neighbouring cells such that \(u_i\) is not defined at the cell interface (there is a "jump"). In order to complete the FV scheme we need to define a so called numerical flux at each cell interface, denoted by \(g_i\), that needs to be consistent with the true \(f(u(x_i))\), i.e. \(g_i = f(u(x_i)) + O(\Delta x^p)\) for some positive \(p\). Hence, the exact equation (4.12) is approximated with the numerical fluxes,

\[
\frac{du_{i+\frac{1}{2}}(t)}{dt} + \frac{1}{\Delta x_{i+\frac{1}{2}}} (g_{i+1} - g_i) = 0. \tag{4.12}
\]

A numerical flux of order 2 is the centred flux \(g_i = \frac{1}{2} (f(u_{i-\frac{1}{2}}) + f(u_{i+\frac{1}{2}}))\). This yields the following scheme for a uniform \(\Delta x\):

\[
\frac{du_{i+\frac{1}{2}}(t)}{dt} + \frac{(f(u_{i+\frac{1}{2}}) - f(u_{i-\frac{1}{2}}))}{2\Delta x} = 0.
\]
Applying this to the linear advection \( f(u) = au \) and using an explicit Euler scheme in time yields

\[
 u^{n+1}_{i+\frac{1}{2}} = u^n_{i+\frac{1}{2}} - \frac{a\Delta t}{2\Delta x}(u^n_{i+\frac{3}{2}} - u^n_{i-\frac{1}{2}}).
\]

We recognise here the centred Finite Difference scheme shifted to the cell centres. This scheme is unstable, so that it cannot be used in practice. In order to get a stable scheme, we need to introduce the notion of upwinding. This can be done very easily in the definition of the numerical flux by simply choosing the value of \( u \) in the upwind cell only to define the numerical flux. We have

\[
 \frac{\partial f(u)}{\partial x} = f'(u)\frac{\partial u}{\partial x}.
\]

This means that locally at each cell interface the direction of the transport is defined by the sign of \( f'(u) \) (in the case of the linear advection \( f'(u) = a \) and the upwind direction is determined by the sign of \( a \)). So the upwind numerical flux is defined by

\[
 g_i = \begin{cases} 
 f(u_{i-\frac{1}{2}}) & \text{if } f'(\frac{u_{i-\frac{1}{2}}+u_{i+\frac{1}{2}}}{2}) \geq 0 \\
 f(u_{i+\frac{1}{2}}) & \text{if } f'(\frac{u_{i-\frac{1}{2}}+u_{i+\frac{1}{2}}}{2}) < 0
 \end{cases}
\]

Again, combining the Finite Volume scheme with an upwind flux and an explicit Euler time discretisation yields for the linear advection with \( a > 0 \)

\[
 u^{n+1}_{i+\frac{1}{2}} = u^n_{i+\frac{1}{2}} - \frac{a\Delta t}{\Delta x}(u^n_{i+\frac{3}{2}} - u^n_{i-\frac{1}{2}}).
\]

We also recognise here the first order in time and space upwind scheme shifted to the cell centres.

**Remark 7.** Using the midpoint rule

\[
 u_{i+\frac{1}{2}} = \frac{1}{\Delta x} \int_{x_i}^{x_{i+1}} u(x) \, dx = u(x_{i+\frac{1}{2}}) + O(\Delta x^2).
\]

Then we can reinterpret the Finite Volume as a Finite Difference scheme at the cell centres, which explains that we get the same formulas. However this is not true for higher orders, for which Finite Volume and Finite Difference schemes are genuinely different.

### 4.3.2 Higher order schemes

In order to get high order FV schemes, instead of assuming \( u \) being equal to its cell average \( u_{i+1/2} \) in the whole cell, the idea is to reconstruct polynomials of some given degree in each cell from the cell averages. The main idea for doing this is to construct an interpolation polynomial for the primitive of the polynomial we are looking for.

At time step \( t_n \), we know \( u^n_{j+\frac{1}{2}} \), the average value of \( u^n \) on cell \([x_j, x_{j+1}]\) of length \( \Delta x_{j+\frac{1}{2}} = x_{j+1} - x_j \). We want to construct a polynomial \( p_m(x) \) of degree \( m \) such
that
\[ \frac{1}{\Delta x_{j+\frac{1}{2}}} \int_{x_j}^{x_{j+1}} p_m(x) \, dx = u_{j+\frac{1}{2}}^n. \]

To this aim we look for the primitive \( \tilde{p}_m(x) \) such that \( \frac{d}{dx} \tilde{p}_m(x) = p_m(x) \). Then
\[ \Delta x_{j+\frac{1}{2}} u_{j+\frac{1}{2}}^n = \int_{x_j}^{x_{j+1}} p_m(x) \, dx = \tilde{p}_m(x_{j+1}) - \tilde{p}_m(x_j). \]

Then we take for \( \tilde{p}_m \) an interpolating polynomial at points \( x_j \) and solve the interpolation problem. There are many ways to choose an interpolating polynomial, one could use spline interpolation or Hermite interpolation, but the simplest and most used choice is to use a Lagrange interpolation. This being said, a Lagrange interpolating polynomial of degree \( k \) is defined with \( k + 1 \) interpolation points. So we need to use as many values in neighbouring cells as needed.

In order to reconstruct a polynomial of a given degree in a given cell there are many possible stencils, i.e. ensembles of cells, that can be used. For the reconstruction of a polynomial of degree \( k \) exactly \( k \) average values corresponding to \( k \) neighbouring cells are needed. The only constraint is that the value on the cell where the polynomial being reconstructed is used. High-order methods are prone to oscillations especially around discontinuities. So one good idea is to use the stencil which minimises the oscillations. This can be easily done by choosing automatically the stencil based on the Newton divided differences which can be used to construct the interpolating polynomial. This method is called ENO (Essentially Non Oscillatory). See for example [8] for a detailed description.

The ENO method can be still improved by taking all possible stencils but putting a weight on each of the polynomials obtained. This is called the WENO method (Weighted Essentially Non Oscillatory) A good review of this technique is given in [10].

4.3.3 Example: FV method for 1D Euler equations

Let us recall that the Euler equations can be written in the conservative form
\[ \partial_t u + \partial_x F(u) = 0, \tag{4.13} \]

with
\[ u = \begin{pmatrix} n \\ \Gamma \\ w \end{pmatrix}, \quad F = \begin{pmatrix} \Gamma \\ 2w \\ \Gamma/n(3w - \Gamma^2/n) \end{pmatrix}. \]

where \( F \) is the flux function. Our aim is to apply a FV method to (4.13), using the notation introduced before. Assuming uniform grid spacing \( \Delta x \), integration over the cells yields
\[ \frac{d}{dt} u_{j+1/2} + \frac{1}{\Delta x} [F(u(x_{j+1})) - F(u(t, x_j))] = 0. \]
Moreover, \( \lambda_k \) of degree \( u \) where

This is still an exact evolution equation for the cell averages. To apply the FV scheme, we approximate the flux function by a suitable numerical flux \( g_j \approx F(u(t, x_j)) \):

\[
\frac{d}{dt} u_{j+1/2} + \frac{1}{\Delta x} (g_{j+1} - g_j) = 0.
\]

We shall choose the **Lax-Friedrichs method** to approximate fluxes by

\[
g_{jF}^{LF} := \frac{1}{2} \left[ F(u_j^+) + F(u_j^-) \right] - \frac{|\lambda_j|}{2} (u_j^+ - u_j^-),
\]

where \( u_j^\pm \) denote the left- and right limits at \( x_j \) of the reconstructed polynomials of degree \( k \) adjacent to \( x_j \), i.e.

\[
u_j^- = \lim_{x \to x_j^-} p_{j-1/2}^k(x) = p_{j-1/2}^k(x_j), \quad (4.14)
\]

\[
u_j^+ = \lim_{x \to x_j^+} p_{j+1/2}^k(x) = p_{j+1/2}^k(x_j). \quad (4.15)
\]

Moreover, \( \lambda_j \) is the maximum eigenvalue of the Jacobian matrix \( \frac{\partial F}{\partial u} \) at \( x_j \), with

\[
\lambda_j = \max(u_j^+, u_j^+ + v_{th,j}^+, v_j^+ - v_{th,j}^-). \quad (4.15)
\]

Here, \( \pm \) refers to right and left limits as in \((4.14)-(4.15)\).

High order reconstruction schemes such as ENO or WENO are available for computing the polynomials \( p_{j+1/2}^k \). Let us present here the ENO scheme. For a \((k + 1)\)-th order accurate reconstruction in each cell,

\[
p_{j+1/2}^k(x) = u(x) + O(\Delta x^{k+1}), \quad x \in [x_j, x_{j+1}],
\]

the cell averages of \( p_{j+1/2}^k \) and \( u \) have to be equal on \( k + 1 \) neighbouring cells, including the cell \( j + 1/2 \). This yields \( k + 1 \) equations to uniquely define \( p_{j+1/2}^k \). The simplest case \( k = 0 \) (Rusanov scheme) is trivial, \( p_{j+1/2}^0 = u_{j+1/2} = \text{const.} \)

Let us discuss here the first non-trivial case \( k = 1 \), called the **MUSCL scheme** (Monotonic Upstream-Centered Scheme for Conservation Laws), which is second-order accurate. In MUSCL, the reconstruction polynomials are of first order,

\[
p_{j+1/2}^1(x) = a_{j+1/2} + b_{j+1/2} (x - x_{j+1/2}),
\]

where \( x_{j+1/2} = (x_j + x_{j+1})/2 \) is the cell center.

The average of \( p_{j+1/2}^1(x) \) over the cell \([x_j, x_{j+1}]\) is \( a_{j+1/2} \), thus \( a_{j+1/2} = u_{j+1/2} \) is one of the equations that determine \( p_{j+1/2}^1 \). To obtain the other equation and thus the second coefficient \( b_{j+1/2} \), we can either use the left or the right adjacent cell. The two possible equations read

\[
\int_{x_{j-1}}^{x_j} p_{j+1/2}^1 \, dx = u_{j-1/2} \quad \Longrightarrow \quad u_{j+1/2} - b_{j+1/2} \Delta x = u_{j-1/2},
\]

\[
\int_{x_{j+1}}^{x_{j+2}} p_{j+1/2}^1 \, dx = u_{j+3/2} \quad \Longrightarrow \quad u_{j+1/2} + b_{j+1/2} \Delta x = u_{j+3/2},
\]

80
One chooses the equation that minimises $b_{j+1/2}$, which corresponds also to minimising the derivative of $p^1_{j+1/2}(x)$. In compact form, this is can be written as

$$b_{j+1/2} = \text{minmod}\left( \frac{u_{j+3/2} - u_{j+1/2}}{\Delta x}, \frac{u_{j+1/2} - u_{j-1/2}}{\Delta x} \right),$$

where the $\text{minmod}$-function is defined as:

$$\text{minmod}(a, b) = \begin{cases} 
  a & \text{if } |a| \leq |b|, ab \geq 0 \\
  b & \text{if } |a| > |b|, ab \geq 0 \\
  0 & \text{if } ab < 0.
\end{cases}$$

The third case is necessary to correctly account for extremal points, where the derivative is zero.
Bibliography


