

Computational plasma physics

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

LECTURE NOTES
SOMMERSEMESTER 2017

April 24, 2017

Contents

1	Introduction	3
1.1	Scientific computing	3
1.2	Plasmas	5
1.3	The N -body model	5
2	Kinetic models	7
2.1	The Vlasov-Maxwell model	7
2.1.1	Reduction of Maxwell's equation	9
2.1.2	Collisions	10
2.2	The Vlasov-Poisson model for electrons	11
2.2.1	The model	11
2.2.2	The Vlasov equation in a given potential	12
2.2.3	Conservation properties	13
2.2.4	Solution of the linearised 1D Vlasov-Poisson equation	15
3	First numerical tools	18
3.1	The finite difference method	18
3.1.1	The 1D Poisson equation and boundary conditions	18
3.1.2	The method of manufactured solutions	19
3.1.3	Obtaining a Finite Difference scheme	19
3.1.4	Higher order finite differences	21
3.1.5	Convergence of finite difference schemes	22
3.2	Fourier analysis	25
3.2.1	Fourier series	25
3.2.2	The Fourier transform	26
3.2.3	The Discrete Fourier Transform	26
3.2.4	Approximation of the Fourier transform with the DFT	28
3.2.5	Circulant matrices	29
3.2.6	Stability of the discrete Laplacian with periodic boundary conditions	30
3.3	The Fourier pseudospectral method	31
3.4	Finite difference methods in 2D	35
3.5	The Finite Difference method for 1D advection	36
3.5.1	Obtaining a Finite Difference scheme	37
3.5.2	The first order explicit upwind scheme	37
3.5.3	The first order upwind implicit scheme	38
3.5.4	The explicit downwind and centred schemes	39
3.5.5	Stability and convergence	39
3.6	Splitting methods	40

3.7	The Finite Element Method	42
3.7.1	Principle of the method	42
3.7.2	The variational (or weak) form of a boundary value problem . . .	46
3.7.3	Lagrange Finite Elements	48
3.7.4	B-spline Finite Elements	51
3.7.5	Convergence of the Finite Element method	53
4	Numerical methods for the Vlasov-Poisson equations	57
4.1	The particle in cell (PIC) method	57
4.1.1	Time scheme for the particles	57
4.1.2	Particle mesh coupling for Finite Elements	58
4.1.3	Particle-Mesh coupling for point based Poisson solvers	58
4.1.4	Time loop.	60
4.1.5	Conservation properties at the semi-discrete level	60
4.1.6	Error estimation in Particle-In-Cell methods	62
4.2	The semi-Lagrangian method	62
5	Fluid models	65
5.1	Derivation of the fluid model from the kinetic model	65
5.2	An isothermal Euler-Poisson model	69
5.2.1	The model	69
5.2.2	Study of the linearised equations	70
5.2.3	Hyperbolicity	72
5.3	The Finite Volume method	73
5.3.1	The first order Finite Volume schemes	73
5.3.2	Higher order schemes	74
5.4	Systems of conservation laws	75
5.4.1	Linear systems - The Riemann problem	75
5.5	Nonlinear systems of conservation laws	77
5.5.1	The Rusanov flux	78
5.5.2	The Roe flux	79

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 [5] 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 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 [17] 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 microelectronics, production of X-rays.

We should also mention that while it is almost absent in the natural state on 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 The N -body model

At the microscopic level, a plasma or a particle beam is composed of a number of 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

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

where m is the mass of the particle, \mathbf{v} its velocity $\gamma = (1 - \frac{|\mathbf{v}|^2}{c^2})^{-\frac{1}{2}}$ is the Lorentz factor (c being the speed of light). The right hand side F_{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,

$$\frac{d\gamma_i m_i \mathbf{v}_i}{dt} = \sum_j q_i (\mathbf{E}_j + \mathbf{v}_i \times \mathbf{B}_j) + q_i (\mathbf{E}_{ext} + \mathbf{v}_i \times \mathbf{B}_{ext}).$$

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 generated by particle j and $\mathbf{E}_{ext}, \mathbf{B}_{ext}$ denote the external electric and magnetic 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 initial positions and velocities of the particles are known as well as the external fields, the evolution of the particles is completely determined by the equations

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i, \quad (1.1)$$

$$\frac{d\gamma_i m \mathbf{v}_i}{dt} = \sum_j q(\mathbf{E}_j + \mathbf{v} \times \mathbf{B}_j), \quad (1.2)$$

where the sum contains the electric and magnetic field generated by each of the other particles as well as the external fields.

In general a plasma consists of a large number of particles, 10^{10} and more. The microscopic model describing the interactions of particles with each other is not used in a simulation because it would be far too expensive. We must therefore find approximate models which, while remaining accurate enough can reach a reasonable computational cost. There is actually a hierarchy of models describing the evolution of a plasma. The base model of the hierarchy and the most accurate model is the N -body model we have described, then there are intermediate models called kinetic and which are based on a statistical description of the particle distribution in phase space and finally the macroscopic or fluid models that identify each species of particles of a plasma with a fluid characterized by its density, velocity and energy. Fluid models are becoming a good approximation when the particles are close to thermodynamic equilibrium, to which they return in long time do 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.

Chapter 2

Kinetic models

In a *kinetic* model, each particle species s in the plasma is characterized by a distribution function $f_s(\mathbf{x}, \mathbf{v}, t)$ 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 $\int_V f_s d\mathbf{x} d\mathbf{v}$ is the average number of particles of species s , whose position and velocity are in the V . Normalising f_s to one, f_s becomes the probability density defining the probability of a particle of species s being at point (\mathbf{x}, \mathbf{v}) in phase space.

The distribution function contains much more information than a fluid description as it includes information on the distributions of particle velocities at each position. A kinetic description of a plasma is essential when the distribution function is far away from the Maxwell-Boltzmann distribution (also called Maxwellian) that corresponds to the thermodynamic equilibrium of plasma. Otherwise a fluid description is sufficient.

2.1 The Vlasov-Maxwell model

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

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_s + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_s = \sum_{\sigma} \mathcal{Q}(f_s, f_{\sigma}), \quad (2.1)$$

in the non relativistic case. In the relativistic case it becomes

$$\frac{\partial f_s}{\partial t} + \mathbf{v}(\mathbf{p}) \cdot \nabla_{\mathbf{x}} f_s + q_s (\mathbf{E} + \mathbf{v}(\mathbf{p}) \times \mathbf{B}) \cdot \nabla_{\mathbf{p}} f_s = \sum_{\sigma} \mathcal{Q}(f_s, f_{\sigma}). \quad (2.2)$$

We denote by $\nabla_{\mathbf{x}} f_s$, $\nabla_{\mathbf{v}} f_s$ and $\nabla_{\mathbf{p}} f_s$, 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 $\mathbf{v}(\mathbf{p}) = \frac{\mathbf{p}}{m_s \gamma_s}$, where γ is the Lorentz factor which can be expressed from the momentum by $\gamma_s = \sqrt{1 + |\mathbf{p}|^2 / (m_s^2 c^2)}$. On the right-hand-side $\mathcal{Q}(f_s, f_{\sigma})$ 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 $f_{s,0}(\mathbf{x}, \mathbf{v})$ 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:

$$\begin{aligned} -\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} + \nabla \times \mathbf{B} &= \mu_0 \mathbf{J}, & (\text{Ampère}) \\ \frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} &= 0, & (\text{Faraday}) \\ \nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0}, & (\text{Gauss}) \\ \nabla \cdot \mathbf{B} &= 0, & (\text{magnetic Gauss}) \end{aligned}$$

where the constants are c the speed of light, μ_0 the permeability of free space and ε_0 the permittivity of free space. The source terms for Maxwell's equation, the charge density $\rho(\mathbf{x}, t)$ and the current density $\mathbf{J}(\mathbf{x}, t)$ can be expressed from the distribution functions of the different species of particles $f_s(\mathbf{x}, \mathbf{v}, t)$ using the relations

$$\begin{aligned} \rho(\mathbf{x}, t) &= \sum_s q_s \int f_s(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}, \\ \mathbf{J}(\mathbf{x}, t) &= \sum_s q_s \int f_s(\mathbf{x}, \mathbf{v}, t) \mathbf{v} d\mathbf{v}. \end{aligned}$$

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

$$\rho(\mathbf{x}, t) = \sum_s q_s \int f_s(\mathbf{x}, \mathbf{p}, t) d\mathbf{p}, \quad \mathbf{J}(\mathbf{x}, t) = \sum_s q_s \int f_s(\mathbf{x}, \mathbf{p}, t) \mathbf{v}(\mathbf{p}) d\mathbf{p}.$$

The Maxwell equations need to be supplemented by boundary and initial conditions so that they admit a unique solution. A classical boundary condition is the perfect conductor boundary condition $\mathbf{E} \times \mathbf{n} = 0$, where \mathbf{n} denotes the unit outgoing normal. No additional condition on \mathbf{B} is needed in that case.

The macroscopic quantities, associated to each particle species are defined as follows:

- The particle density, in physical space, for species s , is defined by

$$n_s(\mathbf{x}, t) = \int f_s(\mathbf{x}, \mathbf{v}, t) d\mathbf{v},$$

- The mean velocity $\mathbf{u}_s(\mathbf{x}, t)$ verifies

$$n_s(\mathbf{x}, t) \mathbf{u}_s(\mathbf{x}, t) = \int f_s(\mathbf{x}, \mathbf{v}, t) \mathbf{v} d\mathbf{v},$$

- The kinetic energy is defined by

$$n_s(\mathbf{x}, t)\mathcal{E}_s(\mathbf{x}, t) = \frac{m}{2} \int f_s(\mathbf{x}, \mathbf{v}, t) |\mathbf{v}|^2 d\mathbf{v},$$

- Introducing the Boltzmann constant k_B , the temperature $T_s(\mathbf{x}, t)$ is related to the kinetic energy, mean velocity and density by

$$\frac{3}{2}k_B T_s(\mathbf{x}, t) = m_s(\mathcal{E}_s(\mathbf{x}, t) - u_s^2(\mathbf{x}, t)),$$

and the pressure is related to the temperature by $p = nk_B T$.

2.1.1 Reduction of Maxwell's equation

In some cases the frequency of the phenomena of interest is sufficiently small that the electric and magnetic fields can be considered quasi-static. This means that the time derivatives can be neglected in Maxwell's equations. We then get two decoupled set of equations. The electric field is then determined by

$$\nabla \times \mathbf{E} = 0, \quad \nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0},$$

and appropriate boundary conditions.

Moreover, provided the computational domain is simply connected, $\nabla \times \mathbf{E} = 0$ implies that there exists a scalar potential ϕ such that $\mathbf{E} = -\nabla\phi$ and ϕ is determined by $-\nabla \cdot \nabla\phi = \rho/\varepsilon_0$, which becomes that standard Poisson equation:

$$-\Delta\phi = \frac{\rho}{\varepsilon_0}.$$

The perfectly conducting boundary condition $\mathbf{E} \times \mathbf{n} = 0$ implies that ϕ is constant on the boundary, and as \mathbf{E} and not ϕ is the physically meaningful field, ϕ is determined up to a constant, and so is generally set to 0 on the boundary for perfect conductors.

Note that in many low frequency plasma, \mathbf{E} is by far the dominating term in the Lorentz force, and hence a good model consists just of the Vlasov-Poisson equations (where the magnetic field is set to 0 or is a known external field).

When still the magnetic field needs to be considered it is solution of

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}, \quad \nabla \cdot \mathbf{B} = 0.$$

Because of the vanishing divergence, provided again some geometric assumptions on the domain, \mathbf{B} derives from a vector potential: $\mathbf{B} = \nabla \times \mathbf{A}$ and \mathbf{A} is a solution of

$$\nabla \times \nabla \times \mathbf{A} = \mu_0 \mathbf{J},$$

along with appropriate boundary conditions. A so-called gauge condition is necessary in addition to uniquely determine the potential, although the fields are independent of it. For steady state problems one generally uses the Coulomb gauge $\nabla \cdot \mathbf{A} = 0$.

2.1.2 Collisions

The Vlasov equation describes a smoothed interaction of the plasma particles, for which the electromagnetic field is averaged over all the plasma particles. In practice, however, when two particles of the same charge get close together there is a strong repulsion between the two particles and this force dominates the force generated by all the other particles. This is called a binary collision. In this case the interaction is modelled by a binary collision operator, like the Boltzmann operator that is used in neutral gases.

When binary collisions between particles are dominant with respect to mean field effects, the distribution function satisfies the Boltzmann equation

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} = \sum_s \mathcal{Q}(f, f_s),$$

where \mathcal{Q} is the non linear Boltzmann operator. This operator is sometimes replaced by simpler models. A sum on the collisions with all the species of particles represented by f_s , including the particles of the same species, is considered. In many cases not all the collisions might be considered. In most plasmas either binary collisions are completely neglected or only represent a complement to the averaged interaction. In this case the collision operator appears on the right-hand side of the full Vlasov equation. The collision operator considered appropriate in most cases then is the Fokker-Planck-Landau operator.

The Fokker-Planck-Landau operator, which is the most commonly used in plasma physics, is the limit of the Boltzmann operator for grazing collisions [8]. It reads for collisions with particles of the same species

$$Q_L(f, f)(v) = \nabla_v \cdot \int A(\mathbf{v} - \mathbf{v}_*) [f(\mathbf{v}_*) \nabla_v f(\mathbf{v}) - f(\mathbf{v}) \nabla_{v_*} f(\mathbf{v}_*)] d\mathbf{v}_* \quad (2.3)$$

where, for the case of Coulomb collisions, Λ being a positive constant,

$$A(\mathbf{v} - \mathbf{v}_*) = \frac{\Lambda}{|\mathbf{v} - \mathbf{v}_*|} \left(\mathbb{I}_3 - \frac{(\mathbf{v} - \mathbf{v}_*) \otimes (\mathbf{v} - \mathbf{v}_*)}{|\mathbf{v} - \mathbf{v}_*|^2} \right).$$

As for the Boltzmann operator, the first three velocity moments of the Landau operator vanish, which implies conservation of mass, momentum and kinetic energy. Moreover the H-theorem is satisfied, so that the equilibrium states are also the Maxwellians.

Sometimes it is more convenient to express the Landau collision operator using the Rosenbluth potentials [11], which reads

$$Q_{L,R}(f, f_*)(v) = \Lambda \nabla_v \cdot [\nabla_v \cdot (f \nabla_v \otimes \nabla_v \mathbf{G}(f_*)) - 4f \nabla_v \cdot \mathbf{H}(f_*)]$$

where the Rosenbluth potentials are defined by

$$\mathbf{G}(f_*)(\mathbf{v}) = \int |\mathbf{v} - \mathbf{v}_*| f_*(\mathbf{v}_*) d\mathbf{v}_*, \quad \mathbf{H}(f_*)(\mathbf{v}) = \int \frac{1}{|\mathbf{v} - \mathbf{v}_*|} f_*(\mathbf{v}_*) d\mathbf{v}_*. \quad (2.4)$$

When the distribution function is close enough to a Maxwellian, the Fokker-Planck-Landau operator can be linearised around a Maxwellian, the collision operator takes a much simpler form as can be seen from the Rosenbluth potential for by taking f_* in the

expression of the potentials \mathbf{G} and \mathbf{H} to be a given Maxwellian. Then we get the linear Fokker-Planck operator, which takes the form

$$Q_{FP}(f)(v) = \nu \nabla_v \cdot (\mu f \mathbf{v} + \frac{D^2}{2} \nabla_v f). \quad (2.5)$$

This operator is also known as the Lenard-Bernstein operator in the plasma physics community [9]. For given constants ν , μ and D , the Lenard-Bernstein operator conserves mass, but not momentum and energy and its equilibrium function is a Maxwellian of the form $\alpha e^{-\frac{\mu v^2}{D^2}}$, where the constant α is determined by the total mass (or number of particles).

The linear Fokker-Planck operator can be made to conserve also total momentum and kinetic energy by using the mean velocity \mathbf{u} and the temperature T associated to f . Then the operator reads

$$Q_{FPC}(f)(v) = \nu \nabla_v \cdot (f \frac{\mathbf{v} - \mathbf{u}}{T} + \nabla_v f).$$

A simplified collision operator that has been build to conserve mass, momentum and kinetic energy and have the Maxwellian as equilibrium states, as the Boltzmann and Fokker-Planck-Landau operators, has been derived by Bhatnagar, Gross and Krook [2]. In the mathematics community this is known as the BGK operator and in the physics community it is called the Krook operator. It simply reads

$$Q_K(f)(v) = \nu(f_M[f] - f),$$

where $f_M[f]$ is the Maxwellian, which has the same mass, mean velocity and temperature as f . It reads

$$f_M(t, \mathbf{x}, \mathbf{v}) = \frac{n(t, \mathbf{x})}{(2\pi T(t, \mathbf{x})/m)^{\frac{3}{2}}} e^{-\frac{|\mathbf{v} - \mathbf{u}(\mathbf{x}, t)|^2}{2T(t, \mathbf{x})/m}}.$$

2.2 The Vlasov-Poisson model for electrons

2.2.1 The model

Starting from the Vlasov-Maxwell equations, consisting of a Vlasov equation for each particle species non linearly coupled by the Maxwell equations determining the evolution of the electromagnetic field of the plasma, we make the assumption, that on the time scale of interest, due to their much larger mass the ions do not move and also that the electric and magnetic fields are slowly varying. If the particles' energy is considered small, the $\mathbf{v} \times \mathbf{B}$ term can be neglected in the Lorentz force, and the remaining simplified model is the Vlasov-Poisson equation for electrons with a neutralizing background. Setting the physical constants to one, making sure to keep the right sign for the charge of electrons and ions, the model reads

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_x f - \mathbf{E} \cdot \nabla_v f = 0, \quad (2.6)$$

$$-\Delta \phi = \rho = 1 - n, \quad \mathbf{E} = -\nabla \phi, \quad (2.7)$$

where n is the electron density defined by

$$n(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}.$$

The domain on which the system is posed is considered periodic of period one in \mathbf{x} and the whole space \mathbb{R}^3 in velocity.

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

$$\frac{\partial f}{\partial t} + \mathbf{A} \cdot \nabla_{\mathbf{x}, \mathbf{v}} f = 0. \quad (2.8)$$

Moreover, as

$$\nabla_{\mathbf{x}, \mathbf{v}} \cdot (\mathbf{A}f) = \mathbf{A} \cdot \nabla f + f \nabla_{\mathbf{x}, \mathbf{v}} \cdot \mathbf{A}$$

and $\nabla_{\mathbf{x}, \mathbf{v}} \cdot \mathbf{A} = 0$, the Vlasov equation (2.6) can also be written in conservative form

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{x}, \mathbf{v}} \cdot (\mathbf{F}f) = 0. \quad (2.9)$$

2.2.2 The Vlasov equation in a given potential

First verification tests for the Vlasov solver consist in considering the Vlasov equation in simple given potentials where the solution can be computed exactly with the method of characteristics. Let us denote by $\mathbf{z} = (\mathbf{x}, \mathbf{v})$ the phase space variable.

Consider the Vlasov equation in advective form:

$$\frac{\partial f}{\partial t} + \mathbf{A} \cdot \nabla f = 0, \quad (2.10)$$

with $f : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}$ and $\mathbf{A} : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}^d$.

Consider now for $s \in \mathbb{R}^+$ given, the differential system

$$\frac{d\mathbf{Z}}{dt} = \mathbf{A}(\mathbf{Z}, t), \quad (2.11)$$

$$\mathbf{Z}(s) = \mathbf{z}, \quad (2.12)$$

which is naturally associated to the advection equation (2.10).

Definition 1 *The solutions of the system (2.11) are called characteristics of the linear advection equation (2.10). We denote by $\mathbf{Z}(t; s, \mathbf{z})$ the solution of (2.11) – (2.12).*

An essential property of the Vlasov equation is that its solution is invariant along the characteristics. This can be verified by computing

$$\frac{d}{dt} f(t, \mathbf{Z}(t)) = \frac{\partial f}{\partial t}(t, \mathbf{Z}(t)) + \frac{d\mathbf{Z}}{dt} \cdot \nabla f(t, \mathbf{Z}(t)) = \frac{\partial f}{\partial t}(t, \mathbf{Z}(t)) + \mathbf{A} \cdot \nabla f(t, \mathbf{Z}(t)).$$

Hence the solution of the Vlasov equation can be expressed using the characteristics.

Proposition 1 *Assuming that the Vlasov equation admits a smooth solution and its characteristics are well defined. The solution can be expressed using the initial condition f_0 and the characteristics \mathbf{Z} as*

$$f(t, \mathbf{x}) = f_0(\mathbf{Z}(0; t, \mathbf{x})).$$

Examples

1. The free streaming equation

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} = 0.$$

The characteristics are solution of

$$\frac{dX}{dt} = V, \quad \frac{dV}{dt} = 0.$$

This we have $V(t; s, x, v) = v$ and $X(t; s, x, v) = x + (t - s)v$ which gives us the solution

$$f(x, v, t) = f_0(x - vt, v).$$

2. Uniform focusing in a particle accelerator (1D model). We then have $E(x, t) = -x$ and the Vlasov writes

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - x \frac{\partial f}{\partial v} = 0.$$

$$\frac{dX}{dt} = V, \quad \frac{dV}{dt} = -X.$$

Whence we get $X(t; s, x, v) = x \cos(t - s) + v \sin(t - s)$ and $V(t; s, x, v) = -x \sin(t - s) + v \cos(t - s)$ from which we compute the solution

$$f(x, v, t) = f_0(x \cos t - v \sin t, x \sin t + v \cos t).$$

2.2.3 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 2 *The Vlasov-Poisson system verifies the following conservation properties:*

- *Maximum principle*

$$0 \leq f(\mathbf{x}, \mathbf{v}, t) \leq \max_{(\mathbf{x}, \mathbf{v})} (f_0(\mathbf{x}, \mathbf{v})). \quad (2.13)$$

- *Conservation of L^p , norms for p integer, $1 \leq p \leq \infty$*

$$\frac{d}{dt} \left(\int (f(\mathbf{x}, \mathbf{v}, t))^p d\mathbf{x} d\mathbf{v} \right) = 0 \quad (2.14)$$

- *Conservation of total momentum*

$$\frac{d}{dt} \int \mathbf{v} f d\mathbf{x} d\mathbf{v} = - \frac{d}{dt} \int \mathbf{J} d\mathbf{x} = 0. \quad (2.15)$$

- *Conservation of total energy*

$$\frac{d}{dt} \left[\frac{1}{2} \int v^2 f d\mathbf{x} d\mathbf{v} + \frac{1}{2} \int \mathbf{E}^2 d\mathbf{x} \right] = 0. \quad (2.16)$$

Proof. The system defining the associated characteristics writes

$$\frac{d\mathbf{X}}{dt} = \mathbf{V}(t), \quad (2.17)$$

$$\frac{d\mathbf{V}}{dt} = -\mathbf{E}(\mathbf{X}(t), t). \quad (2.18)$$

We denote by $(\mathbf{X}(t; \mathbf{x}, \mathbf{v}, s), \mathbf{V}(t; \mathbf{x}, \mathbf{v}, s))$, or more concisely $(\mathbf{X}(t), \mathbf{V}(t))$ when the dependency with respect to the initial conditions is not explicitly needed, the unique solution at time t of this system which takes the value (\mathbf{x}, \mathbf{v}) at time s .

Using (2.17)-(2.18), the Vlasov equation (2.6) can be expressed equivalently

$$\frac{d}{dt}(f(\mathbf{X}(t), \mathbf{V}(t))) = 0.$$

We thus have

$$f(\mathbf{x}, \mathbf{v}, t) = f_0(\mathbf{X}(0; \mathbf{x}, \mathbf{v}, t), \mathbf{V}(0; \mathbf{x}, \mathbf{v})).$$

From this expression, we deduce that f verifies a maximum principle which can be written as f_0 is non negative

$$0 \leq f(\mathbf{x}, \mathbf{v}, t) \leq \max_{(x,v)}(f_0(x, v)).$$

Multiplying the Vlasov equation by (2.6) par f^{p-1} and integrating on the whole phase-space we obtain

$$\frac{d}{dt} \left(\int (f(\mathbf{x}, \mathbf{v}, t))^p d\mathbf{x} d\mathbf{v} \right) = 0,$$

so that the L^p norms of f are conserved for all $p \in \mathbb{N}^*$. Let us notice that the L^∞ is also conserved thanks to the maximum principle (2.13).

Let us now proceed to the conservation of momentum. We shall use the following equality that is verified for any vector \mathbf{u} depending on \mathbf{x} in a periodic domain

$$\int (\nabla \times \mathbf{u}) \times \mathbf{u} d\mathbf{x} = - \int (\mathbf{u}(\nabla \cdot \mathbf{u}) + \frac{1}{2} \nabla u^2) d\mathbf{x} = - \int \mathbf{u}(\nabla \cdot \mathbf{u}) d\mathbf{x}. \quad (2.19)$$

Let us notice in particular that taking $\mathbf{u} = \mathbf{E}$ in the previous equality with \mathbf{E} solution of the Poisson equation (2.7), we get, as $\nabla \times \mathbf{E} = 0$ and $\nabla \cdot \mathbf{E} = -\Delta\phi = 1 - \rho$, that $\int \mathbf{E}(1 - \rho) d\mathbf{x} = 0$. As moreover $\mathbf{E} = -\nabla\phi$ and as we integrate on a periodical domain $\int \mathbf{E} d\mathbf{x} = 0$. It results that

$$\int \mathbf{E}\rho d\mathbf{x} = 0. \quad (2.20)$$

Let us now introduce the Green formula on the divergence:

$$\int_{\Omega} \nabla \cdot \mathbf{F} q + \int_{\Omega} \mathbf{F} \cdot \nabla q = \int_{\partial\Omega} (\mathbf{F} \cdot \mathbf{n}) q \quad \forall \mathbf{F} \in H(\text{div}, \Omega), q \in H^1(\Omega), \quad (2.21)$$

where classically $H^1(\Omega)$ is the subset of $L^2(\Omega)$ the square integrable functions, of the functions whose gradient is in $L^2(\Omega)$; and $H(\text{div}, \Omega)$ is the subset of $L^2(\Omega)$ of the functions whose divergence is in $L^2(\Omega)$.

Let's multiply the Vlasov equation (2.6) by \mathbf{v} and integrate in \mathbf{x} and in \mathbf{v}

$$\frac{d}{dt} \int \mathbf{v} f d\mathbf{x} d\mathbf{v} + \int \nabla_x \cdot (\mathbf{v} \otimes \mathbf{v} f) d\mathbf{x} d\mathbf{v} - \int \mathbf{v} \nabla_v \cdot (\mathbf{E} f) d\mathbf{x} d\mathbf{v} = 0.$$

The second integral vanishes as the domain is periodic in \mathbf{x} and the Green formula on the divergence (2.21) gives for the last integral

$$-\int \mathbf{v} \nabla_v \cdot (\mathbf{E}f) d\mathbf{x}d\mathbf{v} = \int \mathbf{E}f d\mathbf{x}d\mathbf{v} = \int \mathbf{E}\rho d\mathbf{x} = 0,$$

using (2.20). It finally follows that

$$\frac{d}{dt} \int \mathbf{v}f d\mathbf{x}d\mathbf{v} = -\frac{d}{dt} \int \mathbf{J} d\mathbf{x} = 0.$$

In order to obtain the energy conservation property, we start by multiplying the Vlasov equation by $\mathbf{v} \cdot \mathbf{v} = |\mathbf{v}|^2$ and we integrate over phase space

$$\frac{d}{dt} \int |\mathbf{v}|^2 f d\mathbf{x}d\mathbf{v} + \int \nabla_x \cdot (|\mathbf{v}|^2 \mathbf{v}f) d\mathbf{x}d\mathbf{v} - \int |\mathbf{v}|^2 \nabla_v \cdot (\mathbf{E}f) d\mathbf{x}d\mathbf{v} = 0.$$

As f is periodic in \mathbf{x} , we get, integrating in \mathbf{x} that

$$\int \nabla_x \cdot (|\mathbf{v}|^2 \mathbf{v}f) d\mathbf{x}d\mathbf{v} = 0$$

and the Green formula on the divergence (2.21) yields

$$\int |\mathbf{v}|^2 \nabla_v \cdot \mathbf{E} d\mathbf{x}d\mathbf{v} = -2 \int \mathbf{v} \cdot (\mathbf{E}f) d\mathbf{x}d\mathbf{v} = 2 \int \mathbf{E} \cdot \mathbf{J} d\mathbf{x}.$$

So

$$\frac{d}{dt} \int |\mathbf{v}|^2 f d\mathbf{x}d\mathbf{v} = 2 \int \mathbf{E} \cdot \mathbf{J} d\mathbf{x} = -2 \int \nabla \phi \cdot \mathbf{J} d\mathbf{x}. \quad (2.22)$$

On the other hand, integrating the Vlasov equation (2.6) with respect to \mathbf{v} , we get the charge conservation equation, generally called continuity equation: $\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0$. Then, using again the Green formula (2.21), the Poisson equation (2.7) and the continuity equation, we obtain

$$-\int \nabla \phi \cdot \mathbf{J} d\mathbf{x} = \int \phi \nabla \cdot \mathbf{J} d\mathbf{x} = -\int \phi \frac{\partial \rho}{\partial t} d\mathbf{x} = \int \phi \frac{\partial \Delta \phi}{\partial t} d\mathbf{x} = -\frac{1}{2} \frac{d}{dt} \int \nabla \phi \cdot \nabla \phi d\mathbf{x}.$$

And so, plugging this equation in (2.22) and using that $\mathbf{E} = -\nabla \phi$, we get the conservation of energy. ■

2.2.4 Solution of the linearised 1D Vlasov-Poisson equation

Another important verification test, which is often also important for a better understanding of the physics, is to consider the problem linearised around an equilibrium solution.

For the Vlasov-Poisson system, let us first realise that any constant homogeneous distribution function, *i.e.* a distribution function which does not depend on t and \mathbf{x} , but only on \mathbf{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 important case of thermodynamic equilibrium for which the equilibrium distribution that we denote by f^0 is the Maxwellian

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

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

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

The distribution function f verifies the Vlasov-Poisson equations

$$\begin{aligned} \frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - E(x) \frac{\partial f}{\partial v} &= 0, \\ \frac{dE}{dx} &= (1 - \int_{-\infty}^{+\infty} f(x, v, t) dv), \end{aligned}$$

with an initial condition that we assume of the form

$$f_0(x, v) = f^0(v) + \epsilon f_0^1(x, v).$$

Plugging the expansions of f and E in this equation

$$\begin{aligned} \epsilon \left(\frac{\partial f^1}{\partial t} + v \frac{\partial f^1}{\partial x} \right) - (E^0 + \epsilon E^1) \left(\frac{df^0}{dv} + \epsilon \frac{\partial f^1}{\partial v} \right) &= 0, \\ \epsilon \frac{dE^1}{dx} &= \frac{e}{\epsilon_0} (1 - \int_{-\infty}^{+\infty} (f^0(v) + \epsilon f^1(x, v, t)) dv). \end{aligned}$$

Neglecting the terms in ϵ^2 , we obtain, knowing that $E^0 = 0$

$$\frac{\partial f^1}{\partial t} + v \frac{\partial f^1}{\partial x} - E^1(x) \frac{df^0}{dv} = 0, \tag{2.23}$$

$$\frac{dE^1}{dx} = - \int_{-\infty}^{+\infty} f^1(x, v, t) dv, \tag{2.24}$$

with the initial condition $f^1(x, v, 0) = f_0^1(x, v)$. 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 can be expressed simply by

$$D(k, \omega) = 1 + \frac{\omega_p^2}{k^2} \left[1 + \frac{\omega}{\sqrt{2}k} Z\left(\frac{\omega}{\sqrt{2}k}\right) \right], \tag{2.25}$$

using the so-called plasma dispersion function Z defined by

$$Z(\xi) = \sqrt{\pi} e^{-\xi^2} [-i - \operatorname{erfi}(\xi)]$$

where $\operatorname{erfi}(\zeta) = \frac{2}{\sqrt{\pi}} \int_0^\zeta e^{t^2} dt$ is the complex error function.

To obtain an explicit value of this expression of the electric field, it remains to compute numerically for k fixed the values of ω for which $D(k, \omega)$ vanishes. The simplest way to this is to use the Newton method, but this needs a good initial guess. We obtain the following values ω for different k :

k	ω
0.5	$1,4156 - 0,1533i$
0.4	$1,2850 - 0,0661i$
0.3	$1,1598 - 0,0126i$
0.2	$1,0640 - 5,510 \times 10^{-5}i$

Newton's method is very sensitive to the initial guess and gives no insurance to find the most unstable or the least damped mode. A more robust method to compute the zeros of an analytic function.

Chapter 3

First numerical tools

3.1 The finite difference method

3.1.1 The 1D Poisson equation and boundary conditions

We consider the Poisson equation in an interval $[a, b]$. 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: ϕ is given at $x = a$ and $x = b$

$$-\phi''(x) = \rho \quad \text{in } [a, b] \quad (3.1)$$

$$\phi(a) = \alpha, \quad (3.2)$$

$$\phi(b) = \beta. \quad (3.3)$$

2. Neumann boundary conditions: ϕ' is given at boundary. Note that we can do this only at one end of the interval, else there is still an undetermined constant. Moreover as the potential ϕ is determined up to a constant, we can always set it to zero at one point for example at $x = a$. In this case the problem becomes

$$-\phi''(x) = \rho \quad \text{in } [a, b] \quad (3.4)$$

$$\phi(a) = 0, \quad (3.5)$$

$$\phi'(b) = \alpha. \quad (3.6)$$

3. Periodic boundary conditions. In this case all the functions are assumed to be periodic of period L and 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.

$$-\phi''(x) = \rho \quad (3.7)$$

Note however in this case that ϕ is only determined up to a constant, which needs to be set for a numerical computation. Moreover, integrating (3.7) on a period, *e.g.* $[0, L]$ yields

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

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

3.1.2 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 u and apply the operator to it, in our case the Laplacian, to find the corresponding right hand side ρ , and then solve the problem with this ρ 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 one 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.

3.1.3 Obtaining a Finite Difference scheme

We first consider a uniform mesh of the 1D computational domain, i.e. of the interval $[0, L]$ where we want to compute the solution, see Figure 3.1. The cell size or space step

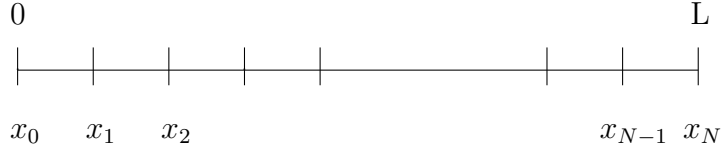


Figure 3.1: Uniform mesh of $[0, L]$

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 will be defined by its values at x_j for $0 \leq j \leq N$.

The principle of Finite Differences is to replace derivatives by finite differences involving neighbouring points approximating those derivatives. The simplest way to do this is to use Taylor expansions around the considered point. 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.

$$\phi(x_{j+1}) = \phi(x_j) + h\phi'(x_j) + \frac{h^2}{2}\phi''(x_j) + \frac{h^3}{6}\phi^{(3)}(x_j) + \frac{h^4}{24}\phi^{(4)}(x_j + \theta_j^+ h), \quad (3.8)$$

$$\phi(x_{j-1}) = \phi(x_j) - h\phi'(x_j) + \frac{h^2}{2}\phi''(x_j) - \frac{h^3}{6}\phi^{(3)}(x_j) + \frac{h^4}{24}\phi^{(4)}(x_j - \theta_j^- h). \quad (3.9)$$

We deduce

$$\phi(x_{j+1}) - 2\phi(x_j) + \phi(x_{j-1}) = h^2\phi''(x_j) + \frac{h^4}{24}(\phi^{(4)}(x_j + \theta_j^+ h) + \phi^{(4)}(x_j - \theta_j^- h)). \quad (3.10)$$

So that

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

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

$$-\frac{\phi(x_{j+1}) - 2\phi(x_j) + \phi(x_{j-1}))}{h^2} = \rho(x_j) + \frac{h^2}{24}(\phi^{(4)}(x_j + \theta_j^+ h) + \phi^{(4)}(x_j - \theta_j^- h)). \quad (3.11)$$

Let us now define ϕ_j 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 we use the boundary conditions to determine the additional unknowns. Then ϕ_j will give an approximation of $\phi(x_j)$ for all the points on the grid $0 \leq j \leq N$.

1. Dirichlet: $\phi_0 = \phi(x_0) = \alpha$, $\phi_N = \phi(x_N) = \beta$. So there remain $N-1$ unknowns $\phi_1, \dots, \phi_{N-1}$ 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

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & & \\ -1 & 2 & -1 & 0 & \\ 0 & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & -1 \\ & & 0 & -1 & 2 \end{pmatrix}, \quad \Phi_h = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{pmatrix}, \quad R_h = \begin{pmatrix} \rho(x_1) + \frac{\alpha}{h^2} \\ \rho(x_2) \\ \vdots \\ \rho(x_{N-1}) + \frac{\beta}{h^2} \end{pmatrix}.$$

2. Neumann. Because we need to set the constant for the potential at one point, we consider now the boundary conditions $\phi(0) = 0$ and $\phi'(L) = \alpha$. In this case the unknown are ϕ_1, \dots, ϕ_N . So there are N unknown. 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 $\phi'(L) = \alpha$. This needs to be expressed from the point values. For this we can simply set

$$\frac{\phi_N - \phi_{N-1}}{h} = \alpha.$$

This is only a first order approximation of the derivative, but this is enough to keep the second order approximation on ϕ at the end.

In this case we get the linear system $A_h \Phi_h = R_h$ with

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & & \\ -1 & 2 & -1 & 0 & \\ 0 & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & -1 \\ & & 0 & -1 & 2 \\ & & 0 & -1 & 1 \end{pmatrix}, \quad \Phi_h = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{pmatrix}, \quad R_h = \begin{pmatrix} \rho(x_1) \\ \rho(x_2) \\ \vdots \\ \rho(x_{N-1}) \\ \frac{\alpha}{h} \end{pmatrix}.$$

3. Periodic. This case is the simplest as there is no boundary. Here all points are interior points and can be used to express $-\phi''(x_j) = \rho(x_j)$. Because of the periodicity $\phi(x_{j+N}) = \phi(x_j)$. Hence only the values of ϕ_j $0 \leq N-1$ 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 $-\phi''(x_j) = \rho(x_j)$, $0 \leq j \leq N-1$ which are expressed by

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

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

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

So that in this case the system in Matrix form reads

$$A_h \Phi_h = R_h \tag{3.12}$$

with

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & & 0 & -1 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ & \ddots & \ddots & \ddots & -1 & 0 \\ & & 0 & -1 & 2 & -1 \\ -1 & & 0 & & -1 & 2 \end{pmatrix}, \quad \Phi_h = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{N-1} \end{pmatrix}, \quad R_h = \begin{pmatrix} \rho(x_0) \\ \rho(x_1) \\ \vdots \\ \rho(x_{N-1}) \end{pmatrix}.$$

We notice that each diagonal of A_h has everywhere the same term. We see also that the vector $(1, \dots, 1)^\top$ 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

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

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, setting the zero mode to zero as is implied by the fact that ϕ has zero average.

3.1.4 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 ϕ 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

$$\phi''(x_j) \approx \frac{-\phi(x_{j+2}) + 16\phi(x_{j+1}) - 30\phi(x_j) + 16\phi(x_{j-1}) - \phi(x_{j-2}))}{12h^2}.$$

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

$$A_h \Phi_h = R_h \quad (3.13)$$

with

$$A_h = \frac{1}{h^2} \begin{pmatrix} \frac{30}{12} & -\frac{16}{12} & \frac{1}{12} & 0 & 0 & 0 & \frac{1}{12} & -\frac{16}{12} \\ -\frac{16}{12} & \frac{30}{12} & -\frac{16}{12} & \frac{1}{12} & 0 & \dots & 0 & \frac{1}{12} \\ \frac{1}{12} & -\frac{16}{12} & \ddots & \ddots & \ddots & \ddots & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & \vdots \\ 0 & & \ddots & \ddots & \ddots & \ddots & \frac{1}{12} & \vdots \\ & & & \ddots & \ddots & \ddots & -\frac{16}{12} & \frac{1}{12} \\ \frac{1}{12} & & & & \frac{1}{12} & -\frac{16}{12} & \frac{30}{12} & -\frac{16}{12} \\ -\frac{16}{12} & \frac{1}{12} & & & 0 & \frac{1}{12} & -\frac{16}{12} & \frac{30}{12} \end{pmatrix}, \quad \Phi_h = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{N-1} \end{pmatrix}, \quad R_h = \begin{pmatrix} \rho(x_0) \\ \rho(x_1) \\ \vdots \\ \rho(x_{N-1}) \end{pmatrix}.$$

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} .

3.1.5 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^∞ (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

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

A simple calculation yields the comparison inequalities between the 2-norm and the max-norm

$$\|v\|_\infty \leq \|v\|_2 \leq \sqrt{N} \|v\|_\infty, \quad \forall v \in \mathbb{R}^N. \quad (3.14)$$

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 Φ_h . For this to provide a good approximation of the corresponding solution of the Poisson equation, we need to verify that for some norm $\|\cdot\|$ we have $\|\Phi - \Phi_h\| \leq Ch^p$ for some integer $p \geq 1$ and a constant C independent of h (we then say that $\|\Phi - \Phi_h\| = O(h^p)$).

In the case of Dirichlet boundary conditions $\Phi = (\phi(x_1), \dots, \phi(x_{N-1}))^\top$ is the vector containing the exact solution of Poisson's equation at the grid points.

Because of (3.11), 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 $A_h(\Phi - \Phi_h) = h^2 S_h$ and also that

$$\|A_h(\Phi - \Phi_h)\| \leq h^2 \|S_h\|.$$

Assuming that the fourth derivative of ϕ is bounded, it follows that

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

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

where C is independent of h . We then say that the numerical scheme defined by the matrix A_h is consistent of order 2 for the max-norm. For the 2-norm, we can use the norm comparison (3.14) and $h = L/N$ to get that

$$\|A_h(\Phi - \Phi_h)\|_2 \leq C_1 h^{3/2}, \text{ or equivalently } \frac{1}{\sqrt{N}} \|A_h(\Phi - \Phi_h)\|_2 \leq C_2 h^2,$$

where C_1 and C_2 are constants independent of h .

Consistency gives us convergence of $A_h(\Phi - \Phi_h)$ to zero. This is not yet enough to prove that Φ_h converges to Φ . This requires another property of the discretisation scheme, which is called *stability*, namely that the norm of the inverse of A_h , $\|A_h^{-1}\|$, is bounded independently of h .

Definition 2 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 3 Let A be diagonalisable in basis of orthonormal eigenvectors and $\lambda_1, \dots, \lambda_N$ 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 orthogonal eigenvectors of A , denote by e_1, \dots, e_N

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

Then assuming that e_i is an eigenvector of A associated to the eigenvalue λ_i , we have

$$Ax = \lambda_1 x_1 e_1 + \dots + \lambda_N x_N e_N. \tag{3.15}$$

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

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

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

For the max-norm, we get from (3.15) 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 $A_h = P\Lambda P^\top$, where Λ is the diagonal matrix containing the eigenvalues. Then $A_h^{-1} = P\Lambda^{-1}P^\top$, where Λ^{-1} contains the inverse of the eigenvalues.

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

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

This leads us to the sufficient condition for stability that for all the eigenvalues λ_i of A_h we have

$$|\lambda_i| \geq C, \quad \text{for some constant } C \text{ independent of } h.$$

Let us now check this for 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:

$$\lambda_k = \frac{k^2\pi^2}{L^2}, \quad \phi_k(x) = \sqrt{\frac{2}{L}} \sin \frac{k\pi x}{L}.$$

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

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

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

$$(\Phi_k)_j = \sqrt{\frac{2}{N}} \sin \frac{kj\pi}{N},$$

and the corresponding eigenvalues are

$$\lambda_k = \frac{4}{h^2} \sin^2 \frac{k\pi}{2N} = \frac{4}{h^2} \sin^2 \frac{kh\pi}{2L}, \quad 1 \leq k \leq N-1.$$

As $0 < k < N$, we have $0 \leq k\pi/(2N) \leq \pi/2$, so that the eigenvalues are positive and in increasing order as sinus is increasing on this interval. It follows that λ_1 is the smallest eigenvalue. Moreover, as $h \rightarrow 0$, we have

$$\lambda_1 = \frac{4}{h^2} \sin^2 \frac{h\pi}{2L} \sim \frac{4}{h^2} \frac{h^2 \pi^2}{4L^2} \rightarrow \frac{\pi^2}{L^2}.$$

This corresponds to the continuous eigenvalue. Then because of the convergence property there exists $h_0 = L/N_0$ such that for $h \leq h_0$ we have for all $N \geq N_0$ that $\lambda_1 = \lambda_1^N \geq (1/2)\pi^2/L^2$, where we use the exponent N to show the dependency of λ_1^N on N (or equivalently on h). Thus for any $N \geq 1$, we have

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

where C is a constant independent on N which proves the stability.

3.2 Fourier analysis

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.

3.2.1 Fourier series

Let f be a periodic function of period L , that is integrable on $[0, L]$. Its Fourier coefficients can then be defined by

$$\hat{f}_k = \frac{1}{L} \int_0^L f(x) e^{-\frac{2i\pi k}{L}x} dx, \quad \forall k \in \mathbb{Z},$$

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

$$S_N(f) = \sum_{k=-N}^N \hat{f}_k e^{\frac{2i\pi k}{L}x}.$$

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

Theorem 1 (Dirichlet) *Assume f L -periodic and piecewise C^1 . Then its Fourier series converges at any point x_0 and*

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

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 :

$$f(x) = \sum_{k=-\infty}^{+\infty} \hat{f}_k e^{\frac{2i\pi k}{L}x}. \quad (3.16)$$

An important identity related to Fourier series is the Parseval identity

$$\frac{1}{L} \int_0^L |f(x)|^2 dx = \sum_{N=-\infty}^{+\infty} |\hat{f}_k|^2. \quad (3.17)$$

3.2.2 The Fourier transform

For functions defined over the whole line (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}

$$L^2(\mathbb{R}) = \{f \mid \int_{-\infty}^{+\infty} |f(x)|^2 dx < +\infty\}$$

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

$$\hat{f}(\zeta) = \int_{-\infty}^{+\infty} f(x) e^{-i\zeta x} dx, \quad (3.18)$$

and its inverse Fourier transform by

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\zeta) e^{i\zeta x} d\zeta. \quad (3.19)$$

Here Parseval's formula reads

$$(f, g) = \int_{-\infty}^{+\infty} f(x) \bar{g}(x) dx = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\zeta) \bar{\hat{g}}(\zeta) d\zeta = \frac{1}{2\pi} (\hat{f}, \hat{g}). \quad (3.20)$$

And in the special case, when $f = g$, we get the Plancherel formula

$$\|f\|^2 = (f, f) = \int_{-\infty}^{+\infty} |f(x)|^2 dx = \frac{1}{2\pi} \|\hat{f}(\zeta)\|^2. \quad (3.21)$$

3.2.3 The Discrete Fourier Transform

Let \mathcal{F}_N be the symmetric matrix formed with the inverse powers of the N^{th} roots of unity, the coefficients of which are given by $\mathcal{F}_{N,jk} = e^{-\frac{2i\pi jk}{N}}$. Denoting by $\omega_N = e^{\frac{2i\pi}{N}}$, 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 , $0 \leq i \leq N-1$ are the vectors, denoted by E_k , interpolating the functions $x \mapsto e^{-2i\pi kx}$ at the grid points $x_j = jL/N$ of the interval $[0, L]$. So the expression of a periodic function in the base of the vectors E_k is naturally associated to the Fourier series of a periodic function.

Definition 3 *Discrete Fourier Transform.*

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

$$\hat{u}_k = \sum_{j=0}^{N-1} u_j e^{-\frac{2i\pi jk}{N}}. \quad (3.22)$$

- 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

$$u_j = \frac{1}{N} \sum_{k=0}^{N-1} \hat{u}_k e^{\frac{2i\pi jk}{N}}. \quad (3.23)$$

Lemma 1 *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

$$(\mathcal{F}_N \mathcal{F}_N^*)_{jk} = \sum_{l=0}^{N-1} \omega_N^{-jl} \omega_N^{lk} = \sum_{l=0}^{N-1} e^{\frac{2i\pi}{N} l(j-k)} = \frac{1 - e^{\frac{2i\pi}{N} N(j-k)}}{1 - e^{\frac{2i\pi}{N} (j-k)}} \text{ if } j \neq k$$

and so $(\mathcal{F}_N \mathcal{F}_N^*)_{jk} = 0$ if $j \neq k$ and $(\mathcal{F}_N \mathcal{F}_N^*)_{jk} = N$ if $j = k$. ■

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:

$$(U, V) = V^\top \bar{V} = \frac{1}{N} \hat{U}^\top \bar{\hat{V}} = \frac{1}{N} (\hat{U}, \hat{V}), \quad (3.24)$$

- The discrete Plancherel identity:

$$\|U\| = \frac{1}{N} \|\hat{U}\|, \quad (3.25)$$

where (\cdot, \cdot) and $\|\cdot\|$ denote the usual euclidian dot product and norm in \mathbb{C}^N .

Proof. The scalar product in \mathbb{C}^N of $U = (u_0, \dots, u_{N-1})^\top$ and $V = (v_0, \dots, v_{N-1})^\top$ 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

$$U^\top \bar{V} = \frac{1}{N^2} (\mathcal{F}_N^* \hat{U})^\top \overline{\mathcal{F}_N^* \hat{V}} = \frac{1}{N^2} \hat{U}^\top \mathcal{F}_N^* \mathcal{F}_N \bar{\hat{V}} = \frac{1}{N} \hat{U}^\top \bar{\hat{V}},$$

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 $U = V$. ■

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.*

3.2.4 Approximation of the Fourier transform with the DFT

On a periodic domain, the Fourier series is just approximated by a truncated Fourier series, the number of modes being equal to the number of grid points, which gives a straightforward interpretation of the DFT as an approximation of the Fourier series. Let us now see, how the DFT can be used to numerically approximate the Fourier transform of a function defined on the whole line. As functions for which the Fourier transform is defined tend to zero at infinity, there exists a positive real number A such that $|f(x)| < \epsilon$ if $|t| > A$, where ϵ is a small number. We can then approximate

$$\int_{-\infty}^{+\infty} f(x) e^{-i\zeta x} dx \approx \int_{-A}^A f(t) e^{-i\zeta x} dx.$$

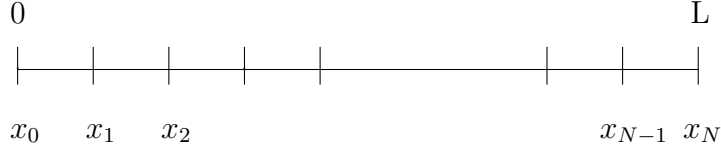


Figure 3.2: 1D mesh.

Consider the mesh of Figure 3.2. Approximating the integral defining the Fourier transform by the rectangle rule we obtain

$$\int_{-A}^A f(x) e^{-i\zeta x} dx \approx \frac{2A}{N} \sum_{j=0}^{N-1} f_j e^{-i\zeta(-A+j\frac{2A}{N})},$$

with $f_j = f(-A+j\frac{2A}{N})$. On the other hand, we will compute the Fourier transform only for the N discrete values $\zeta_k = k\frac{2\pi}{2A} = k\frac{\pi}{A}$. Denoting by $\hat{f}_k = \hat{f}(\omega_k)$, we then obtain

$$\begin{aligned} \hat{f}_k &= \frac{2A}{N} \sum_{j=0}^{N-1} f_j e^{-i\frac{\pi k}{A}(-A+j\frac{2A}{N})} = \frac{2A}{N} \sum_{j=0}^{N-1} f_j e^{-\frac{2i\pi k}{N}(j-\frac{N}{2})}, \\ &= \frac{2A}{N} \left(\sum_{j=0}^{N/2-1} f_j e^{-\frac{2i\pi k}{N}(j+\frac{N}{2})} + \sum_{j=N/2}^{N-1} f_j e^{-\frac{2i\pi k}{N}(j-\frac{N}{2})} \right) \\ &= \frac{2A}{N} \left(\sum_{j=0}^{N/2-1} f_{j+N/2} e^{-\frac{2i\pi jk}{N}} + \sum_{j=N/2}^{N-1} f_{j-N/2} e^{-\frac{2i\pi jk}{N}} \right) \end{aligned}$$

using the periodicity of $e^{-\frac{2i\pi jk}{N}}$ to have only j between 0 and $N-1$. So denoting by g the vector with components $(g_j)_{0 \leq j \leq N-1}$ with $g_j = f_{j+N/2}$ for $0 \leq j \leq N/2-1$ and $g_j = f_{j-N/2}$ for $N/2 \leq j \leq N-1$ and $(\hat{g}_k)_{0 \leq k \leq N-1}$ the coefficients obtained by DFT of g , we have

$$\hat{f}_k = 2A\hat{g}_k.$$

In practice, in order to compute an approximation of the Fourier transform of a function f , start by defining a numerical support of the form $[-A, A]$, sample f on $[-A, A]$, shift (f_j) so that the $N/2$ first term are at the end, then perform a DFT of the so obtained vector g and multiply the result by $2A$. In MATLAB this can be expressed as

```
hatf = 2*A*fftshift(fft(fftshift(f)))
```

Remark 2 *If the numerical support of f is of the form $[a, b]$ non centred in 0, one can start by computing the DFT of the function $t \mapsto f(t - \frac{a+b}{2})$ and then multiply the result by $(e^{-i\frac{a+b}{2}\omega_k})_{0 \leq k \leq N-1}$.*

3.2.5 Circulant matrices

Note that on a uniform grid if the PDE coefficients do not explicitly depend on x a Finite Difference scheme is identical at all the grid points. This implies that a matrix A_h defined with such a scheme has the same coefficients on any diagonal including the periodicity. Such matrices, which are of the form

$$C = \begin{pmatrix} c_0 & c_1 & c_2 & \dots & c_{N-1} \\ c_{N-1} & c_0 & c_1 & & c_{N-2} \\ c_{N-2} & c_{N-1} & c_0 & & c_{N-3} \\ \vdots & & & \ddots & \vdots \\ c_1 & c_2 & c_3 & \dots & c_0 \end{pmatrix}$$

with $c_0, c_1, \dots, c_{N-1} \in \mathbb{R}$ are called *circulant*.

Proposition 4 *The eigenvalues of the circulant matrix C are given by*

$$\lambda_k = \sum_{j=0}^{N-1} c_j \omega^{jk}, \quad (3.26)$$

where $\omega = e^{2i\pi/N}$.

Proof. Let J be the circulant matrix obtained from C by taking $c_1 = 1$ and $c_j = 0$ for $j \neq 1$. We notice that C can be written as a polynomial in J

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

As $J^N = I$, the eigenvalues of J are the N -th roots of unity that are given by $\omega^k = e^{2i\pi k/N}$. Looking for X_k such that $JX_k = \omega^k X_k$ we find that an eigenvector associated

to the eigenvalue λ_k is

$$X_k = \begin{pmatrix} 1 \\ \omega^k \\ \omega^{2k} \\ \vdots \\ \omega^{(N-1)k} \end{pmatrix}.$$

We then have that

$$CX_k = \sum_{j=0}^{N-1} c_j J^j X_k = \sum_{j=0}^{N-1} c_j \omega^{jk} X_k,$$

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

$$\lambda_k = \sum_{j=0}^{N-1} c_j \omega^{jk}.$$

■

Proposition 5 *Any circulant matrix C can be written in the form $C = P\Lambda P^*$ where P is the matrix of the discrete Fourier transform and Λ is the diagonal matrix of the eigenvalues of C . In particular all circulant matrices have the same eigenvectors (which are the columns of P), and any matrix of the form $P\Lambda P^*$ 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, we have $C_1 = P\Lambda_1 P^*$ and $C_2 = P\Lambda_2 P^*$ so $C_1 C_2 = P\Lambda_1 \Lambda_2 P^*$.

If all eigenvalues of $C = P\Lambda P^*$ are non vanishing, Λ^{-1} is well defined and

$$P\Lambda P^* P\Lambda^{-1} P^* = I.$$

So the inverse of C is the circulant matrix $P\Lambda^{-1} P^*$. ■

3.2.6 Stability of the discrete Laplacian with periodic boundary conditions

For periodic boundary conditions the matrix of the discrete Laplacian 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 the formula for computing its eigenvalues can be used to verify its stability. It yields

$$\lambda_k = \frac{1}{h^2} (2 - e^{2i\pi k/N} - e^{2i\pi k(N-1)/N}) = \frac{2}{h^2} (1 - \cos \frac{2k\pi}{N}) = \frac{4}{h^2} \sin^2 \frac{k\pi}{N}, \quad 0 \leq k \leq N-1.$$

In order to fix the constant, we assume that $\Phi_0 + \dots + \Phi_{N-1} = 0$, this sets the constant Fourier mode $\hat{\Phi}_0$ to 0 and discards the eigenvalue $\lambda_0 = 0$, so that the rest of the matrix is invertible and the smallest eigenvalue corresponds to

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

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

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

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.

3.3 The Fourier pseudospectral method

This method is also called the Fourier spectral collocation method. See the books [16, 3] for a detailed description. The principle of spectral methods is to look for an approximation of the solution of some PDE as a discrete N terms expansion over some well chosen discrete 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 in 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

$$Lu = f$$

where L is an arbitrary differential operator, one approximates the exact solution u by

$$u_N(x) = \frac{1}{N} \left(\sum_{k=-N/2+1}^{N/2-1} a_k e^{\frac{2i\pi kx}{L}} + a_{-N/2} \cos \frac{2\pi kx}{L} \right) \quad \text{for } x \in [0, L[\quad (3.27)$$

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

$$Lu_N(x_j) = f(x_j), \quad x_j = j \frac{L}{N} \quad 0 \leq j \leq N-1.$$

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 . Let us denote for integers $-N/2 \leq k \leq N/2$ by E_k the grid modes which are the vectors whose components are

$$e^{\frac{2i\pi kx_j}{L}} = e^{\frac{2i\pi jk}{N}}.$$

Expressing u_N at the collocation points, we get

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

Let us comment of the representation of the last term of u_N in (3.27) as a cos instead of using the $k = N/2$ or $k = -N/2$ modes. We notice that on the grid we have that

$$e^{\frac{2i\pi j N/2}{N}} = e^{i\pi j} = (-1)^j = e^{\frac{-2i\pi j N/2}{N}} = \cos \frac{2\pi j N/2}{N},$$

so the grid representations of the functions $x \mapsto e^{\frac{2i\pi x_j N/2}{L}}$, $x \mapsto e^{-\frac{2i\pi x_j N/2}{L}}$ and $x \mapsto \cos \frac{2\pi x_j N/2}{L}$ 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 (3.27) is needed and not the grid representation is for computing the derivatives. And for this from the three forms only the cos representation has a good behavior. Indeed

$$\begin{aligned} \frac{d}{dx} \left(\cos \frac{2\pi N/2x}{L} \right)_{x=x_j} &= \frac{\pi N}{L} \sin \frac{\pi N x_j}{L} = \frac{\pi N}{L} \sin(\pi j) = 0 \quad 0 \leq j \leq N-1, \\ \frac{d}{dx} \left(e^{\frac{2i\pi N/2x}{L}} \right)_{x=x_j} &= \frac{i\pi N}{L} e^{\frac{i\pi N x_j}{L}} = \frac{i\pi N}{L} e^{i\pi j} = (-1)^j \frac{i\pi N}{L} \quad 0 \leq j \leq N-1, \\ \frac{d}{dx} \left(e^{\frac{-2i\pi N/2x}{L}} \right)_{x=x_j} &= \frac{-i\pi N}{L} e^{\frac{-i\pi N x_j}{L}} = \frac{-i\pi N}{L} e^{-i\pi j} = (-1)^j \frac{-i\pi N}{L} \quad 0 \leq j \leq N-1. \end{aligned}$$

We notice that the three possibilities yield three different grid approximations of the derivatives. In addition of having no good reason too chose either of $e^{\frac{2i\pi N/2x}{L}}$ or $e^{\frac{-2i\pi N/2x}{L}}$ for the approximation, the real problem 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 3 *Note that this problem appear 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

$$u_N^{(m)}(x_j) = \sum_{k=-N/2}^{N/2-1} \left(\frac{2ik\pi}{L} \right)^m a_k e^{\frac{2i\pi k j}{N}} \quad \text{for } 0 \leq j \leq N-1,$$

For m odd, the contibution of the $N/2$ mode vanishes and the m^{th} grid derivative is given by

$$u_N^{(m)}(x_j) = \sum_{k=-N/2+1}^{N/2-1} \left(\frac{2ik\pi}{L} \right)^m a_k e^{\frac{2i\pi k j}{N}} \quad \text{for } 0 \leq j \leq N-1.$$

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

$$-u''(x) = f.$$

For this problem to be well posed we fix the constant by assuming that $\alpha = \int_0^L u(x) dx$ is given.

The pseudospectral method consists in looking for an approximation of the periodic solution u of the form (3.27) and writing that

$$-u_N''(x_j) = f(x_j), \quad \text{for } x_j = jL/N, \quad 0 \leq j \leq N. \quad (3.28)$$

Denoting by $f_j = f(x_j)$, and $F = (f_0, \dots, f_{N-1})^\top$, $\hat{F} = (\hat{f}_0, \dots, \hat{f}_{N-1})^\top = \mathcal{F}_N F$, we have using the discrete Fourier transform that $F = \frac{1}{N} \mathcal{F}_N^* \mathcal{F}_N F$. We can hence write

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

Then (3.28) becomes

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

then using $k' = k + N/2$ in the first sum, using that $e^{-2i\pi j} = 1$

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

The change of summation needs very often to be performed in computations involving the Discrete Fourier Transform. It consists in shifting the first $N/2$ modes at the end. This is done in Matlab using the function `fftshift`. We then get the solution a_k by indentifying the coefficients in the two Fourier series:

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

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 corresponding algorithm then reads, given the grid function $F = (f_0, \dots, f_{N-1})$ and the average value α of u :

1. Perform FFT of F 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 .

Let us now compare the results of the second order Finite Difference scheme (3.12), the fourth order Finite Difference scheme (3.13) and the above Fourier pseudospectral scheme. For this we use the method of manufactured solution. Picking first the L -periodic function $u_{ex}(x) = \exp(\sin(2\pi x/L))$ and computing

$$f(x) = -u''(x) = (2\pi/L)^2 (\sin(2\pi x/L) - \cos^2(2\pi x/L)) \exp(\sin(2\pi x/L)).$$

Figure 3.3 shows the error curves with respect to the cell size $h = L/N$. We notice the straight lines with slopes 2 and 4 respectively for the second and fourth order finite difference and the very quick convergence to round off error for the spectral solver.

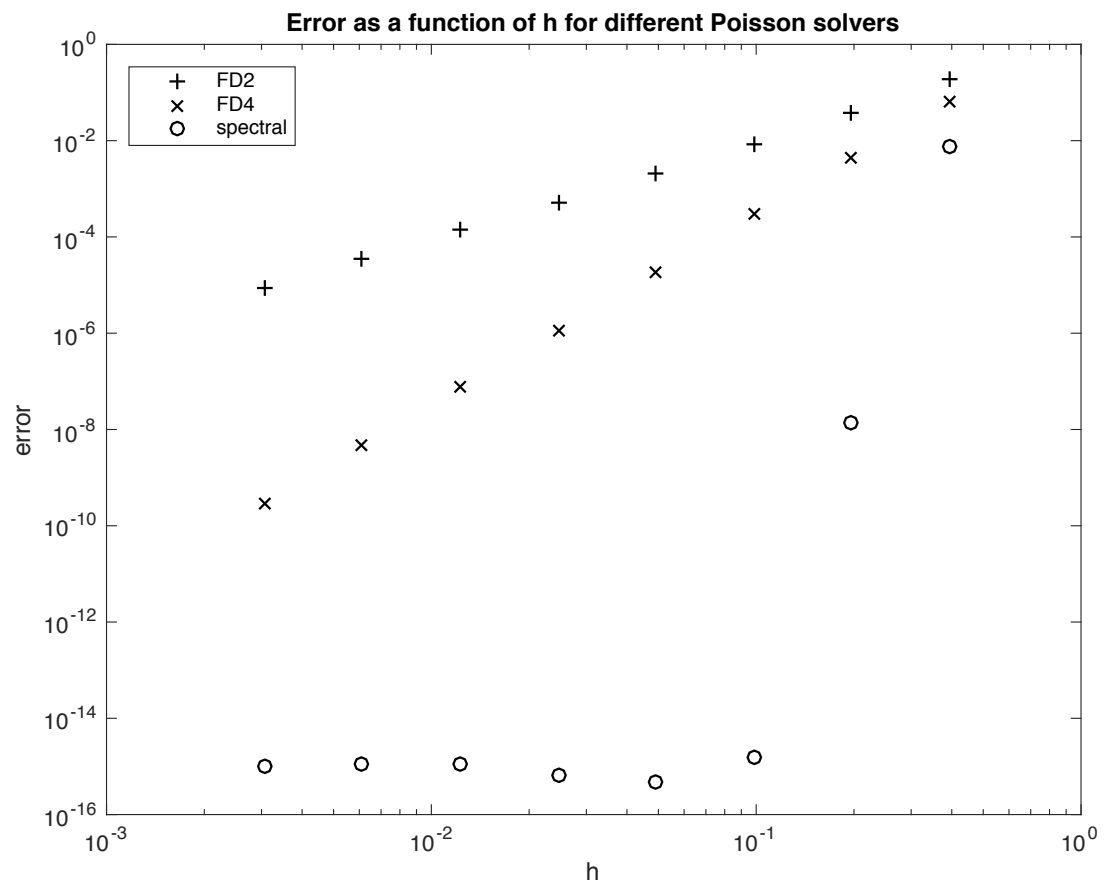


Figure 3.3: Error as a function of cell size periodic Poisson problem for second order finite differences (FD2), fourth order Finite Differences (FD4) and pseudospectral method

3.4 Finite difference methods in 2D

Let us now extend the finite difference method to a cartesian 2D grid, which is a tensor product of 1D grids. We shall assume that the cell size is uniform and equal in the two directions for simplicity, 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.

The 2D Laplace operator is defined by

$$\Delta\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}.$$

The second order finite difference approximation of the second derivative in x and y are obtained from Taylor expansions

$$\begin{aligned} -\Delta\phi(x_i, y_j) &= \frac{-\phi(x_{i+1}, y_j) + 2\phi(x_i, y_j) - \phi(x_{i-1}, y_j))}{h^2} \\ &+ \frac{-\phi(x_i, y_{j+1}) + 2\phi(x_i, y_j) - \phi(x_i, y_{j-1}))}{h^2} + O(h^2). \end{aligned} \quad (3.29)$$

Let us consider the natural numbering of the grid values of the approximate solution Φ_h which is now a matrix with entries $\phi_{i,j} \approx \phi(x_i, y_j)$ for all the grid points $0 \leq i, j \leq N$. Considering the Poisson problem with homogenous Dirichlet boundary conditions: $-\Delta\phi = \rho$ in the domain and $\phi = 0$ on the boundary, there are $(N-1)^2$ unknowns satisfying the $(N-1)^2$ equations

$$\frac{1}{h^2}(-\phi_{i+1,j} - \phi_{i-1,j} + 4\phi_{i,j} - \phi_{i,j+1} - \phi_{i,j-1}) = \rho(x_i, y_j), \quad 1 \leq i, j \leq N-1. \quad (3.30)$$

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

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & & & \\ -1 & 2 & -1 & \ddots & & \\ 0 & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & -1 & 0 \\ & & \ddots & -1 & 2 & -1 \\ & & & 0 & -1 & 1 \end{pmatrix}$$

we notice that the matrix multiplication $A_h\Phi_h$ applies the 1D Finite Difference stencil to the columns of Φ_h which corresponds to the differentiation in y and the left multiplication of Φ_h by A_h , $\Phi_h A_h$, applies the 1D Finite Difference stencil to the lines of Φ which corresponds to the differentiation in x .

$$A_h\Phi_h = \frac{1}{h^2} \begin{pmatrix} 2\phi_{1,1} - \phi_{2,1} & 2\phi_{1,2} - \phi_{2,2} & \dots \\ -\phi_{1,1} + 2\phi_{2,1} - \phi_{3,1} & -\phi_{1,2} + 2\phi_{2,2} - \phi_{3,2} & \dots \\ -\phi_{2,1} + 2\phi_{3,1} - \phi_{4,1} & -\phi_{2,2} + 2\phi_{3,2} - \phi_{4,2} & \dots \\ \vdots & \vdots & \end{pmatrix},$$

$$\Phi_h A_h = \frac{1}{h^2} \begin{pmatrix} 2\phi_{1,1} - \phi_{1,2} & -\phi_{1,1} + 2\phi_{1,2} - \phi_{1,3} & \dots \\ 2\phi_{2,1} - \phi_{2,2} & -\phi_{2,1} + 2\phi_{2,2} - \phi_{2,3} & \dots \\ 2\phi_{3,1} - \phi_{3,2} & -\phi_{3,1} + 2\phi_{3,2} - \phi_{3,3} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix},$$

Then adding the two, yields at each matrix entry the 2D Laplacian stencil, so that the equations (3.30) 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 Φ_h this reads equivalently

$$I_h \Phi_h A_h + A_h \Phi_h I_h = R_h. \quad (3.31)$$

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 [13, 14]. A nice review article on the properties and applications of the Kronecker product was written by Van Loan [18].

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

$$\text{vec}(\Phi_h) = (\phi_{1,1}, \phi_{2,1}, \dots, \phi_{N-1,1}, \phi_{1,2}, \phi_{2,2}, \dots, \phi_{N-1,2}, \phi_{3,1}, \dots, \phi_{N-1,N-1})^\top.$$

We then have for any two matrices B and C of appropriate dimensions and their Kronecker product $B \otimes C$

$$CXB^\top = (B \otimes C)\text{vec}(X).$$

This is all we need to rewrite our 2D discrete Poisson equation using Kronecker products. As A_h is symmetric, (3.31) 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 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 a nD problem into a sequence of 1D problem enabling to obtain some very fast algorithms.

3.5 The Finite Difference method for 1D advection

The Vlasov-Poisson system is a 1D hyperbolic-elliptic system. We have already seen how to deal with the elliptic Poisson equation, let us now investigate numerical methods for hyperbolic problems on the simplest example, which is the 1D advection equation, in a periodic domain.

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad \text{for } x \in [0, L], t \geq 0. \quad (3.32)$$

Let us assume for simplicity that the boundary conditions are periodic. This means that u and all its derivatives are periodic of period L . We have in particular $u(0) = u(L)$. The constant a is given. As the problem is time dependent, we also need an initial condition $u(x, 0) = u_0(x)$.

3.5.1 Obtaining a Finite Difference scheme

We first consider a uniform mesh of the 1D computational domain, i.e. of the interval $[a, b]$ where we want to compute the solution, see Figure 3.4. The cell size or space step

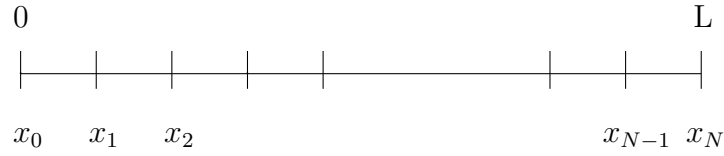


Figure 3.4: Uniform mesh of $[a, b]$

is defined by $\Delta x = \frac{L}{N}$ where N is the number of cells in the mesh. The coordinates of the grid points are then defined by $x_i = x_0 + i\Delta x$. We then need a time step Δt and we will compute approximations of the solution at discrete times $t_n = n\Delta t$, $n \in \mathbb{N}$. As we assume the solution to be periodic of period L it will be defined by its values at x_i for $0 \leq i \leq N - 1$ and we shall have $u(x_N, t_n) = u(x_0, t_n)$.

We shall denote by $u_j^n = u(x_j, t_n)$.

3.5.2 The first order explicit upwind scheme

A Finite Difference scheme is classically obtained by approximating the derivatives appearing in the partial differential equation by a Taylor expansion up to some given order which will give the order of the scheme. As we know only the values of the unknown function at the grid points, we use Taylor expansions at different grid points and linearly combine them so as to eliminate all derivatives up to the needed order.

The same can be done for the time discretisation. For an approximation of order 1 in space and time, we can simply write

$$\frac{\partial u}{\partial t}(x_j, t_n) = \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} + O(\Delta t), \quad (3.33)$$

$$\frac{\partial u}{\partial x}(x_j, t_n) = \frac{u(x_j, t_n) - u(x_{j-1}, t_n)}{\Delta x} + O(\Delta x). \quad (3.34)$$

Denoting by u_j^n , the approximation of the solution at point x_j and time t_n and using the above formulas for the approximation of the partial derivatives we get the following approximation (3.32) at point x_j and time t_n :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0. \quad (3.35)$$

We thus obtain the following explicit formula which enables to compute u_j^{n+1} in function of the values of u at time t_n and points x_{j-1} , x_j and x_{j+1} :

$$u_j^{n+1} = u_j^n - a \frac{\Delta t}{\Delta x} (u_j^n - u_{j-1}^n). \quad (3.36)$$

Denote by U^n the vector of \mathbb{R}^N whose components are u_0^n, \dots, u_{N-1}^n and

$$A = \begin{pmatrix} (1 - \frac{a\Delta t}{\Delta x}) & 0 & & \frac{a\Delta t}{\Delta x} \\ \frac{a\Delta t}{\Delta x} & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ 0 & & \frac{a\Delta t}{\Delta x} & (1 - \frac{a\Delta t}{\Delta x}) \end{pmatrix}.$$

The terms at the end of the first line comes from the periodic boundary conditions. We use that $u_{-1}^n = u_{N-1}^n$ and $u_N^n = u_0^n$. Except on the two diagonals all the terms vanish. So, with this matrix A and denoting the unknown at time t_n $U^n = (u_0^n, \dots, u_{N-1}^n)^\top$ the scheme (3.36) can be written in matrix form

$$U^{n+1} = AU^n.$$

3.5.3 The first order upwind implicit scheme

When using an uncentered difference scheme in the other direction for the time derivative, we get

$$\frac{\partial u}{\partial t}(x_j, t_n) = \frac{u(x_j, t_n) - u(x_j, t_{n-1})}{\Delta t} + O(\Delta t), \quad (3.37)$$

We use the same finite difference approximation for the space derivative. We then get the following formula

$$u_j^n + a \frac{\Delta t}{\Delta x} (u_j^n - u_{j-1}^n) = u_j^{n-1}. \quad (3.38)$$

In this case the u_j^n are defined implicitly from the u_j^{n-1} as solutions of a linear system. This is why this scheme is called implicit.

Denote by B the matrix of the linear system:

$$B = \begin{pmatrix} (1 + \frac{a\Delta t}{\Delta x}) & 0 & & -\frac{a\Delta t}{\Delta x} \\ -\frac{a\Delta t}{\Delta x} & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ 0 & & -\frac{a\Delta t}{\Delta x} & (1 + \frac{a\Delta t}{\Delta x}) \end{pmatrix}.$$

The term at the end of the first line comes from the periodic boundary conditions. We use that $u_{-1}^n = u_{N-1}^n$ and $u_N^n = u_0^n$. The terms not on the two diagonals vanish.

Going now from time step n to $n+1$ the implicit scheme in matrix form becomes

$$BU^{n+1} = U^n.$$

3.5.4 The explicit downwind and centred schemes

Rather than using an upwind approximation of the $\partial_x u(x_j)$, one could in principle also use either downwind or centred finite difference schemes. These read respectively

$$\frac{\partial u}{\partial x}(x_j, t_n) = \frac{u(x_{j+1}, t_n) - u(x_j, t_n)}{\Delta x} + O(\Delta x),$$

for a positive a and

$$\frac{\partial u}{\partial x}(x_j, t_n) = \frac{u(x_{j+1}, t_n) - u(x_{j-1}, t_n)}{2\Delta x} + O(\Delta x^2).$$

This one is the same for positive and negative a , and is of second order in x .

Both those schemes are consistent as they derive from a Taylor approximation, but they cannot be used in practice because they are unstable. The update matrices of these scheme, for a first order explicit method in time read respectively

$$A_{down} = \begin{pmatrix} (1 + \frac{a\Delta t}{\Delta x}) & -\frac{a\Delta t}{\Delta x} & & 0 \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & -\frac{a\Delta t}{\Delta x} \\ -\frac{a\Delta t}{\Delta x} & & 0 & (1 + \frac{a\Delta t}{\Delta x}) \end{pmatrix}, \quad A_{cen} = \begin{pmatrix} 1 & -\frac{a\Delta t}{2\Delta x} & & \frac{a\Delta t}{2\Delta x} \\ \frac{a\Delta t}{2\Delta x} & \ddots & \ddots & \\ & \ddots & \ddots & -\frac{a\Delta t}{\Delta x} \\ -\frac{a\Delta t}{2\Delta x} & & \frac{a\Delta t}{2\Delta x} & 1 \end{pmatrix}. \quad (3.39)$$

3.5.5 Stability and convergence

As for steady-state problems, a scheme is consistent if the exact solution verifies it up to some power in Δx and Δt , which is called the order of consistency in space and time respectively. Stability is defined in the following way:

Definition 4 *A numerical scheme for a time dependent problem is called stable for some given norm $\|\cdot\|$ if there exist constants K and τ independent of Δt such that*

$$\|U^n\| \leq K\|U^0\| \quad \forall \Delta t \text{ such that } 0 < \Delta t < \tau.$$

Theorem 2 (Lax) *A linear scheme is convergent if it is stable and consistent.*

Let us now check the stability in the L^2 norm of our three explicit schemes (upwind, downwind and centred). A useful tool to do this, for periodic boundary conditions is called the *von Neumann stability analysis*. Due to the fact that the discrete Fourier transform conserves the L^2 norm because of the discrete Plancherel inequality and that it diagonalises the Finite Difference operators (provided the original PDE has constant coefficients), it is particularly well adapted for studying the L^2 stability. The von Neumann analysis consists in applying the discrete Fourier transform to the discretised equation. This is equivalent to using the theory of circulant matrices and checking that the modulus of all eigenvalues is smaller than 1. Using the formula of the eigenvalues of our update matrices, we find denoting by $\xi = 2\pi k/N$ for the upwind scheme

$$\lambda_k = 1 - \frac{a\Delta t}{\Delta x} + \frac{a\Delta t}{\Delta x} e^{-\frac{2i\pi k}{N}} = 1 - \frac{a\Delta t}{\Delta x} (1 - \cos \xi - i \sin \xi),$$

so that

$$\begin{aligned}
|\lambda_k|^2 &= \left(1 - \frac{a\Delta t}{\Delta x}(1 - \cos \xi)\right)^2 + \frac{a^2\Delta t^2}{\Delta x^2} \sin^2 \xi, \\
&= 1 - 2\frac{a\Delta t}{\Delta x}(1 - \cos \xi) + \frac{a^2\Delta t^2}{\Delta x^2}(1 - 2\cos \xi + \cos^2 \xi) + \frac{a^2\Delta t^2}{\Delta x^2} \sin^2 \xi, \\
&= 1 - 2\frac{a\Delta t}{\Delta x} \left(1 - \frac{a\Delta t}{\Delta x}\right) (1 - \cos \xi).
\end{aligned}$$

If $0 \leq a\Delta t/\Delta x \leq 1$ all the factors in the second term are positive, so that $|\lambda_k| \leq 1$. Hence we find that the first order explicit upwind scheme is stable provided $a\Delta t/\Delta x \leq 1$. This is a condition on the time step for a given spatial mesh. This condition is the well-known Courant-Friedrichs-Lewy (CFL) condition. If the CFL condition is broken ($a\Delta t > \Delta x$) for $a > 0$, then all the modes except $k = 0$ are unstable $|\lambda_k| > 1$ and the most unstable mode corresponds to $\cos \xi = -1$, *i.e.* $k = N/2$. This is the so-called saw-tooth mode, the eigenvector for $k = N/2$ switches between the values -1 and 1. This is the dominating mode that you will see in a CFL unstable simulation.

For the downwind scheme (which corresponds also to $a < 0$ for the previous case)

$$\lambda_k = 1 + \frac{a\Delta t}{\Delta x} - \frac{a\Delta t}{\Delta x} e^{\frac{2i\pi k}{N}} \Rightarrow \lambda_{N/2} = 1 + 2\frac{a\Delta t}{\Delta x} > 1,$$

for $k = N/2$ so that the scheme is unstable. And for the centred scheme

$$\lambda_k = 1 + \frac{a\Delta t}{2\Delta x} (e^{\frac{2i\pi k}{N}} - e^{\frac{-2i\pi k}{N}}) = 1 + i\frac{a\Delta t}{\Delta x} \sin \xi,$$

which implies obviously that $|\lambda_k| > 1$ whenever $\sin \xi \neq 0$, so that this scheme is also unstable.

One can check similarly that all the corresponding first order in time implicit scheme are stable.

3.6 Splitting methods

In the Vlasov equation without a magnetic field, the advection field in \mathbf{x} , which is \mathbf{v} , does not depend on \mathbf{x} and the advection field in \mathbf{v} , which is $\mathbf{E}(\mathbf{x}, t)$, does not depend on \mathbf{x} . Therefore it is often convenient to decompose these two parts, using the technique called *operator splitting*.

Let us consider the non relativistic Vlasov-Poisson equation which reads

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q}{m} \mathbf{E} \cdot \nabla_{\mathbf{v}} f = 0,$$

coupled with the Poisson equation $-\Delta\phi = 1 - \rho(t, \mathbf{x}) = 1 - \int f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}$, $\mathbf{E}(\mathbf{x}, t) = -\nabla\phi$. Throught this coupling, \mathbf{E} depends on f , which makes the Vlasov-Poisson system non linear.

We shall split the equation into the following two pieces:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f = 0, \tag{3.40}$$

with \mathbf{v} fixed and

$$\frac{\partial f}{\partial t} + \frac{q}{m} \mathbf{E}(\mathbf{x}, t) \cdot \nabla_{\mathbf{v}} f = 0, \tag{3.41}$$

with \mathbf{x} fixed. We then get two constant coefficient advections that can be easier to solve. This is obvious for (3.40) as \mathbf{v} does not depend on t and x . On the other hand, integrating (3.41) with respect to \mathbf{v} , we get that $\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial t} \int f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v} = 0$, so that ρ and consequently \mathbf{E} does not change when this equation is advanced in time. So that $\mathbf{E}(t, x)$ needs to be computed with the initial f for this equation and does then depend neither on t , nor \mathbf{x} .

Remark 4 *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 the following system of equations

$$\frac{du}{dt} = (A + B)u, \quad (3.42)$$

where A and B are any two differential operators (in space), that are assumed constant between t_n and t_{n+1} . The formal solution of this equation on one time step reads:

$$u(t + \Delta t) = e^{\Delta t(A+B)}u(t).$$

Let us split the equation (3.42) into

$$\frac{du}{dt} = Au, \quad (3.43)$$

$$\frac{du}{dt} = Bu. \quad (3.44)$$

The formal solutions of these equations taken separately are

$$u(t + \Delta t) = e^{\Delta t A}u(t) \text{ and } u(t + \Delta t) = e^{\Delta t B}u(t).$$

The standard operator splitting method consists in solving successively on one time step first (3.43) and then (3.44). Then one gets on one time step

$$\tilde{u}(t + \Delta t) = e^{\Delta t B}e^{\Delta t A}u(t).$$

If the operators A and B commute, *i.e.* $AB = BA$, $e^{\Delta t B}e^{\Delta t A} = e^{\Delta t(A+B)}$ and the splitting is exact. This is the case in particular when considering a constant coefficient advection equation of the form

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} = 0.$$

This can be checked using the method of characteristics. Note that such an equation is also a good first test case to validate a Vlasov code.

In the case when A and B do not commute, the splitting error can be decreased by solving first (3.43) on a half time step, and then (3.44) on a full time step and again (3.43) on a half time step. This method is known as the *Strang splitting method*. It corresponds to the formal solution

$$\bar{u}(t + \Delta t) = e^{\frac{\Delta t}{2}A}e^{\Delta t B}e^{\frac{\Delta t}{2}A}u(t).$$

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

Proposition 6 • *The standard splitting method is of order 1 in time.*

• *The Strang splitting method is of order 2 in time.*

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

$$\begin{aligned} e^{\Delta t B} e^{\Delta t A} &= (I + \Delta t B + \frac{\Delta t^2}{2} B^2 + O(\Delta t^3))(I + \Delta t A + \frac{\Delta t^2}{2} A^2 + O(\Delta t^3)) \\ &= I + \Delta t(A+B) + \frac{\Delta t^2}{2}(A^2 + B^2 + 2BA) + O(\Delta t^3). \end{aligned}$$

But as A and B do not commute, we have $(A+B)^2 = A^2 + AB + BA + B^2$. It follows that $e^{\Delta t(A+B)} - e^{\Delta t B} e^{\Delta t A} = O(\Delta t^2)$, which leads to a local error of order 2 and a global error of order 1.

For the Strang splitting method, we have

$$\begin{aligned} e^{\frac{\Delta t}{2} A} e^{\Delta t B} e^{\frac{\Delta t}{2} A} &= (I + \frac{\Delta t}{2} A + \frac{\Delta t^2}{4} A^2 + O(\Delta t^3))(I + \Delta t B + \frac{\Delta t^2}{2} B^2 + O(\Delta t^3)) \\ &\quad (I + \frac{\Delta t}{2} A + \frac{\Delta t^2}{4} A^2 + O(\Delta t^3)) \\ &= I + \Delta t(A+B) + \frac{\Delta t^2}{2}(A^2 + B^2 + BA + AB) + O(\Delta t^3). \end{aligned}$$

We thus obtain a local error of order 3 and thus a global error of order 2 for the method of Strang. ■

Remark 5 *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 [19].*

Remark 6 *The Strang splitting method can also be generalized to more than two operators. If $A = A_1 + \dots + A_n$, the following decomposition will be of global order 2:*

$$e^{\frac{\Delta t}{2} A_1} \dots e^{\frac{\Delta t}{2} A_{n-1}} e^{\Delta t A_n} e^{\frac{\Delta t}{2} A_{n-1}} \dots e^{\frac{\Delta t}{2} A_1}.$$

3.7 The Finite Element Method

3.7.1 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:

$$-\Delta u = f, \quad \text{in } \Omega, \quad u = 0, \quad \text{on } \partial\Omega. \quad (3.45)$$

The first idea, introduced by Ritz in his thesis in Göttingen in 1902, was to transform the boundary problem into an equivalent minimisation problem. Indeed, via the Dirichlet principle (3.45) is equivalent to the minimisation problem

$$\min_{u \in H_0^1(\Omega)} \left(\frac{1}{2} \int_{\Omega} |\nabla u(x)|^2 dx - \int_{\Omega} f(x)u(x) dx \right). \quad (3.46)$$

We shall need the following Hilbert spaces, defined for a domain $\Omega \in \mathbb{R}^d$

$$H^1(\Omega) = \{u \in L^2(\Omega), \nabla u \in (L^2(\Omega))^d\}, \quad H_0^1(\Omega) = \{u \in H^1(\Omega), u = 0 \text{ on } \partial\Omega\}.$$

The scalar product associated to these Hilbert spaces is

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

Then, the original problem being transformed into a minimisation problem in becomes quite natural to look for an approximation in a finite dimensional subspace of the function space in which the minimisation problem is posed (in our case $H_0^1(\Omega)$), which means that the minimisation is performed by considering only minima in a finite dimensional subspace. Then if the form of 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. 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.

The standard method to solve a minimisation problem with a cost functional J defined on a Hilbert space V , of the form

$$\min_{u \in V} J[u],$$

is to solve the associated Euler equation $J'[u] = 0$ obtained by computing the Fréchet derivative of the functional that we want to minimise. Note that the Fréchet derivative gives a rigorous definition of the functional derivative used in physics for functions that are in a Banach (including Hilbert) space. Consider a functional J from a Hilbert space V into \mathbb{R} . Its Fréchet derivative J' , assuming it exists, is a linear form on V , which means that it maps any function from V to a scalar. It can be computed using the Gâteaux formula:

$$J'[u](v) = \lim_{\varepsilon \rightarrow 0} \frac{J[u + \varepsilon v] - J[u]}{\varepsilon}. \quad (3.47)$$

Let us apply this formula to our problem for which

$$J[u] = \frac{1}{2} \int_{\Omega} |\nabla u(x)|^2 dx - \int_{\Omega} f(x)u(x) dx.$$

We have for any $v \in V = H_0^1(\Omega)$

$$\begin{aligned} J[u + \varepsilon v] &= \frac{1}{2} \int_{\Omega} |\nabla u(x) + \varepsilon v(x)|^2 dx - \int_{\Omega} f(x)(u(x) + \varepsilon v(x)) dx \\ &= \frac{1}{2} \left(\int_{\Omega} |\nabla u(x)|^2 dx + 2\varepsilon \int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx + \varepsilon^2 \int_{\Omega} |\nabla v(x)|^2 dx \right) \\ &\quad - \int_{\Omega} f(x)u(x) dx - \varepsilon \int_{\Omega} f(x)v(x) dx. \end{aligned}$$

From which we deduce, using the Gâteaux formula (3.47) that

$$J'[u](v) = \int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx - \int_{\Omega} f(x)v(x) dx.$$

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 (3.46), is a solution of the Euler equation $J'[u] = 0$ or equivalently $J'[u](v) = 0$ for all $v \in V$, which reads

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx = \int_{\Omega} f(x)v(x) dx \quad \forall v \in H_0^1(\Omega). \quad (3.48)$$

This is also what is called the variational formulation, or the weak formulation of the original boundary value problem (3.45). Note that this variational formulation expresses in some sense the orthogonality of the residual to the space in which the solution is sought. This is more general than Euler equations 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 in a finite dimensional subspace $V_h \subset V$ of the original space and to use the same variational formulation (3.48) as the one defining the exact solution, with test functions also in V_h to characterise the solution. 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 (3.48).

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 known 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 $(1, x, x^2, \dots)$ has very bad properties. Best approximations are provided by the orthogonal Legendre polynomials or by the Chebyshev 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 being 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. An example of an unstructured Finite Element mesh in 2D is displayed in Figure 3.5, 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.

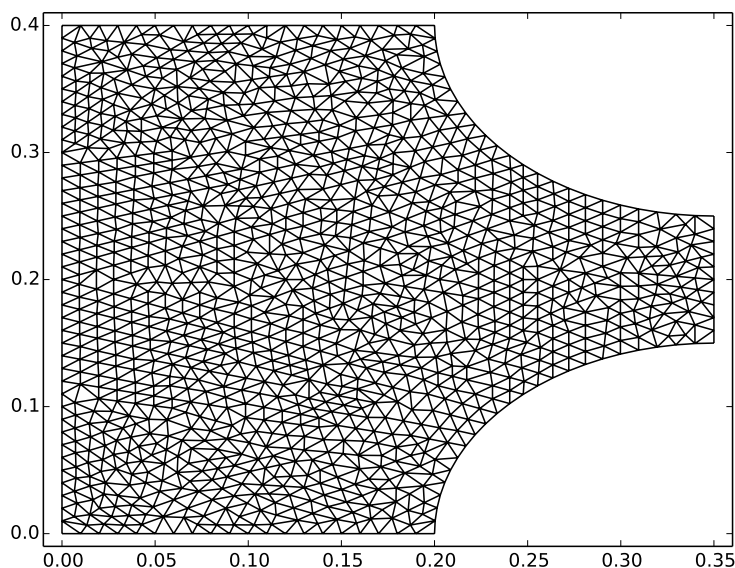


Figure 3.5: Example of a 2D finite element mesh consisting of triangles.

The article by Gander and Wanner [6] 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 [15].

In summary, the finite element method consists in looking for a solution of a variational problem like (3.48), in a finite dimensional subspace V_h of the space V where the exact solution is defined. The space V_h is characterised by a basis $(\varphi_1, \dots, \varphi_N)$ so that finding the solution of the variational problem amounts to solving a linear system. Indeed, express the trial function u_h and the test function v_h on this basis:

$$u_h(x) = \sum_{j=1}^N u_j \varphi_j(x), \quad v_h(x) = \sum_{i=1}^N v_i \varphi_i(x),$$

and plug these expressions in the variational problem (3.48). This yields

$$\sum_{i=1}^N \sum_{j=1}^N u_j v_i \int_{\Omega} \nabla \varphi_i(x) \cdot \nabla \varphi_j(x) \, dx = \sum_{i=1}^N v_i \int_{\Omega} f(x) \varphi_i(x) \, dx.$$

This can be expressed in matrix form, $\tilde{U}_h^\top A_h U_h = \tilde{U}_h^\top b_h$, which is equivalent to the linear system $A_h U_h = b_h$ as the previous equality is true for all \tilde{U}_h , where

$$U_h = (u_1, \dots, u_N)^\top, \tilde{U}_h = (v_1, \dots, v_N)^\top, b_h = \left(\int_{\Omega} f(x) \varphi_1(x) \, dx, \dots, \int_{\Omega} f(x) \varphi_N(x) \, dx \right)^\top$$

and the matrix A_h whose entries are

$$\left(\int_{\Omega} \nabla \varphi_i(x) \cdot \nabla \varphi_j(x) \, dx \right)_{1 \leq i, j \leq N}.$$

3.7.2 The variational (or weak) form of a boundary value problem

The variational 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 variational 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 variational formulation, we will need the following Green formula: For $u \in H^2(\Omega)$ and $v \in H^1(\Omega)$

$$-\int_{\Omega} \Delta u \, v \, dx = \int_{\Omega} \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\sigma. \quad (3.49)$$

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

Case of homogeneous Dirichlet boundary conditions

Let $f \in L^2(\Omega)$. Consider the boundary value problem

$$-\Delta u = f \quad \text{in } \Omega, \quad (3.50)$$

$$u = 0 \quad \text{on } \partial\Omega. \quad (3.51)$$

Assume that $u \in H^2(\Omega)$, multiply (3.50) by $v \in H^1(\Omega)$ and integrate using the Green formula (3.49), 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 u does not appear 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. This is the case of an essential boundary condition. So we take test functions v vanishing on the boundary. We then get the following variational formulation:
Find $u \in H_0^1(\Omega)$ such that

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx \quad \forall v \in H_0^1(\Omega). \quad (3.52)$$

The solutions of this variational formulation 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 that such a solution is also a solution of the initial boundary value problem (3.50)-(3.51). If $u \in H^2(\Omega)$, the Green formula (3.49) can be used, and as φ vanishes on the boundary it yields

$$-\int_{\Omega} \Delta u \, \varphi \, dx = \int_{\Omega} f \varphi \, dx \quad \forall \varphi \in H_0^1(\Omega).$$

This implies, as $H_0^1(\Omega)$ is dense in $L^2(\Omega)$, that $-\Delta u = f$ in $L^2(\Omega)$ and so almost everywhere. On the other hand as $u \in H_0^1(\Omega)$, $u = 0$ on $\partial\Omega$. So u is a strong solution of (3.50)-(3.51).

Case of non homogeneous Dirichlet boundary conditions

Let $f \in L^2(\Omega)$ and $u_0 \in H^1(\Omega)$. We consider the problem

$$-\Delta u = f \quad \text{in } \Omega, \quad (3.53)$$

$$u = u_0 \quad \text{on } \partial\Omega. \quad (3.54)$$

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 $\tilde{u} = u - u_0$. We then show as previously that \tilde{u} is a solution of the variational problem
Find $\tilde{u} \in H_0^1(\Omega)$ such that

$$\int_{\Omega} \nabla \tilde{u} \cdot \nabla v \, dx = \int_{\Omega} f v \, dx - \int_{\Omega} \nabla u_0 \cdot \nabla v \, dx \quad \forall v \in H_0^1(\Omega). \quad (3.55)$$

This is the variational 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.

Case of Neumann boundary conditions

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

$$-\Delta u + u = f \quad \text{in } \Omega, \quad (3.56)$$

$$\frac{\partial u}{\partial n} = g \quad \text{on } \partial\Omega. \quad (3.57)$$

Assuming that $u \in H^2(\Omega)$, we multiply by a test function $v \in H^1(\Omega)$ and integrate using the Green formula (3.49), which yields

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\sigma + \int_{\Omega} uv \, dx = \int_{\Omega} fv \, dx.$$

Replacing $\frac{\partial u}{\partial n}$ by its value g on the boundary, we obtain the variational formulation Find $u \in H^1(\Omega)$ such that

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx + \int_{\Omega} uv \, dx = \int_{\Omega} fv \, dx + \int_{\partial\Omega} gv \, d\sigma \quad \forall v \in H^1(\Omega). \quad (3.58)$$

Let us now show that u is a strong solution of the boundary value problem, provided it is in $H^2(\Omega)$. As $H_0^1(\Omega) \subset H^1(\Omega)$ one can first take only test functions in $H_0^1(\Omega)$. Then as in the case of homogeneous Dirichlet conditions it follows from the Green formula (3.49) that

$$\int_{\Omega} (-\Delta u + u) \varphi \, dx = \int_{\Omega} f \varphi \, dx \quad \forall \varphi \in H_0^1(\Omega).$$

This implies, as $H_0^1(\Omega)$ 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

$$\frac{\partial u}{\partial n} = g \text{ on } \partial\Omega.$$

For that we start from (3.58) and apply the Green formula (3.49), which yields

$$-\int_{\Omega} \Delta u v \, dx + \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\sigma + \int_{\Omega} uv \, dx = \int_{\Omega} fv \, dx + \int_{\partial\Omega} gv \, d\sigma \quad \forall v \in H^1(\Omega),$$

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

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

which yields that $\frac{\partial u}{\partial n} = g$ on $\partial\Omega$.

3.7.3 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, whence 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).

Let us start with a non uniform 1D mesh of the domain $[a, b]$ defined by the grid points $a = x_0 < x_1 < \dots < x_N = b$. The elements of the mesh are here the intervals $[x_\nu, x_{\nu+1}]$. The restriction of V_h to each element is defined to be the space of polynomials of degree k , denote by $\mathbb{P}_k([x_\nu, x_{\nu+1}])$. The basis restricted to each element is defined

via a reference element, which is conveniently chosen, for later Gauss integration, as the interval $[-1, 1]$ and an affine map

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

$$\hat{x} \mapsto \frac{x_\nu + x_{\nu+1}}{2} + \left(\frac{x_{\nu+1} - x_\nu}{2} \right) \hat{x}.$$

An important aspect when choosing the local basis of a finite element is the global continuity requirement coming from the fact that $V_h \subset V$. Indeed a function with discontinuities is not in H^1 , this is why 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 $\mathbb{P}_k([x_\nu, x_{\nu+1}])$ as being defined by its value at $k + 1$ points in $[x_\nu, x_{\nu+1}]$, including the endpoints of the interval x_ν and $x_{\nu+1}$. Such a basis is called a *nodal basis* and the naturally associated basis functions are the Lagrange basis functions.

So the subspace V_h on the mesh $x_0 < x_1 < \dots < x_N$ is defined by

$$V_h = \{v_h \in C^0([a, b]) \mid v_h|_{[x_\nu, x_{\nu+1}]} \in \mathbb{P}_k([x_\nu, x_{\nu+1}])\}.$$

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 $-1 = y_0 < y_1 < \dots < y_k = 1$ the Lagrange basis functions of degree k denoted by $l_{k,i}$, $0 \leq i \leq k$, are the unique polynomials of degree k verifying $l_{k,i}(y_j) = \delta_{i,j}$. Because of this property, any polynomial $p(x) \in \mathbb{P}_k([-1, 1])$ can be expressed as $p(x) = \sum_{j=0}^k p(y_j) l_{j,k}(x)$ and conversely any polynomial $p(x) \in \mathbb{P}_k([-1, 1])$ is uniquely determined by its values at the interpolation points y_j , $0 \leq j \leq k$. Hence in order to ensure the continuity of the piecewise polynomial at the cell interface x_ν it is enough that the values of the polynomials on both sides of x_ν have the same value at x_ν . This constraint removes one degree of freedom in each cell, moreover the two end points are known for Dirichlet boundary conditions, which removes two other degrees of freedom so that the total dimension of V_h is $Nk - 1$ and 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 denoted of V_h denoted by $(\varphi_i)_{0 \leq i \leq Nk-1}$ are such that their restriction on each cell is a Lagrange basis function.

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. For stability and conveniency issues this are most commonly taken to be the Gauss-Lobatto points on each cell. We are now ready to assemble the linear system

The discrete variational problem in 1D reads: *Find $u_h \in V_h$ such that*

$$\int_a^b \frac{du_h}{dx} \frac{dv_h}{dx} dx = \int_a^b f(x) v_h(x) dx \quad \forall v_h \in V_h.$$

Now expressing u_h (and v_h) in the basis of V_h as $u_h(x) = \sum_{j=1}^{Nk-1} u_j(t) \varphi_j(x)$, $v_h(x) = \sum_{j=1}^{Nk-1} v_j \varphi_j(x)$ and plugging these expression in the variational formulation, denoting by $U = (u_1, \dots, u_{Nk-1})^\top$ and similarly for V yields: *Find $U \in \mathbb{R}^{Nk-1}$ such that*

$$\sum_{i,j} u_j v_i \int_a^b \frac{\partial \varphi_i(x)}{\partial x} \frac{\partial \varphi_j(x)}{\partial x} dx = \sum_{i,j} v_i \int_a^b f(x) \varphi_i(x) dx \quad \forall V \in \mathbb{R}^{Nk-1},$$

which can be expressed in matrix form

$$V^\top AU = V^\top b \quad \forall V \in \mathbb{R}^{nk},$$

which is equivalent to

$$AU = b$$

where the square $(Nk - 1) \times (Nk - 1)$ matrix A and right hand side b are defined by

$$A = \left(\int_0^L \frac{d\varphi_i(x)}{dx} \frac{d\varphi_j(x)}{dx} dx \right)_{i,j}, \quad b = \left(\int_0^L f(x) \varphi_i(x) dx \right)_i.$$

Another option for computing the right-hand side and which yields the same order of approximation is to project first the unknown f onto f_h in the space V_h , then $f_h(x) = \sum_i f_i \varphi_i(x)$, and the right hand side can be approximated with $\tilde{b} = MF$, with F the vector of components f_i and the mass matrix

$$M = \left(\int_0^L \varphi_i(x) \varphi_j(x) dx \right)_{i,j}.$$

Note that these matrices can be computed exactly as they involve integration of polynomials on each cell. Moreover because the Gauss-Lobatto quadrature rule is exact for polynomials of degree up to $2k-1$, A can be computed exactly with the Gauss-Lobatto quadrature rule. Moreover, approximating the mass matrix M with the Gauss-Lobatto rule introduces an error which does not decrease the order of accuracy of the scheme [4] and has the big advantage of yielding a diagonal matrix. This is what is mostly done in practice.

Usually for Finite Elements the matrices M and A are computed from the corresponding elementary matrices which are obtained by change of variables onto the reference element $[-1, 1]$ for each cell. So

$$\int_0^L \varphi_i(x) \varphi_j(x) dx = \sum_{\nu=0}^{n-1} \int_{x_\nu}^{x_{\nu+1}} \varphi_i(x) \varphi_j(x) dx,$$

and doing the change of variable $x = \frac{x_{\nu+1}-x_\nu}{2} \hat{x} + \frac{x_{\nu+1}+x_\nu}{2}$, we get

$$\int_{x_\nu}^{x_{\nu+1}} \varphi_i(x) \varphi_j(x) dx = \frac{x_{\nu+1} - x_\nu}{2} \int_{-1}^1 \hat{\varphi}_\alpha(\hat{x}) \hat{\varphi}_\beta(\hat{x}) d\hat{x},$$

where $\hat{\varphi}_\alpha(\hat{x}) = \varphi_i(\frac{x_{\nu+1}-x_\nu}{2} \hat{x} + \frac{x_{\nu+1}+x_\nu}{2})$. The local indices α on the reference element go from 0 to k and the global numbers of the basis functions not vanishing on element ν are $j = k\nu + \alpha$. The $\hat{\varphi}_\alpha$ are the Lagrange polynomials at the Gauss-Lobatto points in the interval $[-1, 1]$.

The mass matrix in V_h can be approximated with no loss of order of the finite element approximation using the Gauss-Lobatto quadrature rule. Then because the products $\hat{\varphi}_\alpha(\hat{x}) \hat{\varphi}_\beta(\hat{x})$ vanish for $\alpha \neq \beta$ at the Gauss-Lobatto points by definition of the $\hat{\varphi}_\alpha$ which are the Lagrange basis functions at these points, the elementary matrix M is diagonal and we have

$$\int_{-1}^1 \hat{\varphi}_\alpha(\hat{x})^2 d\hat{x} \approx \sum_{\beta=0}^k w_\beta^{GL} \varphi_\alpha(\hat{x}_\beta)^2 = w_\alpha^{GL}$$

using the quadrature rule, where w_α^{GL} is the Gauss-Lobatto weight at Gauss-Lobatto point $(\hat{x}_\alpha) \in [-1, 1]$. So that finally $\hat{M} = \text{diag}(w_0^{GL}, \dots, w_k^{GL})$ is the matrix with $k + 1$ lines and columns with the Gauss-Lobatto weights on the diagonal.

Let us now compute the elements of A . As previously we go back to the interval $[-1, 1]$ with the change of variables $x = \frac{x_{\nu+1}-x_\nu}{2}\hat{x} + \frac{x_{\nu+1}+x_\nu}{2}$ and we define $\hat{\varphi}_\alpha(\hat{x}) = \varphi_i(\frac{x_{\nu+1}-x_\nu}{2}\hat{x} + \frac{x_{\nu+1}+x_\nu}{2})$. Note that a global basis function φ_i associated to a grid point has a support which overlaps two cells and is associated to two local basis functions. Thus one needs to be careful to add the two contributions as needed in the final matrix.

We get $\hat{\varphi}'_\alpha(\hat{x}) = \frac{x_{\nu+1}-x_\nu}{2}\varphi'_i(\frac{x_{\nu+1}-x_\nu}{2}(\hat{x} + 1) + x_\nu)$. It follows that

$$\begin{aligned} \int_{x_\nu}^{x_{\nu+1}} \varphi'_j(x)\varphi'_i(x) dx &= \int_{-1}^1 \left(\frac{2}{x_{\nu+1} - x_\nu} \right)^2 \hat{\varphi}'_\beta(\hat{x})\hat{\varphi}'_\alpha(\hat{x}) \frac{x_{\nu+1} - x_\nu}{2} d\hat{x} \\ &= \frac{2}{x_{\nu+1} - x_\nu} \int_{-1}^1 \hat{\varphi}'_\beta(\hat{x})\hat{\varphi}'_\alpha(\hat{x}) d\hat{x} = \frac{2}{x_{\nu+1} - x_\nu} \sum_{m=0}^k w_m^{GL} \hat{\varphi}'_\beta(\hat{x}_m)\hat{\varphi}'_\alpha(\hat{x}_m). \end{aligned}$$

As the polynomial being integrated is of degree $2(k - 1) = 2k - 2$ the Gauss-Lobatto quadrature rule with $k + 1$ points is exact for the product which is of order $2k - 1$. Using this rule

$$\int_{-1}^1 \hat{\varphi}'_\beta(\hat{x})\hat{\varphi}'_\alpha(\hat{x}) d\hat{x} = \sum_{m=0}^k w_m^{GL} \hat{\varphi}'_\beta(\hat{x}_m)\hat{\varphi}'_\alpha(\hat{x}_m) = w_\alpha^{GL} \hat{\varphi}'_\beta(\hat{x}_\alpha),$$

As before, because $\hat{\varphi}_\alpha$ are the Lagrange polynomials at the Gauss-Lobatto points, only the value at x_α in the sum is one and the others are 0. On the other hand evaluating the derivatives of the Lagrange polynomial at the Gauss-Lobatto points at these Gauss-Lobatto points can be done using the formula

$$\hat{\varphi}'_\alpha(\hat{x}_\beta) = \frac{p_\beta/p_\alpha}{\hat{x}_\beta - \hat{x}_\alpha} \text{ for } \beta \neq \alpha \text{ and } \hat{\varphi}'_\alpha(\hat{x}_\alpha) = - \sum_{\beta \neq \alpha} \hat{\varphi}'_\beta(\hat{x}_\alpha),$$

where $p_\alpha = \prod_{\beta \neq \alpha} (\hat{x}_\alpha - \hat{x}_\beta)$. This formula is obtained straightforwardly by taking the derivative of the explicit formula for the Lagrange polynomial

$$\hat{\varphi}_\alpha(\hat{x}) = \frac{\prod_{\beta \neq \alpha} (\hat{x} - \hat{x}_\beta)}{\prod_{\beta \neq \alpha} (\hat{x}_\alpha - \hat{x}_\beta)}$$

and using this expression at the Gauss-Lobatto point $\hat{x}_\beta \neq \hat{x}_\alpha$. We refer to [1] for a detailed description.

This can be extended via Kronecker product to tensor product meshes in 2D or 3D or more.

3.7.4 B-spline Finite Elements

Let us now construct a different kind of Finite Element discretization using B-Splines as basis functions.

In order to define a family of n B-splines 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 5 (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:

$$N_j^k(x) = w_j^k(x)N_j^{k-1}(x) + (1 - w_{j+1}^k(x))N_{j+1}^{k-1}(x)$$

where,

$$w_j^k(x) = \frac{x - x_j}{x_{j+k} - x_j} \quad N_j^0(x) = \chi_{[x_j, x_{j+1}]}(x)$$

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 $[x_j, \dots, x_{j+k+1}]$
- 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 $\mathcal{C}^{(k-m)}$ at x_i .

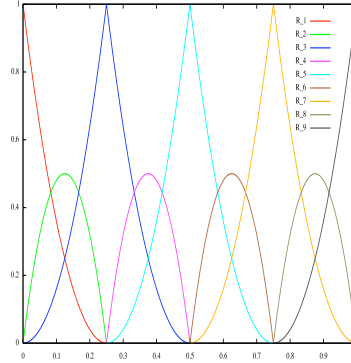


Figure 3.6: All B-splines functions associated to a knot sequence defined by $n = 9$, $k = 2$, $T = \{0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1\}$

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

$$N_i^{k'}(x) = k \left(\frac{N_i^{k-1}(x)}{x_{i+k} - x_i} - \frac{N_{i+1}^{k-1}(x)}{x_{i+k+1} - x_{i+1}} \right). \quad (3.59)$$

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

$$N_i^{k'}(x) = D_i^k(x) - D_{i+1}^k(x). \quad (3.60)$$

Remark 7 In the case where all knots, except the boundary knots are of multiplicity 1, the set $(N_i^k)_{0 \leq i \leq n-1}$ of B-splines of degree k forms a basis of the spline space defined by

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

The boundary knots are chosen to have multiplicity $k+1$ so that the spline becomes interpolatory on the boundary in order to simplify the application of Dirichlet boundary conditions.

Then due to the definitions it follows immediately that $(D_i^k)_{1 \leq i \leq n-1}$ is a basis of \mathcal{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.

Remark 8 Splines can be easily defined in the case of periodic boundary conditions by taking a periodic knot sequence.

Assuming only knots of multiplicity 1 and denoting by $\mathcal{S}_\#^k$ the set of periodic splines associated to a periodic knot sequences, we can take $V_k = \mathcal{S}_\#^k$ whose basis functions are the N_i^k and $W_k = \mathcal{S}_\#^{k-1}$ with basis functions the D_i^k . This defines the Finite Element spaces that we can use with the discrete variational formulation of Poisson's equations, e.g. (3.52) for Dirichlet boundary conditions. We can then construct the mass and stiffness matrices like for the Lagrange Finite Elements, just replacing the basis functions by their spline counterparts, i.e. φ_i by N_i^k and ψ_i by D_i^k . The matrices can be computed with no quadrature error using adequate Gauss or Gauss-Lobatto formulas.

3.7.5 Convergence of the Finite Element method

The variational problems we consider can be written in the following abstract form

Find $u \in V$ such that

$$a(u, v) = l(v) \quad \forall v \in V, \quad (3.61)$$

where V is a Hilbert space, a is a symmetric continuous and coercive bilinear form and l 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 3 (Lax-Milgram) Let V a Hilbert space with the norm $\|\cdot\|_V$. Let $a(\cdot, \cdot)$ continuous, symmetric and coercive bilinear form on $V \times V$, i.e.

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

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

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

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

Let $l(\cdot)$ 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

$$a(u, v) = l(v) \quad \forall v \in V.$$

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 :

Find $u_h \in V_h$ such that

$$a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_h, \quad (3.62)$$

where $V_h \subset V$ is a vector space of dimension N . Let $(\varphi_1, \dots, \varphi_N)$ 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 $v_h = \varphi_i$ the equation (3.62) becomes using the linearity

$$\sum_{j=1}^N u_j a(\varphi_j, \varphi_i) = l(\varphi_i).$$

Then using the symmetry of a , we notice that the discrete variational formulation (3.62) is equivalent to the linear system

$$AU_h = L, \quad (3.63)$$

where $A = (a(\varphi_i, \varphi_j))_{1 \leq i, j \leq N}$, L is the column vector with components $l(\varphi_i)$ and U is the column vector with the unknowns u_i that are the coefficients of u_h in the basis $(\varphi_1, \dots, \varphi_N)$.

Theorem 4 Assume that a is a symmetric continuous and coercive bilinear form on a Hilbert space V and l a continuous linear form on V . Then the system (3.63) is equivalent to the discrete variational form (3.62) 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 $(\varphi_1, \dots, \varphi_N)$.

- Thanks to the bilinearity of a and the linearity of l the relation (3.62) can be written equivalently

$${}^t\tilde{V}AU_h = {}^t\tilde{V}L \quad \forall \tilde{V} \in \mathbb{R}^N, \quad (3.64)$$

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 (3.63) implies (3.64) and so (3.62).

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

$${}^t\tilde{V}A\tilde{V} = a(v_h, v_h) \geq \alpha \|v_h\|^2 \geq 0,$$

and ${}^t\tilde{V}A\tilde{V} = 0 = a(v_h, v_h) \Rightarrow \|v_h\| = 0$, which implies that $v_h = 0$ and so $\tilde{V} = 0$. So A is symmetric, positive definite and therefore invertible.

■

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 (C  a) *Let $u \in V$ the solution of (3.61) and $u_h \in V_h$ the solution of (3.62), with $V_h \subset V$. Then*

$$\|u - u_h\| \leq C \inf_{v \in V_h} \|u - v\|.$$

Proof. We have

$$\begin{aligned} a(u, v) &= l(v) \quad \forall v \in V, \\ a(u_h, v_h) &= l(v_h) \quad \forall v_h \in V_h, \end{aligned}$$

as $V_h \subset V$, we can take $v = v_h$ in the first equality and take the difference which yields

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

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 β such that

$$\begin{aligned} \alpha \|u - u_h\|^2 &\leq a(u - u_h, u - u_h) && \text{as } a \text{ is coercive,} \\ &\leq a(u - u_h, u - v_h) \quad \forall v_h \in V_h, \\ &\leq \beta \|u - u_h\| \|u - v_h\| && \text{as } a \text{ is continuous.} \end{aligned}$$

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

For the global error estimates, we make the following hypotheses on the triangulation \mathcal{T}_h :

(H1) We assume that the family of triangulations is regular in the following sense:

(i) There exists a constant σ such that

$$\forall K \in \cup_h \mathcal{T}_h \quad \frac{h_K}{\rho_K} \leq \sigma.$$

(ii) The quantity $h = \max_{K \in \mathcal{T}_h} h_K$ tend to 0.

(H2) All finite elements (K, P, Σ) , $K \in \cup_h \mathcal{T}_h$ are affine equivalent to a unique reference element $(\hat{K}, \hat{P}, \hat{\Sigma})$.

(H3) All finite elements (K, P, Σ) , $K \in \cup_h \mathcal{T}_h$ are of class C^0 .

Theorem 5 *We assume the hypotheses (H1), (H2) and (H3) are verified. Moreover we assume that there exists an integer $k \geq 1$ such that*

$$\begin{aligned} \mathbb{P}_k &\subset \hat{P} \subset H^1(\hat{K}), \\ H^{k+1}(\hat{K}) &\subset C^0(\hat{K}) \quad (\text{true if } k+1 > \frac{n}{2}). \end{aligned}$$

Then there exists a constant C independent of h such that for any function $v \in H^{k+1}(\Omega)$ we have

$$\|v - \pi_h v\|_{k+1} \leq Ch^k |v|_{k+1, \Omega},$$

where π_h is the finite element interpolation operator defined by

$$\pi_h v = \sum_{i=1}^N v(x_i) p_i.$$

We consider a variational problem posed in $V \subset H^1(\Omega)$.

Theorem 6 *We assume that (H1), (H2) and (H3) are verified. Moreover we assume that there exists an integer $k \geq 1$ such that $k + 1 > \frac{n}{2}$ with $\mathbb{P}_k(\hat{K}) \subset P \subset H^1(\hat{K})$ and that the exact solution of the variational problem is in $H^{k+1}(\Omega)$, then*

$$\|u - u_h\|_{1,\Omega} \leq Ch^k |u|_{k+1,\Omega},$$

where $u_h \in V_h$ is the discrete solution.

Proof. We have because of the polynomial approximation theorem

$$\|u - \pi_h u\|_{1,\Omega} \leq Ch^k |u|_{k+1,\Omega}.$$

On the other hand Céa's lemma gives us

$$\|u - u_h\|_{1,\Omega} \leq C \inf_{v_h \in V_h} \|u - v_h\|_{1,\Omega} \leq C \|u - \pi_h u\|_{1,\Omega}.$$

The result follows. ■

Chapter 4

Numerical methods for the Vlasov-Poisson equations

4.1 The particle in cell (PIC) method

Due to its simplicity and its efficiency in high dimensions, the most used method is still the particle in cell method, which consists in drawing randomly a finite number of origins for the characteristics and follow them in time by solving the equations of motion. These need the electric field which is in turn computed on a grid, using any standard grid based method for the Poisson equation, in general Finite Difference, Fourier spectral or Finite Element. The distribution function is then approximated by a sum of Dirac masses

$$f_h(t, \mathbf{x}, \mathbf{v}) = \sum_k w_k \delta(\mathbf{x} - \mathbf{x}_k(t)) \delta(\mathbf{v} - \mathbf{v}_k(t)).$$

Note that the charge density, source of the Poisson equation, needs to be computed from the particles. A crucial part is the particle mesh coupling. In Finite Element methods, the Finite Element basis function provide a natural way to express the electric fields everywhere in space and also the weak formulation of the right-hand-side is compatible with the expression of the distribution function as a sum of Dirac masses.

4.1.1 Time scheme for the particles

Let us consider first only the case when the magnetic field vanishes (Vlasov-Poisson). Then the macro-particles obey the following equations of motion:

$$\frac{d\mathbf{x}_k}{dt} = \mathbf{v}_k, \quad \frac{d\mathbf{v}_k}{dt} = \frac{q}{m} \mathbf{E}(\mathbf{x}_k, t).$$

This 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. We assume \mathbf{x}_k^n , \mathbf{v}_k^n and \mathbf{E}_k^n known.

$$\mathbf{v}_k^{n+\frac{1}{2}} = \mathbf{v}_k^n + \frac{q\Delta t}{2m} \mathbf{E}_k^n(\mathbf{x}_k^n), \quad (4.1)$$

$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + \Delta t \mathbf{v}_k^{n+\frac{1}{2}}, \quad (4.2)$$

$$\mathbf{v}_k^{n+1} = \mathbf{v}_k^{n+\frac{1}{2}} + \frac{q\Delta t}{2m} \mathbf{E}_k^{n+1}(\mathbf{x}_k^{n+1}). \quad (4.3)$$

We notice that step (4.3) needs the electric field at time t_{n+1} . It can be computed after step (4.2) by solving the Poisson equation which uses as input ρ_h^{n+1} that needs only \mathbf{x}_k^{n+1} and not \mathbf{v}_k^{n+1} .

4.1.2 Particle mesh coupling for Finite Elements

The coupling between mesh and particles is obtained in a natural way in the Finite Element method. Indeed once the degrees of freedom have been computed, the electrostatic potential is given by

$$\phi_h(t, \mathbf{x}) = \sum_{j=1}^{N_g} \phi_j(t) \Lambda_j(\mathbf{x}). \quad (4.4)$$

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

$$\mathbf{E}_h(t, \mathbf{x}_k) = \sum_{j=1}^{N_g} \phi_j(t) \nabla \Lambda_j(\mathbf{x}_k).$$

On the other hand, the weak form of the Poisson equation reads

$$\int \nabla \phi_h \cdot \nabla \psi \, d\mathbf{x} = n_0 - \int f_h(t, \mathbf{x}, \mathbf{v}) \, d\mathbf{v} = \sum_{k=1}^{N_p} w_k \psi(\mathbf{x}_k). \quad (4.5)$$

4.1.3 Particle-Mesh coupling for point based Poisson solvers

The particle approximation f_h of the distribution function does not naturally give an expression for this function at all points of phase space. Thus for the coupling with the field solver which is defined on the mesh a regularizing step is necessary. To this aim we need to define convolution kernels which can be used in this regularization procedure. On cartesian meshes B-splines are mostly used as this convolution kernel. B-splines can be defined recursively: The degree 0 B-spline that we shall denote by S^0 is defined by

$$S^0(x) = \begin{cases} \frac{1}{\Delta x} & \text{if } -\frac{\Delta x}{2} \leq x < \frac{\Delta x}{2}, \\ 0 & \text{else.} \end{cases}$$

Higher order B-splines are then defined by:

For all $m \in \mathbb{N}^*$,

$$\begin{aligned} S^m(x) &= (S^0)^{*m}(x), \\ &= S^0 * S^{m-1}(x), \\ &= \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} S^{m-1}(u) \, du. \end{aligned}$$

In particular the degree 1 spline is

$$S^1(x) = \begin{cases} \frac{1}{\Delta x} (1 - \frac{|x|}{\Delta x}) & \text{si } |x| < \Delta x, \\ 0 & \text{sinon,} \end{cases}$$

the degree 2 spline is

$$S^2(x) = \frac{1}{\Delta x} \begin{cases} \frac{1}{2} \left(\frac{3}{2} - \frac{|x|}{\Delta x} \right)^2 & \text{si } \frac{1}{2} \Delta x < |x| < \frac{3}{2} \Delta x, \\ \frac{3}{4} - \left(\frac{x}{\Delta x} \right)^2 & \text{si } |x| < \frac{1}{2} \Delta x, \\ 0 & \text{sinon,} \end{cases}$$

the degree 3 spline is

$$S^3(x) = \frac{1}{6\Delta x} \begin{cases} \left(2 - \frac{|x|}{\Delta x} \right)^3 & \text{si } \Delta x \leq |x| < 2\Delta x, \\ 4 - 6 \left(\frac{x}{\Delta x} \right)^2 + 3 \left(\frac{|x|}{\Delta x} \right)^3 & \text{si } 0 \leq |x| < \Delta x, \\ 0 & \text{sinon.} \end{cases}$$

B-splines verify the following important properties

Proposition 7 • *Unit mean*

$$\int S^m(x) dx = 1.$$

- *Partition of unity.* For $x_j = j\Delta x$,

$$\Delta x \sum_j S^m(x - x_j) = 1.$$

- *Parity*

$$S^m(-x) = S^m(x).$$

The sources for Maxwell's equations ρ_h and \mathbf{J}_h are defined from the numerical distribution function f_h . In order to be able to defined them at the grid points, we apply the convolution kernel S to define them at any point of space and in particular at the grid points:

$$\rho_h(\mathbf{x}, t) = \int S(\mathbf{x} - \mathbf{x}') f_h(t, \mathbf{x}', \mathbf{v}') d\mathbf{x}' d\mathbf{v}' = q \sum_k w_k S(\mathbf{x} - \mathbf{x}_k), \quad (4.6)$$

$$\mathbf{J}_h(\mathbf{x}, t) = \int S(\mathbf{x} - \mathbf{x}') \mathbf{v} f_h(t, \mathbf{x}', \mathbf{v}') d\mathbf{x}' d\mathbf{v}' = q \sum_k w_k S(\mathbf{x} - \mathbf{x}_k) \mathbf{v}_k. \quad (4.7)$$

In order to get conservation of total momentum, when a regularization kernel is applied to the particles, the same kernel needs to be applied to the field seen as Dirac masses at the grid points in order to compute the field at the particle positions. We then obtain

$$\mathbf{E}_h(\mathbf{x}, t) = \sum_j \mathbf{E}_j(t) S(\mathbf{x} - \mathbf{x}_j), \quad \mathbf{B}_h(\mathbf{x}, t) = \sum_j \mathbf{B}_j(t) S(\mathbf{x} - \mathbf{x}_j), \quad (4.8)$$

where $\mathbf{E}_j(t) = \mathbf{E}(\mathbf{x}_j, t)$ and $\mathbf{B}_j(t) = \mathbf{B}(\mathbf{x}_j, t)$. Note that in the classical case where $S = S^1$ this regularization is equivalent to a linear interpolation of the fields defined at the grid points to the positions of the particles, but for higher order splines this is not an interpolation anymore and the regularized field at the grid points is not equal to its original value \mathbf{E}_j anymore, but for example in the case of S^3 , to $\frac{1}{6}\mathbf{E}_{j-1} + \frac{2}{3}\mathbf{E}_j + \frac{1}{6}\mathbf{E}_{j+1}$.

4.1.4 Time loop.

Let us now summarize the main stages to go from time t_n to time t_{n+1} :

1. We compute the charge density ρ_h and current density \mathbf{J}_h on the grid using relations (4.6)-(4.7).
2. We update the electromagnetic field using a classical mesh based solver (finite differences, finite elements, spectral,).
3. We compute the fields at the particle positions using relations (4.8).
4. Particles are advanced using a numerical scheme for the characteristics for example Verlet (4.1)-(4.3).

4.1.5 Conservation properties at the semi-discrete level

Conservation of mass. The discrete mass is defined as $\int f_h(\mathbf{x}, \mathbf{v}, t) d\mathbf{x}d\mathbf{v} = \sum_k w_k$. This is obviously conserved if no particle gets in or out of the domain, as w_k is conserved for each particle when the particles move.

Conservation of momentum. The total momentum of the system of particles of mass m and charge q is defined as

$$\mathcal{P} = m \int \mathbf{v} f_h(\mathbf{x}, \mathbf{v}, t) d\mathbf{x}d\mathbf{v} = \sum_k m w_k \mathbf{v}_k(t).$$

So

$$\frac{d\mathcal{P}}{dt} = \sum_k m w_k \frac{d\mathbf{v}_k}{dt} = \sum_k w_k q \mathbf{E}_h(\mathbf{x}_k, t).$$

In the case \mathbf{E}_h is computed using a Finite Difference approximation, its value at the particle position should be computed using the same convolution kernel as is used for computing the charge and current densities from the particle positions. Then $\mathbf{E}_h(\mathbf{x}_k, t) = \sum_j \mathbf{E}_j(t) S(\mathbf{x}_k - \mathbf{x}_j)$ and so

$$\frac{d\mathcal{P}}{dt} = \sum_k w_k q \sum_j \mathbf{E}_j(t) S(\mathbf{x}_k - \mathbf{x}_j).$$

Then exchanging the sum on the grid points i and the sum on the particles k we get

$$\frac{d\mathcal{P}}{dt} = \sum_j \mathbf{E}_j(t) \sum_k w_k q S(\mathbf{x}_k - \mathbf{x}_j) = \sum_j \mathbf{E}_j(t) \rho_j(t),$$

so that the total momentum is conserved provided the field solver is such that $\sum_j \mathbf{E}_j(t) \rho_j(t) = 0$. This is in particular true for a Fourier spectral Poisson solver for which

$$\rho_j = \sum_{m=-N/2+1}^{N/2-1} \hat{\rho}_m e^{-\frac{2i\pi j m}{N}}, \quad E_j = \sum_{m=-N/2+1}^{N/2-1} \hat{E}_m e^{-\frac{2i\pi j m}{N}},$$

with $\hat{E}_m = i \frac{\hat{\rho}_m}{m}$ for $m \neq 0$, because of the Poisson equation. Moreover we have set all 0 and $-N/2$ modes to zero. Then using the discrete Parseval formula

$$\sum_{j=0}^{N-1} \rho_j E_j = \sum_{m=-N/2+1}^{N/2-1} \hat{\rho}_m \hat{E}_m$$

Then as $\hat{E}_{-m} = \hat{E}_m$ because E_j is real, we get

$$\sum_j \rho_j E_j = \sum_{m=-N/2+1}^{N/2-1} \hat{\rho}_m \hat{E}_m = i \sum_{m=-N/2+1}^{N/2-1} m \hat{E}_m^2 = 0.$$

This is also true for the standard second order Finite Difference Poisson solver provided the electric field is computed from the potential with a centred finite difference approximation. Indeed in this case

$$\begin{aligned} \sum_j E_j \rho_j &= - \frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x} \frac{2\phi_j - \phi_{j+1} - \phi_{j-1}}{\Delta x^2} \\ &= \frac{1}{2\Delta x^3} \sum_j (-(\phi_{j+1}^2 - \phi_{j-1}^2) + 2\phi_j \phi_{j+1} - 2\phi_j \phi_{j-1}) = 0, \end{aligned} \quad (4.9)$$

using the periodicity of the grid and changing the indices in the last term.

The Finite Element PIC solver we introduced above does not conserve total momentum.

Remark 9 *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 motions for the particles

$$\frac{d\mathbf{x}_k}{dt} = \mathbf{v}_k, \quad \frac{d\mathbf{v}_k}{dt} = -\frac{q_k}{m_k} \nabla \phi_h(t, \mathbf{x}_k),$$

coupled with a finite element discretisation of the Poisson equation

$$\int \nabla \phi_h \cdot \nabla \psi \, d\mathbf{x} = \sum_k q_k w_k \psi(\mathbf{x}_k) \quad \forall \psi \in V_h.$$

Then the following semi-discrete energy is exactly conserved

$$\mathcal{E}_h(t) = \sum_k \frac{w_k m_k}{2} |\mathbf{v}_k|^2 + \frac{1}{2} \int |\nabla \phi_h|^2 \, d\mathbf{x}.$$

Let us verify this by direct computation. First taking $\psi = \phi_h$ as a test function, the weak Poisson equation yields

$$\frac{d}{dt} \int |\nabla \phi_h|^2 \, d\mathbf{x} = \sum_k q_k w_k \left(\frac{\partial \phi_h}{\partial t}(t, \mathbf{x}_k) + \frac{d\mathbf{x}_k}{dt} \cdot \nabla \phi_h(t, \mathbf{x}_k) \right). \quad (4.10)$$

On the other hand taking $\psi = \partial_t \phi_h(t, \mathbf{x}_k)$ in the weak Poisson equation, we also have that

$$\sum_k q_k w_k \frac{\partial \phi_h}{\partial t}(t, \mathbf{x}_k) = \int \nabla \phi_h \cdot \nabla \frac{\partial \phi_h}{\partial t} d\mathbf{x} = \frac{1}{2} \frac{d}{dt} \int |\nabla \phi_h|^2 d\mathbf{x},$$

so that equation (4.10) becomes

$$\frac{1}{2} \frac{d}{dt} \int |\nabla \phi_h|^2 d\mathbf{x} = \sum_k q_k w_k \frac{d\mathbf{x}_k}{dt} \cdot \nabla \phi_h(t, \mathbf{x}_k) = \sum_k q_k w_k \mathbf{v}_k \cdot \nabla \phi_h(t, \mathbf{x}_k).$$

Now using this, we find

$$\frac{d\mathcal{E}_h(t)}{dt} = \sum_k \left(w_k m_k \mathbf{v}_k \cdot \frac{d\mathbf{v}_k}{dt} + q_k w_k \mathbf{v}_k \cdot \nabla \phi_h(t, \mathbf{x}_k) \right) = 0$$

as

$$\frac{d\mathbf{v}_k}{dt} = -\frac{q_k}{m_k} \nabla \phi_h(t, \mathbf{x}_k).$$

4.1.6 Error estimation in Particle-In-Cell methods

Monte Carlo interpretation.

4.2 The semi-Lagrangian method

Semi-Lagrangian methods have become, far behind the Particle-In-Cell (PIC) method a classical choice for the numerical solution of the Vlasov equation, thanks to their good precision and their lack of numerical noise as opposite to PIC methods. They need a phase space mesh and thus are very computationally intensive when going to higher dimensions. Indeed a 3D simulation requires a 6D mesh of phase space. For this reason, semi-Lagrangian methods have become very popular for 1D or 2D problems, but there are still relatively few 3D simulations being performed with this kind of method.

The specificity of semi-Lagrangian methods, compared to classical methods for numerically solving PDEs on a mesh, is that they use the characteristics of the scalar hyperbolic equation, along with an interpolation method, to update the unknown from one time step to the next. These semi-Lagrangian methods exist in different varieties: backward, forward, point based or cell based. The most classical semi-Lagrangian method though is a backward point based method. This is the one we are going to consider in this lecture.

Let us consider an abstract scalar advection equation of the form

$$\frac{\partial f}{\partial t} + \mathbf{a}(\mathbf{x}, t) \cdot \nabla f = 0. \quad (4.11)$$

The characteristic curves associated to this equation are the solutions of the ordinary differential equations

$$\frac{d\mathbf{X}}{dt} = \mathbf{a}(\mathbf{X}(t), t).$$

We shall denote by $\mathbf{X}(t, \mathbf{x}, s)$ the unique solution of this equation associated to the initial condition $\mathbf{X}(s) = \mathbf{x}$.

The classical semi-Lagrangian method is based on a backtracking of characteristics. Two steps are needed to update the distribution function f^{n+1} at t_{n+1} from its value f^n at time t_n :

1. For each grid point \mathbf{x}_i compute $\mathbf{X}(t_n; \mathbf{x}_i, t_{n+1})$ the value of the characteristic at t_n which takes the value \mathbf{x}_i at t_{n+1} .
2. As the distribution solution of equation (4.11) verifies

$$f^{n+1}(\mathbf{x}_i) = f^n(\mathbf{X}(t_n; \mathbf{x}_i, t_{n+1})),$$

we obtain the desired value of $f^{n+1}(x_i)$ by computing $f^n(\mathbf{X}(t_n; \mathbf{x}_i, t_{n+1}))$ by interpolation as $\mathbf{X}(t_n; \mathbf{x}_i, t_{n+1})$ is in general not a grid point.

These operations are represented on Figure 4.1.

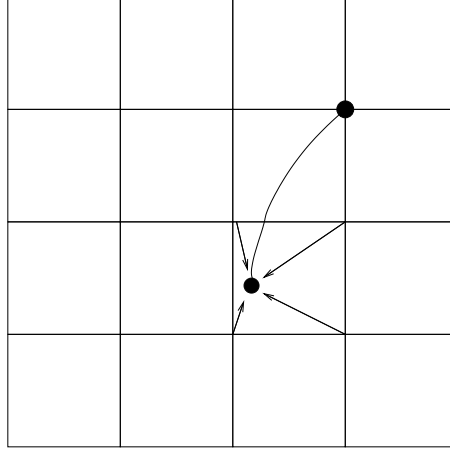


Figure 4.1: Sketch of the classical semi-Lagrangian method.

Remark 10 *This semi-Lagrangian method is very diffusive if a low order (typically linear) interpolation is used. In practice one often used cubic splines or cubic Hermite interpolation, which offer a good compromise between accuracy and efficiency.*

Let us now specify the algorithm for the 1D Vlasov-Poisson problem where the unknown is the distribution function for the electrons and in presence of motionless neutralizing background ions on a domain $[0, L]$ periodic in x and infinite in v . The equations then read

$$\begin{aligned} \frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - E(x, t) \frac{\partial f}{\partial v} &= 0, \\ \frac{dE}{dx} = \rho(x, t) &= 1 - \int f(x, v, t) dv, \end{aligned}$$

with the initial condition $f(x, v, 0) = f_0(x, v)$, verifying $\int f_0(x, v) dx dv = L$.

The infinite velocity space is truncated to a segment $[-A, A]$ sufficiently large so that f stays of the order of the round off errors for velocities less than $-A$ or larger than A during the whole simulation. In practice in the normalized examples we are going to consider, taking A of the order of 10 is very safe for all our test cases. Let us define a uniform grid of phase space $x_i = iL/N$, $i = 0, \dots, N-1$ (the point x_N which corresponds to x_0 is not used), $v_j = -A + j2A/M$, $j = 0, \dots, M$.

The full algorithm can in this case be written:

1. **Initialization.** Assume the initial distribution function $f_0(\mathbf{x}, \mathbf{v})$ given. We deduce $\rho(x, 0) = 1 - \int f_0(x, v) dv$, and then compute the initial electric field $E(x, 0)$ solving the Poisson equation.
2. **Update from t_n to t_{n+1} .** The function f^n is known at all grid points (x_i, v_j) of phase space and E^n is known at all grid points x_i of the configuration space.
 - We compute f^* by solving

$$\frac{\partial f}{\partial t} + E^n \frac{\partial f}{\partial v} = 0$$

on a half time step $\frac{\Delta t}{2}$ using the semi-Lagrangian method.

- We compute f^{**} by solving on a full time step

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} = 0$$

from the initial condition f^* .

- We compute $\rho^{n+1}(x) = 1 - \int f^{**}(x, v) dv$ and then the corresponding electric field E^{n+1} using the Poisson equation.
- We compute f^{n+1} by solving on a half time step

$$\frac{\partial f}{\partial t} + E^{n+1} \frac{\partial f}{\partial v} = 0 \tag{4.12}$$

from the initial condition f^{**} .

Note that the actual ρ^{n+1} can be computed using $f^{**}(x, v)$ (instead of $f^{n+1}(x, v)$), as the charge density corresponding to $f^{**}(x, v)$ is identical to that associated to $f^{n+1}(x, v)$. Indeed, we go from $f^{**}(x, v)$ to $f^{n+1}(x, v)$ by solving (4.12), and we notice, integrating this equation in v that it implies that $\frac{d}{dt} \int f(x, v, t) dv = 0$ and so that ρ is not modified during this stage.

Chapter 5

Fluid models

5.1 Derivation of the fluid model from the kinetic model

Due to collisions, the particles relax in long time to a Maxwellian, which is a thermodynamical equilibrium. When this state is approximately attained particles can be described by a fluid like model, where each particle species is modelled as a charged fluid.

This fluid model for each particle species can be derived from the corresponding Vlasov equation. The fluid models then replace the Vlasov equations and are still coupled to Maxwell's equation, or some reduced model, for the determination of the self-consistent electromagnetic field.

We start from the Vlasov-Boltzmann equation:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f}{\partial \mathbf{v}} = \mathcal{Q}(f, f). \quad (5.1)$$

Remark 11 *The Boltzmann collision operator $\mathcal{Q}(f, f)$ on the right hand side is necessary to provide the relaxation to thermodynamic equilibrium. However it will have no direct influence on our derivation, as we will consider only the first three velocity moments which vanish for the Boltzmann operator.*

The macroscopic quantities on which the fluid equations will be established are defined using the first three velocity moments of the distribution function $f(\mathbf{x}, \mathbf{v}, t)$

- The particle density is defined by

$$n(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v},$$

- The mean velocity $\mathbf{u}(\mathbf{x}, t)$ verifies

$$n(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t)\mathbf{v} d\mathbf{v},$$

- The pressure tensor $\mathbb{P}(\mathbf{x}, t)$ is defined by

$$\mathbb{P}(\mathbf{x}, t) = m \int f(\mathbf{x}, \mathbf{v}, t)(\mathbf{v} - \mathbf{u}(\mathbf{x}, t)) \otimes (\mathbf{v} - \mathbf{u}(\mathbf{x}, t)) d\mathbf{v}.$$

- The scalar pressure is one third of the trace of the pressure tensor

$$p(\mathbf{x}, t) = \frac{m}{3} \int f(\mathbf{x}, \mathbf{v}, t) |\mathbf{v} - \mathbf{u}(\mathbf{x}, t)|^2 d\mathbf{v},$$

- The temperature $T(\mathbf{x}, t)$ is related to the pressure and the density by

$$T(\mathbf{x}, t) = \frac{p(\mathbf{x}, t)}{n(\mathbf{x}, t)}.$$

- The energy flux is a vector defined by

$$\mathbf{Q}(\mathbf{x}, t) = \frac{m}{2} \int f(\mathbf{x}, \mathbf{v}, t) |\mathbf{v}|^2 \mathbf{v}(\mathbf{x}, t) d\mathbf{v}.$$

where we denote by $|\mathbf{v}| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$ and for two vectors $\mathbf{a} = (a_1, a_2, a_3)^\top$ and $\mathbf{b} = (b_1, b_2, b_3)^\top$, their tensor product $\mathbf{a} \otimes \mathbf{b}$ is the 3×3 matrix whose components are $(a_i b_j)_{1 \leq i, j \leq 3}$.

We obtain equations relating these macroscopic quantities by taking the first velocity moments of the Vlasov equation. In the actual computations we shall make use that f vanishes at infinity and that the plasma is periodic in space. This takes care of all boundary condition problems.

Let us first notice that as \mathbf{v} is a variable independent of \mathbf{x} , we have $\mathbf{v} \cdot \nabla_x f = \nabla_x \cdot (f \mathbf{v})$. Moreover, as $\mathbf{E}(\mathbf{x}, t)$ does not depend on \mathbf{v} and that the i^{th} component of

$$\mathbf{v} \times \mathbf{B}(\mathbf{x}, t) = \begin{pmatrix} v_2 B_3(\mathbf{x}, t) - v_3 B_2(\mathbf{x}, t) \\ v_3 B_1(\mathbf{x}, t) - v_1 B_3(\mathbf{x}, t) \\ v_1 B_2(\mathbf{x}, t) - v_2 B_1(\mathbf{x}, t) \end{pmatrix}$$

is independent of v_i , we also have

$$(\mathbf{E}(\mathbf{x}, t) + \mathbf{v} \times \mathbf{B}(\mathbf{x}, t)) \cdot \nabla_v f = \nabla_v \cdot (f(\mathbf{E}(\mathbf{x}, t) + \mathbf{v} \times \mathbf{B}(\mathbf{x}, t))).$$

Integrating the Vlasov equation (5.1) with respect to velocity \mathbf{v} we obtain

$$\frac{\partial}{\partial t} \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} + \nabla_x \cdot \int f(\mathbf{x}, \mathbf{v}, t) \mathbf{v} d\mathbf{v} + 0 = 0.$$

Whence, as $n(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t) \mathbf{v} d\mathbf{v}$, we get

$$\frac{\partial n}{\partial t} + \nabla_x \cdot (n \mathbf{u}) = 0. \quad (5.2)$$

Multiplying the Vlasov by $m \mathbf{v}$ and integrating with respect to \mathbf{v} , we get

$$\begin{aligned} m \frac{\partial}{\partial t} \int f(\mathbf{x}, \mathbf{v}, t) \mathbf{v} d\mathbf{v} + m \nabla_x \cdot \int (\mathbf{v} \otimes \mathbf{v}) f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \\ - q(\mathbf{E}(\mathbf{x}, t) \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} + \int f(\mathbf{x}, \mathbf{v}, t) \mathbf{v} d\mathbf{v} \times \mathbf{B}(\mathbf{x}, t)) = 0. \end{aligned}$$

Moreover,

$$\int \mathbf{v} \otimes \mathbf{v} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} = \int (\mathbf{v} - \mathbf{u}) \otimes (\mathbf{v} - \mathbf{u}) f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} + n \mathbf{u} \otimes \mathbf{u}.$$

Whence

$$m \frac{\partial}{\partial t} (n \mathbf{u}) + m \nabla \cdot (n \mathbf{u} \otimes \mathbf{u}) + \nabla \cdot \mathbb{P} = qn(\mathbf{E} + \mathbf{u} \times \mathbf{B}). \quad (5.3)$$

Finally multiplying the Vlasov equation by $\frac{1}{2}m|\mathbf{v}|^2 = \frac{1}{2}m\mathbf{v} \cdot \mathbf{v}$ and integrating with respect to \mathbf{v} , we obtain

$$\begin{aligned} \frac{1}{2}m \frac{\partial}{\partial t} \int f(\mathbf{x}, \mathbf{v}, t) |\mathbf{v}|^2 d\mathbf{v} + \frac{1}{2}m \nabla_x \cdot \int (|\mathbf{v}|^2 \mathbf{v}) f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \\ + \frac{1}{2}q \int |\mathbf{v}|^2 \nabla_v \cdot [(\mathbf{E}(\mathbf{x}, t) + \mathbf{v} \times \mathbf{B}(\mathbf{x}, t)) f(\mathbf{x}, \mathbf{v}, t)] d\mathbf{v} = 0. \end{aligned}$$

An integration by parts then yields

$$\begin{aligned} \int |\mathbf{v}|^2 \nabla_v \cdot (\mathbf{E}(\mathbf{x}, t) + \mathbf{v} \times \mathbf{B}(\mathbf{x}, t)) f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \\ = -2 \int \mathbf{v} \cdot [(\mathbf{E}(\mathbf{x}, t) + \mathbf{v} \times \mathbf{B}(\mathbf{x}, t)) f(\mathbf{x}, \mathbf{v}, t)] d\mathbf{v}. \end{aligned}$$

Then, developing $\int f|\mathbf{v} - \mathbf{u}|^2 d\mathbf{v}$ we get

$$\int f|\mathbf{v} - \mathbf{u}|^2 d\mathbf{v} = \int f|\mathbf{v}|^2 d\mathbf{v} - 2\mathbf{u} \cdot \int \mathbf{v} f d\mathbf{v} + |\mathbf{u}|^2 \int f d\mathbf{v} = \int f|\mathbf{v}|^2 d\mathbf{v} - n|\mathbf{u}|^2,$$

whence

$$\frac{\partial}{\partial t} \left(\frac{3}{2}p + \frac{1}{2}mn|\mathbf{u}|^2 \right) + \nabla \cdot \mathbf{Q} = \mathbf{E} \cdot (qn\mathbf{u}). \quad (5.4)$$

We could continue to calculate moments of f , but we see that each new expression reveals a moment of higher order. So we need additional information to have as many unknowns as equations to solve these equations. This additional information is called a *closure relation*.

In our case, we will use as a closure relation the physical property that at thermodynamic equilibrium the distribution function approaches a Maxwellian distribution function that we will note $f_M(\mathbf{x}, \mathbf{v}, t)$ and that can be expressed as a function of the macroscopic quantities $n(\mathbf{x}, t)$, $\mathbf{u}(\mathbf{x}, t)$ and $T(\mathbf{x}, t)$ which are the density, mean velocity and temperature of the charged fluid:

$$f_M(\mathbf{x}, \mathbf{v}, t) = \frac{n(\mathbf{x}, t)}{(2\pi T(\mathbf{x}, t)/m)^{3/2}} e^{-\frac{|\mathbf{v} - \mathbf{u}(\mathbf{x}, t)|^2}{2T(\mathbf{x}, t)/m}}.$$

We also introduce a classical quantity in plasma physics which is the thermal velocity of the particle species considered

$$v_{th} = \sqrt{\frac{T}{m}}.$$

It is easy to verify that the first three moments of the distribution function f_M are consistent with the definition of the macroscopic quantities n , \mathbf{u} and T defined for an arbitrary distribution function. We have indeed performing each time the change of variable $\mathbf{w} = \frac{\mathbf{v} - \mathbf{u}}{v_{th}}$

$$\begin{aligned} \int f_M(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} &= n(\mathbf{x}, t), \\ \int f_M(\mathbf{x}, \mathbf{v}, t) \mathbf{v} d\mathbf{v} &= n(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t), \\ \int f_M(\mathbf{x}, \mathbf{v}, t) |\mathbf{v} - \mathbf{u}|^2 d\mathbf{v} &= 3n(\mathbf{x}, t) T(\mathbf{x}, t)/m. \end{aligned}$$

On the other hand, replacing f by f_M in the definitions of the pressure tensor \mathbb{P} and the energy flux \mathbf{Q} , we can express these terms also in function of n , \mathbf{u} and T which enables us to obtain a closed system in these three unknowns as opposed to the case of an arbitrary distribution function f . Indeed, we first notice that, denoting by w_i the i^{th} component of \mathbf{w} ,

$$\int w_i w_j e^{-\frac{|\mathbf{w}|^2}{2}} d\mathbf{w} = \begin{cases} 0 & \text{if } i \neq j, \\ \int e^{-\frac{|\mathbf{w}|^2}{2}} d\mathbf{w} & \text{if } i = j. \end{cases}$$

It follows that the pressure tensor associated to the Maxwellian is

$$\mathbb{P} = m \frac{n}{(2\pi T/m)^{3/2}} \int e^{-\frac{|\mathbf{v}-\mathbf{u}|^2}{2T/m}} (\mathbf{v} - \mathbf{u}) \otimes (\mathbf{v} - \mathbf{u}) d\mathbf{v},$$

and so, thanks to our previous computation, the off diagonal terms of \mathbb{P} vanish, and by the change of variable $\mathbf{w} = \frac{\mathbf{v}-\mathbf{u}}{v_{th}}$, we get for the diagonal terms

$$\mathbb{P}_{ii} = m \frac{n}{(2\pi)^{3/2}} \frac{T}{m} \int e^{-\frac{\mathbf{w}^2}{2}} w_i^2 d\mathbf{w} = nT.$$

It follows that $\mathbb{P} = nT\mathbb{I} = p\mathbb{I}$ where \mathbb{I} is the 3×3 identity matrix. It now remains to compute in the same way \mathbf{Q} as a function of n , \mathbf{u} and T for the Maxwellian with the same change of variables, which yields

$$\begin{aligned} \mathbf{Q} &= \frac{m}{2} \frac{n}{(2\pi T/m)^{3/2}} \int e^{-\frac{|\mathbf{v}-\mathbf{u}|^2}{2T/m}} |\mathbf{v}|^2 \mathbf{v}(\mathbf{x}, t) d\mathbf{v}, \\ &= \frac{m}{2} \frac{n}{(2\pi)^{3/2}} \int e^{-\frac{\mathbf{w}^2}{2}} (v_{th} \mathbf{w} + \mathbf{u})^2 (v_{th} \mathbf{w} + \mathbf{u}) d\mathbf{w}, \\ &= \frac{m}{2} \frac{n}{(2\pi)^{3/2}} \int e^{-\frac{\mathbf{w}^2}{2}} (v_{th}^2 \mathbf{w}^2 \mathbf{u} + 2v_{th}^2 \mathbf{u} \cdot \mathbf{w} \mathbf{w} + |\mathbf{u}|^2 \mathbf{u}) d\mathbf{w}, \\ &= \frac{m}{2} n \left(3 \frac{T}{m} \mathbf{u} + 2 \frac{T}{m} \mathbf{u} + |\mathbf{u}|^2 \mathbf{u} \right), \end{aligned}$$

as the odd moments in \mathbf{w} vanish. We finally get

$$\mathbf{Q} = \frac{5}{2} nT \mathbf{u} + \frac{m}{2} n |\mathbf{u}|^2 \mathbf{u} = \frac{5}{2} p \mathbf{u} + \frac{m}{2} n |\mathbf{u}|^2 \mathbf{u}.$$

Then, plugging the expressions of \mathbb{P} and of \mathbf{Q} in (5.2)-(5.3)-(5.4) we obtain the fluid equations for one species of particles of a plasma:

$$\frac{\partial n}{\partial t} + \nabla_x \cdot (n\mathbf{u}) = 0 \quad (5.5)$$

$$m \frac{\partial}{\partial t} (n\mathbf{u}) + m \nabla \cdot (n\mathbf{u} \otimes \mathbf{u}) + \nabla p = qn(\mathbf{E} + \mathbf{u} \times \mathbf{B}) \quad (5.6)$$

$$\frac{\partial}{\partial t} \left(\frac{3}{2} p + \frac{1}{2} mn |\mathbf{u}|^2 \right) + \nabla \cdot \left(\frac{5}{2} p \mathbf{u} + \frac{m}{2} n |\mathbf{u}|^2 \mathbf{u} \right) = \mathbf{E} \cdot (qn\mathbf{u}), \quad (5.7)$$

which corresponds in three dimensions to a system of 5 scalar equation with 5 scalar unknowns which are the density n , the three components of the mean velocity \mathbf{u} and the scalar pressure p . These equations need of course to be coupled to Maxwell's equations for the computation of the self-consistent electromagnetic field with, in the case of only one particle species $\rho = qn$ and $\mathbf{J} = qn\mathbf{u}$. Let us also point out that an approximation often used in plasma physics is that of a cold plasma, for which $T = 0$ and thus $p = 0$. Only the first two equations are needed in this case.

5.2 An isothermal Euler-Poisson model

5.2.1 The model

For simplicity, we consider here only the 1D case. Our computational domain will then be the interval $[0, L]$, with periodic boundary conditions. Due to their much larger mass, on small time scales one can assume that the ions do not move. They are then characterised by their density n_0 . The electrons are characterised by their density $n(t, x)$ and average velocity $u(t, x)$. Their temperature T is assumed to be uniform and constant. Their mass is denoted by m and their charge by $-e$. Due to the global neutrality of the plasma, assuming singly charged ions,

$$\int_0^L n(t, x) dx = Ln_0.$$

Moreover, electron density and average velocity obey the conservation of mass equation (5.2)

$$\frac{\partial n}{\partial t} + \frac{\partial nu}{\partial x} = 0, \quad (5.8)$$

and the conservation of momentum equation (5.3) along with the isothermal pressure law $p = nT$, the temperature T being fixed, (note that in most plasma physics textbooks an adiabatic pressure law, which reads $\partial_x p = \gamma T \partial_x n$ with $\gamma = 3$ in 1D, is used)

$$\frac{\partial(nu)}{\partial t} + \frac{\partial(nu^2)}{\partial x} + \frac{T}{m} \frac{\partial n}{\partial x} + \frac{e}{m} nE = 0. \quad (5.9)$$

We assume in addition that the magnetic field can be neglected and that the electric field is given by Poisson's equation

$$-\frac{d^2\phi}{dx^2} = \frac{e}{\varepsilon_0}(n_0 - n), \quad E(t, x) = -\frac{d\phi}{dx}. \quad (5.10)$$

These completely characterise the unknowns, along with initial and boundary conditions, as the temperature is given. Else the conservation of energy equation would also be needed, but this does not add any new difficulty for the numerical simulation.

Let us now check that the model is conservative, in the sense that it conserves exactly the number of particles $\int_0^L n(t, x) dx$ and also $\int_0^L n(t, x)u(t, x) dx$ in time. Integrating (5.8) over one period $[0, L]$ immediately yields

$$\frac{d}{dt} \int_0^L n(t, x) dx = 0.$$

Then, integrating (5.9) over one period $[0, L]$ yields

$$\frac{d}{dt} \int_0^L n(t, x)u(t, x) dx + \frac{e}{m} \int_0^L nE dx = 0.$$

On the other hand multiplication of the Poisson equation by E and integrating yields

$$\int_0^L E \frac{dE}{dx} dx = \frac{1}{2} \int_0^L \frac{dE^2}{dx} dx = 0 = \frac{e}{\varepsilon_0} \int_0^L (n_0 - n)E dx = -\frac{e}{\varepsilon_0} \int_0^L nE dx$$

as $E = -\frac{d\phi}{dx}$. It follows that $\int_0^L nE dx = 0$ and then the conservation property

$$\frac{d}{dt} \int_0^L n(t, x)u(t, x) dx = 0.$$

5.2.2 Study of the linearised equations

In order to understand the linear behaviour of the model and also to get a verification test for our codes, we shall perform a linear analysis of this model. Let us for this linearise the model around a constant steady state, assuming

$$n(t, x) = n_0 + \epsilon n_1(t, x), \quad u(t, x) = u_0 + \epsilon u_1(t, x) \text{ with } u_0 = 0.$$

Performing the same expansion for the electrostatic potential and the electric field, it follows from Poisson's equation (5.10), identifying the terms of same order in ϵ that $E_0 = 0$ and

$$-\frac{d^2\phi_1}{dx^2} = -\frac{e}{\epsilon_0}n_1, \quad E_1(t, x) = -\frac{d\phi_1}{dx}.$$

Then, plugging these expressions into our equations (5.8)-(5.9), we get

$$\epsilon \frac{\partial n_1}{\partial t} + \epsilon n_0 \frac{\partial u_1}{\partial x} + \epsilon^2 \frac{\partial(n_1 u_1)}{\partial x} = 0,$$

$$\epsilon n_0 \frac{\partial u_1}{\partial t} + \epsilon^2 \frac{\partial(n_1 u_1)}{\partial x} + \epsilon^2 \frac{\partial(n_0 u_1^2)}{\partial x} + \epsilon^3 \frac{\partial(n_1 u_1^2)}{\partial x} + \epsilon \frac{T}{m} \frac{\partial n_1}{\partial x} + \epsilon \frac{e}{m} n_0 E_1 + \epsilon^2 \frac{e}{m} n_1 E_1 = 0.$$

Now, dividing these equations by ϵ and letting ϵ go to zero, we get the linearised equations

$$\frac{\partial n_1}{\partial t} + n_0 \frac{\partial u_1}{\partial x} = 0, \tag{5.11}$$

$$n_0 \frac{\partial u_1}{\partial t} + \frac{T}{m} \frac{\partial n_1}{\partial x} + \frac{e}{m} n_0 E_1 = 0, \tag{5.12}$$

that are coupled to the Poisson equation

$$-\frac{d^2\phi_1}{dx^2} = -\frac{e}{\epsilon_0}n_1, \quad E_1(t, x) = -\frac{d\phi_1}{dx}. \tag{5.13}$$

These linearised equations can be exactly solved by performing a Fourier expansion in space and a Laplace transform in time. Assuming a periodic domain in x of period 2π , the Fourier series of n_1 and u_1 read

$$n_1(t, x) = \sum_{k=-\infty}^{+\infty} \hat{n}_k(t) e^{ikx}, \quad u_1(t, x) = \sum_{k=-\infty}^{+\infty} \hat{u}_k(t) e^{ikx}.$$

Plugging this into (5.11) and (5.16) and identifying each of the modes this yields

$$\frac{d\hat{n}_k}{dt} + ikn_0\hat{u}_k = 0, \tag{5.14}$$

$$n_0 \frac{d\hat{u}_k}{dt} + ik \frac{T}{m} \hat{n}_k + \frac{e}{m} n_0 \hat{E}_k = 0, \tag{5.15}$$

Then from the Poisson equation (5.13) we find $\hat{E}_k = (e/\epsilon_0)\hat{n}_k/k$. Plugging this into (5.15) yields

$$n_0 \frac{d\hat{u}_k}{dt} + i \left(kv_{th}^2 + \frac{\omega_p^2}{k} \right) \hat{n}_k = 0, \tag{5.16}$$

introducing the electron plasma frequency $\omega_p = \sqrt{e^2 n_0 / (m \epsilon_0)}$ and the electron thermal velocity $v_{th} = \sqrt{T/m}$.

We define the Laplace transform of a function f for $\Im(\omega)$ large enough by

$$\tilde{f}(\omega) = \int_0^{+\infty} f(t) e^{i\omega t} dt.$$

It follows immediately from the definition that the Laplace transform of the time derivative is

$$\int_0^{+\infty} \frac{d\tilde{f}}{dt} = -f(0) - i\omega \tilde{f}.$$

Using this we find the Laplace transforms of (5.11)-(5.16)

$$-\hat{n}_k(0) - i\omega \tilde{n}_k + ikn_0 \tilde{u}_k = 0, \quad (5.17)$$

$$-n_0(\hat{u}_k(0) + i\omega \tilde{u}_k) + i \left(kv_{th}^2 + \frac{\omega_p^2}{k} \right) \tilde{n}_k = 0. \quad (5.18)$$

Multiplying (5.18) by ω and plugging in the expression of $i\omega \tilde{n}_k$ from (5.17) yields

$$-n_0(\omega \hat{u}_k(0) + i\omega^2 \tilde{u}_k) + \left(kv_{th}^2 + \frac{\omega_p^2}{k} \right) (ikn_0 \tilde{u}_k - \hat{n}_k(0)) = 0$$

So that we find the following expression for \tilde{u}_k and \tilde{n}_k

$$\tilde{u}_k = i \frac{\omega \hat{u}_k(0) + (kv_{th}^2 + \omega_p^2/k) \hat{n}_k(0)/n_0}{\omega^2 - (k^2 v_{th}^2 + \omega_p^2)} \quad (5.19)$$

$$\tilde{n}_k = i \frac{\omega \hat{n}_k(0) + kn_0 \hat{u}_k(0)}{\omega^2 - (k^2 v_{th}^2 + \omega_p^2)} \quad (5.20)$$

The expression of the corresponding Fourier modes, from their Laplace transform, is computed using the inverse Laplace transform defined by, for a given large enough number R ,

$$f(t) = \frac{1}{2\pi} \lim_{A \rightarrow +\infty} \int_{-A+iu}^{A+iu} \tilde{f}(s) e^{-i\omega t} d\omega, \quad \forall t > 0, u > R$$

This integral can be computed using the residue theorem, by closing the contour with a half circle towards the bottom of the complex plane for some given A large enough to enclose the poles. It can be shown that the closing piece tends to zero when $A \rightarrow \infty$ so that

$$f(t) = \frac{1}{i} \sum_{\omega_j} \text{Res}_{\omega_j} \tilde{f} e^{-i\omega_j t} \quad (5.21)$$

where $\text{Res}_{\omega_j} \tilde{f} = \lim_{\omega \rightarrow \omega_j} (f(\omega)(\omega - \omega_j))$. The poles ω_j are found by solving the so-called dispersion relation

$$\omega^2 = \omega_p^2 + v_{th}^2 k^2. \quad (5.22)$$

This corresponds to the plasma oscillations or *Langmuir waves*. The two solutions are $\omega_0(k)$ and $-\omega_0(k)$

$$\omega_0 = \sqrt{\omega_p^2 + v_{th}^2 k^2}.$$

Plugging these into (5.21) and computing the corresponding residues, we find

$$\hat{n}_k(t) = \hat{n}_k(0) \cos(\omega_0 t) - \hat{u}_k(0) \frac{ikn_0}{\omega_0} \sin(\omega_0 t), \quad (5.23)$$

$$\hat{u}_k(t) = \hat{u}_k(0) \cos(\omega_0 t) - \hat{n}_k(0) \frac{i\omega_0}{kn_0} \sin(\omega_0 t). \quad (5.24)$$

5.2.3 Hyperbolicity

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

$$\frac{\partial u}{\partial t} + \mathbf{a} \cdot \nabla u = 0,$$

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 have the abstract form

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

where \mathbf{u} is a vector of unknown values. This can be written also

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

where $A(\mathbf{u})$ is the Jacobian matrix with components $((\frac{\partial F_i}{\partial u_j}))_{i,j}$. The system is called *hyperbolic* if for all \mathbf{u} the matrix A has only real eigenvalues and is diagonalisable. It is called *strictly hyperbolic* if all eigenvalues are distinct.

Let us write our isothermal Euler-Poisson equations as a hyperbolic system, the coupling term between Euler and Poisson being handled as a source term. For this it is important to work with the conserved variables n and nu and not with the natural physical variables n and u . So our vector of unknowns \mathbf{u} and flux $\mathbf{F}(\mathbf{u})$ are

$$\mathbf{u} = \begin{pmatrix} n \\ nu \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad \mathbf{F}(\mathbf{u}) = \begin{pmatrix} nu \\ nu^2 + \frac{T}{m}n \end{pmatrix} = \begin{pmatrix} u_2 \\ u_2^2/u_1 + \frac{T}{m}u_1 \end{pmatrix}.$$

Then we can compute the jacobian matrix of the flux

$$A(\mathbf{u}) = \begin{pmatrix} 0 & 1 \\ -u_2^2/u_1^2 + \frac{T}{m} & 2u_2/u_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -u^2 + v_{th}^2 & 2u \end{pmatrix}.$$

In order to check that the corresponding system is hyperbolic we need to compute the eigenvalues of $A(\mathbf{u})$. They are the solutions λ of $\det(\lambda \mathbb{I} - A(\mathbf{u})) = 0$:

$$-\lambda(2u - \lambda) - (v_{th}^2 - u^2) = 0.$$

Grouping the terms of same degree yields the quadratic equation

$$\lambda^2 - 2u\lambda - (v_{th}^2 - u^2) = 0.$$

The discriminant is $\Delta = 12v_{th}^2 \geq 0$, so that the matrix has only real eigenvalues, which are

$$\lambda_1 = u + v_{th}, \quad \lambda_2 = u - v_{th}.$$

5.3 The Finite Volume method

5.3.1 The first order Finite Volume schemes

Let us introduce the Finite Volume method on the generic scalar conservation law of the form

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0. \quad (5.25)$$

In the case of our linear advection equation, we have $f(u) = au$.

In the Finite Volume method, the computational domain is divided into cells (intervals in 1D) and the unknown quantity that is numerically computed is the cell average of u on each cell. Recall that for Finite Differences the unknowns were the point values of u at the grid points. We need to number the cells. In 1D a convenient way to do it in order to avoid confusion with the grid points, is to assign half integers. Let us denote by

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

The Finite Volume numerical scheme is then obtained by integrating the original equation on each cell of the domain. As for the time scheme there are at least two classical ways, the first is to also integrate in time between t_n and t_{n+1} and then use a quadrature formula to compute the integral, for example, the left hand rectangle rule yields a first order explicit formula. The second is to use the method of lines and separate space discretisation from time discretisation. Then standard ODE discretisation schemes can be used. This is what we shall do mostly in this lecture.

So integrating (5.25) on the cell $[x_i, x_{i+1}]$ and dividing by $\Delta x_{i+\frac{1}{2}} = x_{i+1} - x_i$ yields

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

Here we see that a second ingredient is needed in order to define the algorithm. We only know the cell averages of u , how do we define the value at u at the cell interfaces. 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. But it is not defined at the cell interface. In order to complete the Finite Volume scheme we need to define a so called numerical flux at each cell interface denoted by g_i that needs to be consistent with $f(x_i)$, i.e. $g_i = f(u(x_i)) + O(\Delta x^p)$ for some positive p . 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 Δx :

$$\frac{du_{i+\frac{1}{2}}(t)}{dt} + \frac{(f(u_{i+\frac{3}{2}}) - f(u_{i-\frac{1}{2}}))}{2\Delta x} = 0.$$

Coupling it with an explicit Euler scheme in time this becomes, and applying it to the linear advection ($f(u) = au$) we get

$$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^n - \frac{a\Delta t}{2\Delta x}(u_{i+\frac{3}{2}}^n - u_{i-\frac{1}{2}}^n). \quad (5.26)$$

We recognise here the centred Finite Difference scheme shifted to the cell centres. Remember that 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 like for Finite Differences. 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 $a > 0$

$$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^n - \frac{a\Delta t}{\Delta x}(u_{i+\frac{1}{2}}^n - u_{i-\frac{1}{2}}^n). \quad (5.27)$$

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

Remark 12 *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.

5.3.2 Higher order schemes

In order to get high order Finite Volume schemes, the idea is to reconstruct polynomials of some given degree from the cell averages that are obtained with the Finite Volume procedure. 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 $u_{j+\frac{1}{2}}^n$ known 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 $\tilde{p}_m(x)$ such that $\frac{d}{dx}\tilde{p}_m(x) = p_m(x)$. Then

$$\Delta x_{j+\frac{1}{2}} u_{j+\frac{1}{2}}^n = \int_{x_j}^{x_{j+1}} p_m(x) dx = \tilde{p}_m(x_{j+1}) - \tilde{p}_m(x_j).$$

Let $W(x) = \int_{x_0}^x \tilde{u}^n(x) dx$ a primitive of the piecewise constant function \tilde{u}^n with value $u_{j+\frac{1}{2}}^n$ on $[x_j, x_{j+1}]$. Then $W(x_{j+1}) = \sum_{k=1}^j h_{k+\frac{1}{2}} u_{k+\frac{1}{2}}^n$ and

$$W(x_{j+1}) - W(x_j) = \Delta x_{j+\frac{1}{2}} u_{j+\frac{1}{2}}^n = \tilde{p}_m(x_{j+1}) - \tilde{p}_m(x_j).$$

Then we take for \tilde{p}_m an interpolating polynomial at points x_j of W so that

$$\begin{aligned} \frac{1}{\Delta x_{j+\frac{1}{2}}} \int_{x_j}^{x_{j+1}} p_m(x) dx &= \frac{1}{\Delta x_{j+\frac{1}{2}}} (\tilde{p}_m(x_{j+1}) - \tilde{p}_m(x_j)) \\ &= \frac{1}{\Delta x_{j+\frac{1}{2}}} (W(x_{j+1}) - W(x_j)) = u_{j+\frac{1}{2}}^n. \end{aligned}$$

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 polynomial. 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 [10] 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 [12].

5.4 Systems of conservation laws

5.4.1 Linear systems - The Riemann problem

Let us now consider linear systems of conservation laws in 1D. This can be written in the general form

$$\frac{\partial U}{\partial t} + A \frac{\partial U}{\partial x} = 0,$$

where $U(t, x)$ is a vector in \mathbb{R}^n and A a given matrix with constant coefficients. We will focus on the Finite Volume method.

The main numerical issue when constructing a Finite Volume scheme is to find a good numerical flux that is consistent (i.e. converges towards the exact flux when the cell size goes to 0) and stable. As we saw previously in the linear scalar case enhanced stability is given by upwinding. We now need to generalise the idea of upwinding to the case of systems.

The construction of a numerical flux is a local procedure at the interface between two cells, where a different value is given on the left side and on the right side from the

polynomial reconstruction. In order to get information from the equation itself the idea is to solve it locally using an initial condition which is a step function. The Riemann problem is the corresponding initial value problem:

$$\begin{aligned} \frac{\partial U}{\partial t} + A \frac{\partial U}{\partial x} &= 0, \\ U(0, x) &= \begin{cases} U_L & \text{if } x < 0, \\ U_R & \text{if } x \geq 0, \end{cases} \end{aligned}$$

where U_L and U_R are two given constant vectors.

The system being hyperbolic implies that A has real eigenvalues and can be diagonalised. Hence $A = P\Lambda P^{-1}$, where Λ is the diagonal matrix containing the eigenvalues. Then introducing the so-called characteristic variables $V = P^{-1}U$, and multiplying the system by P^{-1} on the left we get

$$P^{-1} \frac{\partial U}{\partial t} + P^{-1} A P P^{-1} \frac{\partial U}{\partial x} = \frac{\partial V}{\partial t} + \Lambda \frac{\partial V}{\partial x} = 0.$$

So in the variable V the system is diagonal and reduces to the set of linear advection equations

$$\frac{\partial v_i}{\partial t} + \lambda_i \frac{\partial v_i}{\partial x} = 0, \quad 1 \leq i \leq n$$

where the v_i are the components of V and the λ_i the eigenvalues of A . The exact solution of these equations is given by $v_i(t, x) = v_i(0, x - \lambda_i t)$, where the $v_i(0, x)$ are the components of the initial vector which take the constant values $V_L = P^{-1}U_L$ if $x < 0$ and $V_R = P^{-1}U_R$ if $x \geq 0$. In other terms

$$v_i(t, x) = \begin{cases} v_{i,L} & \text{if } x < \lambda_i t, \\ v_{i,R} & \text{if } x \geq \lambda_i t. \end{cases}$$

In practice we want to use the Riemann problem to determine the value of V (and U) at the cell interface, corresponding to $x = 0$, the discontinuity point at any strictly positive time. And we deduce from the previous solution that

$$v_i(t, 0) = \begin{cases} v_{i,L} & \text{if } 0 < \lambda_i, \\ v_{i,R} & \text{if } 0 \geq \lambda_i. \end{cases}$$

In order to get a vector expression, we introduce the diagonal matrices Λ_+ where the negative eigenvalues are replaced by 0 and Λ_- where the positive eigenvalues are replaced by 0. Obviously $\Lambda = \Lambda_+ + \Lambda_-$. Then for $t > 0$ we have

$$\Lambda V(t, 0) = \Lambda_+ V(t, 0) + \Lambda_- V(t, 0) = \Lambda_+ V_L + \Lambda_- V_R,$$

as for all positive eigenvalues the corresponding component of $V(t, 0)$ is $v_{i,L}$ and for all negative eigenvalues the corresponding component of $V(t, 0)$ is $v_{i,R}$. Note that as $V(t, 0)$ is multiplied by Λ the components of $V(t, 0)$ corresponding to 0 eigenvalues do not need to be considered as they are multiplied by 0 anyway. So the side where the strict inequality is used for the initial condition of the Riemann problem plays no role.

Denoting by $A_+ = P\Lambda_+P^{-1}$ and $A_- = P\Lambda_-P^{-1}$ the flux $AU(t, 0)$ associated to the solution of the Riemann problem at the cell interface can also be expressed conveniently directly in terms of U

$$AU(t, 0) = P\Lambda_+V(t, 0) + P\Lambda_-V(t, 0) = P\Lambda_+V_L + P\Lambda_-V_R = A_+U_L + A_-U_R.$$

This expression $AU(t, 0) = A_+U_L + A_-U_R$ can be used to define the numerical flux at the cell interface, using the value U_L coming from the left-hand side of the interface and U_R coming from the right-hand side of the interface. For actual computations, the matrices A_+ and A_- need to be computed explicitly from the eigenvalues and eigenvectors of the matrix A . Notice that in the case of a scalar equation the matrix A is reduced to the scalar a which is then obviously the only eigenvalue of the 1×1 matrix and if $a > 0$ we have $A_+ = a$ and $A_- = 0$, so that the numerical flux becomes $au(t, 0) = au_L$ and the same way if $a < 0$ $au(t, 0) = au_R$, so that the numerical flux obtained from the solution of the Riemann problem reduces to the upwind flux.

Example. We consider the 1D Maxwell equations which can be written in dimensionless units:

$$\begin{aligned}\frac{\partial E}{\partial t} + \frac{\partial B}{\partial x} &= 0, \\ \frac{\partial B}{\partial t} + \frac{\partial E}{\partial x} &= 0.\end{aligned}$$

This can be written in the form of a linear system

$$\frac{\partial U}{\partial t} + A \frac{\partial U}{\partial x} = 0, \quad \text{with } U = \begin{pmatrix} E \\ B \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The eigenvalues of A are the solutions of $\det(A - \lambda I) = 0$, *i.e.* $\lambda^2 = 1$. So the eigenvalues are $\lambda_1 = -1$ and $\lambda_2 = 1$. They are real and distinct so that the system is strictly hyperbolic. Let V_i be a normalised eigenvector associated to the eigenvalue λ_i , $i = 1, 2$. We have $AV_1 = -V_1$ so that $V_1 = \frac{1}{\sqrt{2}}(1, -1)^T$ and $AV_2 = V_2$ so that $V_2 = \frac{1}{\sqrt{2}}(1, 1)^T$. We define P the matrix whose columns are V_1 and V_2 . P is obviously orthonormal, so that its inverse is its transpose. Then we have $PA = \Lambda P$. So that we can define:

$$\begin{aligned}A_+ &= P\Lambda_+P^T = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \\ A_- &= P\Lambda_-P^T = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}\end{aligned}$$

Hence, the upwind flux is given by

$$AU(t, 0) = A_+U_L + A_-U_R = \frac{1}{2} \begin{pmatrix} U_{L,1} + U_{L,2} + (-U_{R,1} + U_{R,2}) \\ U_{L,1} + U_{L,2} + (U_{R,1} - U_{R,2}) \end{pmatrix}.$$

Remark 13 *As for the scalar problem, the numerical flux can be taken as a linear combination of the centred flux and the upwind flux (solution the Riemann problem):*

$$G_j = \mu \frac{1}{2} A(U_L + U_R) + (1 - \mu)(A_+U_L + A_-U_R), \quad 0 \leq \mu \leq 1.$$

5.5 Nonlinear systems of conservation laws

The specificity of non linear conservations laws as opposed to linear conservation laws is that discontinuities, called shocks, can appear during the evolution even when starting from smooth solutions. Then derivatives are not longer well defined and the concept

of weak solutions, as for finite elements, putting the derivative on a test function must be defined. The major problem of weak solutions is that they are not unique. However the concept of vanishing viscosity, considering a conservation law as the limit, when the viscosity term tends to zero of the same equation with an added diffusion term. For a 1D scalar conservation law this has the form

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} - \epsilon \frac{\partial^2 u}{\partial x^2} = 0,$$

where ϵ is the small viscosity parameter. This PDE has a unique solution for all $\epsilon > 0$ and the unique physical solution of the conservation law is the limit of this equation when ϵ goes to zero. This solution is called the vanishing viscosity solution or entropy solution. Non conservative schemes can converge to a wrong weak solution and even some conservative schemes which do not have enough numerical viscosity can converge to a non entropic solution. In order to avoid this, one should always use conservative schemes that have locally enough viscosity to make sure that the solution converges towards the right entropy solution.

Going from the scalar case to systems in the non linear case, is similar to what is done in the linear case. The hyperbolicity of the system is essential so that the system can be locally diagonalised and the eigenvalues explicitly used in the definition of the flux.

The derivation of a Finite Volume scheme can be done component by component and so reduces to the scalar case except for the definition of the numerical flux which in general mixes the different components and needs to be specific to the system at hand. We shall restrict in this lecture to the introduction of two of the most used numerical fluxes, namely the Rusanov (or local Lax-Friedrichs) flux and the Roe flux.

5.5.1 The Rusanov flux

The main idea here is to use a centred flux to which just enough dissipation is added to ensure stability in all cases. In the scalar case our conservative scheme reads

$$\frac{du_{i+\frac{1}{2}}}{dt} + \frac{1}{\Delta x}(g_{i+1} - g_i) = 0$$

so we see that adding a term of the form $g_i = u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}$ yields a contribution to the flux of the form

$$\Delta x \frac{u_{i+\frac{3}{2}} - 2u_{i+\frac{1}{2}} + u_{i-\frac{1}{2}}}{\Delta x^2}.$$

We see that this models a diffusion term which vanishes in the limit $\Delta x \rightarrow 0$. This has a stabilising effect but also adds diffusion. For linear advection a stability analysis yields that the scheme becomes stable when a coefficient in front is added which is the wave speed. A system of n components corresponds to the superposition of n waves the local speed of each being given by the corresponding eigenvalue. So taking the viscosity coefficient in the flux as the maximum over all eigenvalues should do the job. This yields the Rusanov flux for systems, which is the simplest stable flux. It is defined for a nonlinear system of the form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = 0,$$

as

$$\mathbf{G}(\mathbf{U}_L, \mathbf{U}_R) = \frac{1}{2} \left(\mathbf{F}(\mathbf{U}_L) + \mathbf{F}(\mathbf{U}_R) - \max_{U \in [U_L, U_R]} |\lambda(\mathbf{F}'(\mathbf{U}))| (\mathbf{U}_R - \mathbf{U}_L) \right),$$

where $\max_{U \in [U_L, U_R]} |\lambda(\mathbf{F}'(\mathbf{U}))|$ denotes the maximum modulus of the eigenvalues of the Jacobian matrix $\mathbf{F}'(\mathbf{U})$.

5.5.2 The Roe flux

Roe's method consists in locally linearising the non linear flux with a well chosen procedure. The linearised matrix between two constant states \mathbf{U}_L and \mathbf{U}_R is denoted by $A(\mathbf{U}_L, \mathbf{U}_R)$ and constructed such that the following properties are verified:

- $\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L) = A(\mathbf{U}_L, \mathbf{U}_R)(\mathbf{U}_R - \mathbf{U}_L)$.
- $A(\mathbf{U}, \mathbf{U}) = \mathbf{F}'(\mathbf{U})$.
- $A(\mathbf{U}_L, \mathbf{U}_R)$ is diagonalisable, has real eigenvalues and a complete system of eigenvectors.

Such a matrix is not always easy to find, but there are procedures, described in [10] for example, to construct them. Moreover classical Roe matrices are known for the most usual systems [10].

Once the Roe matrix is defined, the flux can be computed by solving the corresponding linear Riemann problem that we treated previously. Let us rewrite the formula, so that we can also include the entropy fix, which is needed for non linear systems to make sure that the scheme always converges to the correct entropy solution.

In the case of linear systems, the flux was defined as $AU(t, 0) = A_+U_L + A_-U_R$. Defining the absolute value of a matrix as $|A| = A_+ - A_-$, the flux can also be expressed as

$$AU(t, 0) = \frac{1}{2} (AU_L + AU_R - |A|(U_R - U_L)).$$

Using the properties of the Roe matrix in the non linear case, the same expression will be used to define the Roe flux:

$$\mathbf{G}(\mathbf{U}_L, \mathbf{U}_R) = \frac{1}{2} (A(\mathbf{U}_L, \mathbf{U}_R)\mathbf{U}_L + A(\mathbf{U}_L, \mathbf{U}_R)\mathbf{U}_R - |A(\mathbf{U}_L, \mathbf{U}_R)|(\mathbf{U}_R - \mathbf{U}_L)).$$

Here we also see that the numerical viscosity vanishes when the eigenvalues of $A(u_L, u_R)$ are close to zero, which can happen close to the minimum of the convex function f . Then a non entropic shock might be selected by the scheme. A simple fix, introduced by Harten, consists in smoothing the graph of the absolute value close to 0 (see [10] for details). This consists in replacing the absolute value in the formula defining the flux by

$$\phi(\lambda) = \begin{cases} |\lambda| & |\lambda| \geq \epsilon, \\ (\lambda^2 + \epsilon^2)/(2\epsilon) & |\lambda| < \epsilon. \end{cases}$$

This ensures that $\phi(\lambda) \geq \epsilon$ and that there is always some dissipation. This works and yields the correct entropy solution provided ϵ is well tuned to the problem at hand.

Bibliography

- [1] Jean-Paul Berrut and Lloyd N Trefethen. Barycentric lagrange interpolation. *Siam Review*, 46(3):501–517, 2004.
- [2] Prabhu Lal Bhatnagar, Eugene P Gross, and Max Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical review*, 94(3):511, 1954.
- [3] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.
- [4] Gary Cohen. *Higher-Order Numerical Methods for Transient Wave equation*. Springer-Verlag, 2001.
- [5] Victor Eijkhout. Introduction to high-performance scientific computing. <http://pages.tacc.utexas.edu/~eijkhout/istc/istc.html>.
- [6] Martin J Gander and Gerhard Wanner. From euler, ritz, and galerkin to modern computing. *SIAM Review*, 54(4):627–666, 2012.
- [7] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Number 16 in Frontiers in Applied Mathematics. SIAM, Philadelphia, 1995.
- [8] LD Landau. The transport equation in the case of coulomb interactions. *Collected Papers of LD Landau, Pergamon press, Oxford*, pages 163–170, 1981.
- [9] Andrew Lenard and Ira B Bernstein. Plasma oscillations with diffusion in velocity space. *Physical Review*, 112(5):1456, 1958.
- [10] Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.
- [11] Marshall N Rosenbluth, William M MacDonald, and David L Judd. Fokker-planck equation for an inverse-square force. *Physical Review*, 107(1):1, 1957.
- [12] Chi-Wang Shu. High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM review*, 51(1):82–126, 2009.
- [13] Willi-Hans Steeb. *Kronecker product of matrices and applications*. BI-Wissenschaftsvlg, 1991.
- [14] Willi-Hans Steeb. *Matrix calculus and Kronecker product with applications and C++ programs*. World Scientific, 1997.

- [15] Vidar Thomée. From finite differences to finite elements: A short history of numerical analysis of partial differential equations. *Journal of Computational and Applied Mathematics*, 128(1):1–54, 2001.
- [16] Lloyd N Trefethen. *Spectral methods in MATLAB*, volume 10. Siam, 2000.
- [17] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.
- [18] Charles F. van Loan. The ubiquitous kronecker product. *J. Comput. Appl. Math.*, 123:85–100, November 2000.
- [19] H. Yoshida. Construction of higher order symplectic integrators. *Phys. Lett. A*, 150:262, 1990.