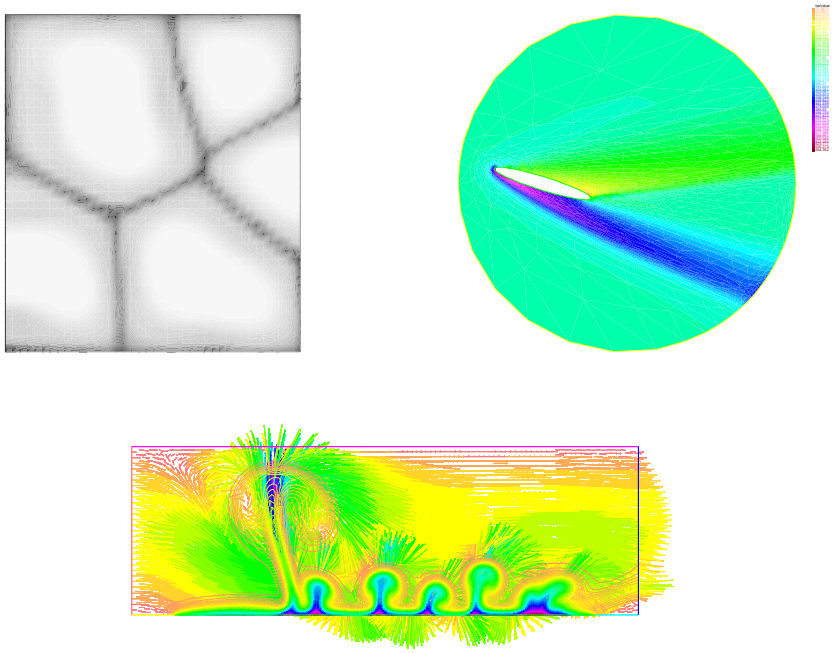
Numerical modeling of transport problems using freefem++ software – with examples in biology, CFD, traffic flow and energy transfer

Florian de Vuyst

Numerical modeling of transport problems using freefem++ software

with examples in biology, CFD, traffic flow and energy transfer



Florian De Vuyst

Department of Mathematics – ENS CACHAN

Contents

1 Transport equations **7**

1.1 Pure transport equations . . . 7

1.1.1 Non homogeneous case . . . 8

1.1.2 Stationary case . . . 8

1.2 Conservative case . . . 9

1.3 Connections between transport and conservation equations. . . . 11

2 Numerical analysis of Eulerian difference schemes **13**

2.1 Framework . . . 13

2.2 Finite difference schemes . . . 14

2.3 Numerical stability . . . 15

2.3.1 ℓ_∞ stability . . . 15

2.3.2 ℓ_1 stability . . . 16

2.3.3 ℓ_2 stability, von Neumann stability . . . 17

2.4 Consistency properties . . . 18

2.4.1 Lax-Wendroff scheme . . . 18

2.4.2 Von-Neumann stability of the Lax-Wendroff scheme . . . 20

2.5 Lax-Friedrichs scheme . . . 20

2.5.1 Von Neumann stability analysis of the Lax-Friedrichs scheme . . . 21

2.6 Hybrid interpolated fluxes . . . 22

2.6.1 Stability analysis of the hybrid interpolated scheme . . . 22

2.7 Equivalent equation . . . 23

2.8 Numerical experiments . . . 23

2.8.1 `Scilab` source code . . . 23

2.8.2 Numerical results . . . 24

3 Introduction to freefem++ **29**

3.1 Stationary elliptic problem . . . 29

3.1.1 Finite element method . . . 31

3.1.2 Practical implementation . . . 32

3.2 Heat problem . . . 32

3.2.1 Implementation in `freefem++` . . . 34

3.3 A problem of thermal engineering . . . 34

4 The Method of characteristics **41**
4.1 Mathematical setting . . . 41
4.1.1 `freefem++` source code of the pure transport problem . . . 42
4.1.2 Numerical Results . . . 43

5 Stokes equations and Navier-Stokes equations **47**
5.1 Setting of the equations . . . 47
5.2 Analysis of the stationary Stokes problem . . . 48
5.3 Numerical method for the Navier-Stokes equations . . . 51
5.4 Numerical experiments . . . 51
5.4.1 `freefem++` source code . . . 51
5.4.2 Numerical results . . . 53

6 Fractional step methods **57**
6.1 Introduction . . . 57
6.2 Continuous analysis, case of a linear system . . . 58
6.3 Strang second-order symmetric splitting . . . 59
6.4 Discrete time advance schemes . . . 60
6.4.1 First order scheme . . . 60
6.4.2 Second order schemes . . . 61
6.5 Chorin-Temam fractional step method for the Navier-Stokes equations . . . 62
6.5.1 Boundary conditions . . . 63

7 Case study. Population dynamics and migration flux analysis **65**
7.1 Lokta-Volterra equations . . . 65
7.1.1 Analysis of some qualitative properties of the solutions . . . 67
7.2 A fractional step approach to solve the Lokta-Volterra equations. . . . 68
7.3 Introducing spatial effects, population diffusion phenomenon . . . 70
7.4 Adding seasonal migration into the model . . . 73

8 Model of biological spatial pigment pattern formation **77**
8.1 Cell chemotaxis model . . . 77
8.1.1 Dimensionless equations . . . 78
8.2 Zero-dimensional model . . . 79
8.3 Linear stability analysis of the complete model . . . 79
8.3.1 Continuous variation of a single parameter . . . 81
8.4 Numerical discretization . . . 81
8.4.1 Fractional step method . . . 82
8.4.2 Full discretization . . . 83
8.4.3 `freefem++` source code of the numerical scheme and numerical results . . . 83

9 Vehicle traffic flow modeling **87**
9.1 Setting of the problem . . . 87
9.2 Some mathematical aspects of nonlinear transport equations . . . 88
9.2.1 Smooth autosimilar solutions . . . 89
9.2.2 Shock wave discontinuous solution . . . 90
9.3 Transport equation of a vehicle fraction . . . 92

9.4 System of conservation laws . . . 92
9.5 Finite difference methods for nonlinear transport equations . . . 93
9.5.1 Nonlinear extension of the Lax-Wendroff scheme . . . 94
9.6 Nonlinear Lax-Friedrichs scheme, hybrid scheme . . . 94
9.7 Numerical scheme for the transport equation of a vehicle fraction . . . 95
9.8 Numerical experiment . . . 97
9.8.1 `Scilab` source code . . . 97
9.8.2 One-dimensional numerical results . . . 99
9.9 Pseudo two-dimensional model for lane connection modeling . . . 99
9.9.1 `freefem++` source code for the viscous two-dimensional model . . . 100
9.9.2 Numerical results . . . 102

10 Biological cell migration and proliferation **105**
10.1 Biological and mathematical requirements . . . 105
10.2 Guidelines for PDE modeling . . . 106
10.3 Numerical results . . . 107
10.3.1 `freefem++` source code . . . 107
10.3.2 Numerical results . . . 108

11 Gas Dynamics **113**
11.1 Perfect gas . . . 114
11.2 Discretization in time . . . 115
11.3 Full discretization . . . 115
11.4 Numerical experiments . . . 116
11.4.1 `freefem++` source code of the supersonic flow problem around an ellipse . . . 116
11.4.2 Numerical results at infinite Mach number equal to 1.5 . . . 117
11.4.3 Numerical results at infinite Mach number equal to 4 . . . 117

12 Fluid Mechanics and heat transfer **123**
12.1 Model assumptions . . . 123
12.1.1 Dimensionless equations . . . 124
12.1.2 Boundary conditions . . . 125
12.2 Mathematical problem . . . 125
12.3 Numerical discretization . . . 127
12.3.1 Time discretization . . . 127
12.3.2 Variational formulation . . . 127
12.4 Numerical experiments . . . 127
12.4.1 `freefem++` source code of the heat transfer problem . . . 127
12.4.2 Numerical results . . . 130

13 Stochastic diffusion processes, Fokker-Planck equations **133**
13.1 Ordinary and Stochastic differential equations . . . 133
13.2 Fokker-Planck equations . . . 134
13.3 Computational approaches based on Stochastic Differential Equations . . . 135
13.3.1 Monte-Carlo methods . . . 135
13.4 Numerical solution of the Fokker-Planck equations . . . 135

13.4.1 Boundary conditions 136
13.5 Numerical example : metabolite reactions 137
13.5.1 `Scilab` source code of the Monte Carlo approach 138
13.5.2 Numerical results of the Monte-Carlo method 139
13.5.3 `freefem++` source of the Fokker-Planck solver 139
13.5.4 Numerical results with the Fokker-Planck model 141

14 Multiphase flows **145**
14.1 Setting of the equations 145
14.1.1 Transmission conditions, jump conditions 146
14.1.2 Final model 148
14.2 Semi-discretization in time 149
14.3 Front tracking by a level function 149
14.4 Application. Liquid sloshing in a box. 150
14.4.1 `freefem++` source code of the sloshing problem 150
14.4.2 Numerical results 151
14.5 Application. Injection moulding problem 151
14.5.1 `freefem++` source code of the injection moulding problem 151
14.5.2 Numerical results 155

Chapter 1

Transport equations

1.1 Pure transport equations

Let $\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$ denotes the position of a material point with initial position $\boldsymbol{x}_0$ at time t_0. If there is no ambiguity, we will denote it in the simpler way $\boldsymbol{x}(t)$. If the particle $\boldsymbol{x}(t)$ moves at velocity $\boldsymbol{u}(\boldsymbol{x}(t), t)$, then the kinematic equation writes

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u}(\boldsymbol{x}(t), t). \tag{1.1}$$

Now suppose that $\boldsymbol{x}(t)$ carries a quantity q which does not evolve during time:

$$q(\boldsymbol{x}(t), t) = C. \tag{1.2}$$

Of course we have

$$\frac{d}{dt}[q(\boldsymbol{x}(t), t)] = 0.$$

Using partial derivatives, we get

$$\frac{\partial q}{\partial t} + \frac{\partial q}{\partial \boldsymbol{x}} \frac{d\boldsymbol{x}}{dt} = 0.$$

Combining with (1.1), a standard transport equation is obtained

$$\frac{\partial q}{\partial t} + \nabla_{\boldsymbol{x}} q \cdot \boldsymbol{u} = 0. \tag{1.3}$$

The notation $\nabla_{\boldsymbol{x}}$ means that the gradient applies in the $\boldsymbol{x}$-direction. Without ambiguity, we will only denote it ∇ as usually.

In the literature, it is usual to find what is referred to as Lagrangian derivatives or particle derivative

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \nabla_{\boldsymbol{x}} \tag{1.4}$$

which is the time derivative of a quantity following particle trajectories. Of course, the transport equation (1.3) is equivalent to

$$\frac{Dq}{Dt} = 0. \tag{1.5}$$

1.1.1 Non homogeneous case

In the non homogeneous case, we have an additional nonzero right-hand side in equation (1.5) that expresses the gain or loss of the quantity q. Imagine for example a reacting chemical species that would be convected by a working fluid. Then the equation becomes

$$\frac{Dq}{Dt} = s(q, x, t) \tag{1.6}$$

or equivalently

$$\frac{\partial q}{\partial t} + \boldsymbol{u} \cdot \nabla_{\boldsymbol{x}} q = s(q, x, t). \tag{1.7}$$

The notation s for the right hand side is often used to refer to as a source term.

1.1.2 Stationary case

Let us now consider the stationary case of the non-homogeneous transport equation. That means that the field q does not depend on time directly. In particular, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x})$ and $s = s(q, x)$. Stationary flows or steady states are of practical interest because they often occur as the limit case (large time) of an unsteady process. Thus we have to solve the equation

$$\boldsymbol{u} \cdot \nabla_{\boldsymbol{x}} q = s(q, x). \tag{1.8}$$

Equation (1.8) alone does not define a mathematically well-posed problem. We need to add some boundary conditions of conditions at infinity. Let now consider a bounded domain Ω of $\mathbb{R}^d$ as shown in figure 3.1. We will denote $\partial\Omega$ the boundary of Ω, $\boldsymbol{n}(\boldsymbol{x})$ the exterior normal vector to the boundary $\partial\Omega$ at position $\boldsymbol{x}$ and $\partial\Omega^+$ (*resp.* $\partial\Omega^-$) the set

$$\partial\Omega^{\pm} = \{\boldsymbol{x} \in \partial\Omega \ / \ \pm \boldsymbol{u}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) > 0.\}\,. \tag{1.9}$$

The boundary $\partial\Omega^-$ in the context of (1.9) is referred to as the inflow boundary because of the negative

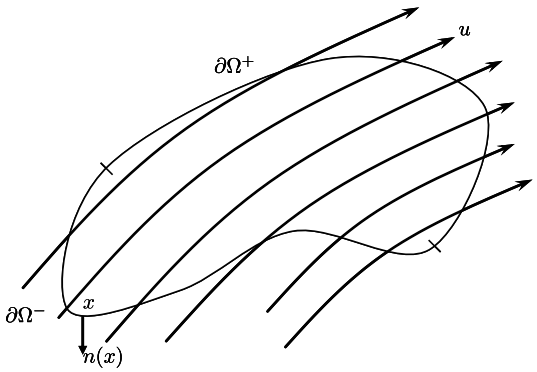


Figure 1.1: Steady transport problem in a bounded domain Ω.

sign of $\boldsymbol{u} \cdot \boldsymbol{n}$. Then we must give some information on the inflow boundary, for example Dirichlet boundary conditions

$$q = q^- \quad \text{on } \partial\Omega^- \tag{1.10}$$

for a given function q^- defined on $\partial\Omega^-$. From the Lagrangian description of equation (1.8), it is possible to analytically solve problem (1.8)-(1.10). Let $(\boldsymbol{x}(\alpha;\boldsymbol{x}_0))_{\alpha\geq 0}$ be the parameterized trajectory of a point at position $\boldsymbol{x}$ for parameter α with initial position $\boldsymbol{x}_0 \in \partial\Omega^-$. That means that $\boldsymbol{x}(\alpha;\boldsymbol{x}_0))$ is solution of the differential problem

$$\begin{cases} \dfrac{d\boldsymbol{x}}{d\alpha} = u(\boldsymbol{x}), \quad \alpha > 0, \\ \boldsymbol{x}(s(\alpha = 0)) = \boldsymbol{x}_0 \in \partial\Omega^-. \end{cases} \tag{1.11}$$

The solution of (1.11) is in integral form

$$\boldsymbol{x}(\alpha) = \boldsymbol{x}_0 + \int_0^\alpha u(\boldsymbol{x}(\alpha'))\, d\alpha'. \tag{1.12}$$

The Lagrangian form of equation (1.8) then reads

$$\frac{dq}{d\alpha} = s(q(\alpha), \boldsymbol{x}(\alpha)). \tag{1.13}$$

From the initial value $q(\alpha = 0) = q^-$, we get the analytical q solution in integral form too

$$q(\alpha) = q^- + \int_0^\alpha s(q(\alpha'), \boldsymbol{x}(\alpha'))\, d\alpha'. \tag{1.14}$$

1.2 Conservative case

Let us consider a time-dependent bounded smooth domain $\Omega_t = \Omega(t)$ which is convected itself by a velocity field $\boldsymbol{u}$ (imagine for example a bubble rising up into the water). The Reynolds theorem (see the Continuum Mechanics literature, [6] for example) states that, for a convected quantity $q = q(\boldsymbol{x}, t)$ into Ω_t, we have

$$\frac{d}{dt}\int_{\Omega_t} q(\boldsymbol{x}, t)\, d\boldsymbol{x} = \int_{\Omega_t} \left\{ \frac{\partial q}{\partial t} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{u}\, q) \right\} d\boldsymbol{x}. \tag{1.15}$$

The Reynolds theorem formula can be rigorously derived using differential calculus and the Jacobian of the transformation that maps the space-time variables $(\boldsymbol{x}, t)$ into the couple $(\boldsymbol{x}(t), t)$ where $\boldsymbol{x}(t)$ is moving according to the velocity field $\boldsymbol{u}$, i.e. $\dot{\boldsymbol{x}}(t) = \boldsymbol{u}(\boldsymbol{x}(t), t)$.

Let us focus on some corollaries of the Reynolds Theorem. First, consider $q = 1$. Denoting $|\Omega_t|$ the measure of the domain Ω_t, expression (1.15) states that

$$\frac{d|\Omega_t|}{dt} = \int_{\Omega_t} (\nabla \cdot \boldsymbol{u})\, d\boldsymbol{x}. \tag{1.16}$$

It is observed that if the velocity field is divergence-free, then any moving domain Ω_t preserves its volume. The flow generated by $\boldsymbol{u}$ is said to be incompressible. Otherwise Ω_t increases or decreases according to the sign of $\nabla \cdot \boldsymbol{u}$.

As a verification of the Reynolds Theorem, let us multiply equation (1.15) by $|\Omega_t|^{-1}$. We successively have

$$\begin{aligned}\frac{1}{|\Omega_t|}\int_{\Omega_t}\left\{\frac{\partial q}{\partial t}+\nabla_{\boldsymbol{x}}\cdot(\boldsymbol{u}\,q)\right\}d\boldsymbol{x} &= \frac{1}{|\Omega_t|}\frac{d}{dt}\int_{\Omega_t} q(\boldsymbol{x},t)\,d\boldsymbol{x}\\ &= \frac{d}{dt}\left[\frac{1}{|\Omega_t|}\int_{\Omega_t} q(\boldsymbol{x},t)\,d\boldsymbol{x}\right]-\int_{\Omega_t} q(\boldsymbol{x},t)\,d\boldsymbol{x}\,\frac{d}{dt}\left[\frac{1}{|\Omega_t|}\right]\\ &= \frac{d}{dt}\left[\frac{1}{|\Omega_t|}\int_{\Omega_t} q(\boldsymbol{x},t)\,d\boldsymbol{x}\right]+\int_{\Omega_t} q(\boldsymbol{x},t)\,d\boldsymbol{x}\,\frac{1}{|\Omega_t|^2}\frac{d|\Omega_t|}{dt}.\end{aligned}$$

From equation (1.16), we get

$$\frac{1}{|\Omega_t|}\int_{\Omega_t}\left\{\frac{\partial q}{\partial t}+\nabla_{\boldsymbol{x}}\cdot(\boldsymbol{u}\,q)\right\}d\boldsymbol{x}=\frac{d}{dt}\left[\frac{1}{|\Omega_t|}\int_{\Omega_t} q(\boldsymbol{x},t)\,d\boldsymbol{x}\right]+\left(\frac{1}{|\Omega_t|}\int_{\Omega_t} q(\boldsymbol{x},t)\,d\boldsymbol{x}\right)\left(\frac{1}{|\Omega_t|}\int_{\Omega_t}(\nabla\cdot\boldsymbol{u})\,dx\right).$$

Considering an infinitesimal volume, for $|\Omega_t|\to 0$, we get

$$\frac{\partial q}{\partial t}+\nabla_{\boldsymbol{x}}\cdot(\boldsymbol{u}\,q)=\frac{dq}{dt}+q\,\nabla_{\boldsymbol{x}}\cdot u \tag{1.17}$$

and then retrieve the particle derivative of q:

$$\frac{dq}{dt}=\frac{\partial q}{\partial t}+\boldsymbol{u}\cdot\nabla_{\boldsymbol{x}}q.$$

The Reynolds Theorem is also at the origin of the conservation laws that governs many Physics in the nature. Indeed, suppose that the production rate of q is s:

$$\frac{d}{dt}\int_{\Omega_t} q(\boldsymbol{x},t)\,d\boldsymbol{x}=\int_{\Omega_t} s(\boldsymbol{x},t)\,d\boldsymbol{x}. \tag{1.18}$$

Again, dividing (1.18) by $|\Omega_t|$, applying the Reynolds Theorem formula (1.15) and considering an infinitesimal volume Ω_t leads to the following balance equation in conservation form

$$\frac{\partial q}{\partial t}+\nabla\cdot(q\boldsymbol{u})=s(\boldsymbol{x},t). \tag{1.19}$$

Conservation forms of equations are always sought if it is possible because they provide stability and conversation properties on the solutions of the equations. For example, integrating equation (1.19) over any fixed bounded domain ω and applying Green's formula to the divergence term gives the balance equation

$$\frac{d}{dt}\int_{\omega} q(\boldsymbol{x},t)\,dx-\int_{\partial\omega} q\boldsymbol{u}\cdot\boldsymbol{n}_\omega\,d\sigma=\int_{\omega} s(\boldsymbol{x},t)\,d\boldsymbol{x}. \tag{1.20}$$

By denoting for any quantity z

$$z_\omega=\int_{\omega} z(\boldsymbol{x},t)\,d\boldsymbol{x},$$

we get the balance law

$$\frac{dq_\omega}{dt}=s_\omega+\int_{\partial\omega} q\boldsymbol{u}\cdot\boldsymbol{n}_\omega\,d\sigma. \tag{1.21}$$

The evolution of q_ω is governed by the production rate s_ω but also by the flux of q through the boundary surface ∂_ω. The flux $\Phi = q\boldsymbol{u}\cdot\boldsymbol{n}$ defines a 'mass flux' of quantity q at velocity $\boldsymbol{u}\cdot\boldsymbol{n}$ in the normal direction. That's the reason of why it is called a conservation form. For an adjacent neighboring volume ω', both volumes ω and ω' exchange some 'mass' q through a surface flux

$$\int_{\partial\omega\cap\partial\omega'} q\boldsymbol{u}\cdot\boldsymbol{n}_\omega\, d\sigma \;=\; -\int_{\partial\omega\cap\partial\omega'} q\boldsymbol{u}\cdot\boldsymbol{n}_{\omega'}\, d\sigma.$$

and in opposite sense.

1.3 Connections between transport and conservation equations.

Transport and conservation equations in their homogeneous form respectively are

$$\partial_t q + \boldsymbol{u}\cdot\nabla q = 0 \tag{1.22}$$

and

$$\partial_t q + \nabla\cdot(q\boldsymbol{u}) = 0. \tag{1.23}$$

For q smooth enough, we can of course rewrite equation (1.23) as

$$\partial_t q + \boldsymbol{u}\cdot\nabla q + q\nabla\cdot\boldsymbol{u} = 0.$$

If $\nabla\cdot\boldsymbol{u} = 0$, then both transport equation and convection equation are equivalent. Remember that a divergence-free velocity field induces an incompressible flow. Consequently, there is no effect of compressibility or dilatation and q is actually a conserved quantity.

The well-known continuity equation in Gas Dynamics expresses the conservation of the mass of the gas from its density $\rho\ [kg.m^{-3}]$ (which is an intensive variable):

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}) = 0. \tag{1.24}$$

It is interesting to notice that if c is a convected quantity in the gas (for example the concentration of a non-reacting chemical pollutant),

$$\frac{\partial c}{\partial t} + \boldsymbol{u}\cdot\nabla c = 0 \tag{1.25}$$

it is always possible to write a conservation equation, at least for smooth solutions of the equations. By multiplying by ρ equation (1.25) and by c equation (1.24), summing up we get the conservation law

$$\frac{\partial(\rho c)}{\partial t} + \nabla\cdot(\rho c\boldsymbol{u}) = 0. \tag{1.26}$$

Quantity c is not conserved (only convected) but quantity ρc is conserved (but not convected in the strict sense, except for incompressible flows).

Chapter 2

Numerical analysis of Eulerian difference schemes

In this chapter, we shall consider computational methods to solve the transport equation

$$\partial_t q + \boldsymbol{u} \cdot \nabla q = s. \tag{2.1}$$

There are many methods like finite difference (FD) schemes but also finite volume (FV) scheme, finite elements methods (FEM), method of characteristics, particle methods, etc. This chapter is aimed at introducing important concepts of numerical analysis like numerical stability, consistency and order of accuracy.

2.1 Framework

As a starting point, we first consider homogeneous one-dimensional transport equations. Suppose also that the transport velocity is constant, let say $a > 0$. We here consider the infinite spatial domain $\Omega = \mathbb{R}$. The linear transport equation is

$$\partial_t q + a\partial_x q = 0, \quad , \; x \in \mathbb{R}, \; t > 0. \tag{2.2}$$

The initial data q^0 is supposed to belong to all the L^p-spaces, $p \geq 1$ and $p = \infty$. Of course, the solution q of (2.2) is analytically known, namely

$$q(x,t) = q^0(x - at) \quad \forall t > 0. \tag{2.3}$$

In particular we have

$$||q(.,t)||_{L^p} = ||q^0||_{L^p}, \quad \forall t > 0, \; p \geq 1, \; p = \infty. \tag{2.4}$$

The solution for the continuous problem is stable in L^p-norm meaning that it does not blow up in time, and even stays constant in time. This is a good framework to study the numerical stability of discrete time advance schemes.

2.2 Finite difference schemes

We shall discretize the PDE problem (2.2),(2.3). Let us consider a uniform mesh step $h > 0$ and the discretization points

$$x_j = jh, \quad j \in \mathbb{Z}. \tag{2.5}$$

We also consider a time discretization, not necessary uniform. By defining the initial discrete instant $t^0 = 0$, we build a sequences $(t^n)_{n\in\mathbb{N}}$ from a time step sequences $(\Delta t^n > 0)_{n\in\mathbb{N}}$ such that

$$t^{n+1} = t^n + \Delta t^n. \tag{2.6}$$

At each discretization point, we consider q_j^n which is a discretization of the continuous solution q at spatial point x_j and instant t^n, i.e.

$$q_j^n \approx q(x_j, t^n). \tag{2.7}$$

As starting point, one can define the values $(q_j^0)_{j\in\mathbb{Z}}$ from the continuous initial data q^0 by exactly taking

$$q_j^0 = q^0(x_j), \quad j \in \mathbb{Z}. \tag{2.8}$$

The time advance finite difference scheme consists in computing new values $(q_j^{n+1})_{j\in\mathbb{Z}}$ from already known values $(q_j^n)_{j\in\mathbb{Z}}$ at the previous instant t^n using a discrete version of (2.2).

They are many ways to discretize (2.2) in both space and time. A semi-discretization in time can be

$$\frac{q(x, t^{n+1}) - q(x, t^n)}{\Delta t^n} + a\partial_x q(x, t^n) = 0 \tag{2.9}$$

using the backward Euler scheme formula, which defines a time-explicit scheme, or

$$\frac{q(x, t^{n+1}) - q(x, t^n)}{\Delta t^n} + a\partial_x q(x, t^{n+1}) = 0 \tag{2.10}$$

using the forward Euler scheme formula, which defines a time-implicit one and requires a linear inversion process to get all the q_j^{n+1}. Of course, one could also use time discrete schemes more accurate than the first order Euler formula. Let us focus on the time discrete scheme (2.9). To get a really computational scheme, we also discretize (2.9) in space. There are again many ways to approximate $\partial_x q$ at point (x_j, t^n). For example, the centered difference scheme

$$(\partial_x q)(x_j, t^n) \approx \frac{q_{j+1}^n - q_{j-1}^n}{2h} \tag{2.11}$$

defines the centered scheme. The upwind discretization rule

$$(\partial_x q)(x_j, t^n) \approx \frac{q_j^n - q_{j-1}^n}{h} \tag{2.12}$$

defines the so-called explicit Euler upwind scheme whereas the downwind rule

$$(\partial_x q)(x_j, t^n) \approx \frac{q_{j+1}^n - q_j^n}{h} \tag{2.13}$$

defines the explicit Euler downwind scheme. As a matter of fact, some of numerical presented above are numerically stable whereas some of them are not.

2.3 Numerical stability

For numerical stability purposes it is convenient to consider the ℓ^p norms of the sequences $(q_j^n)_{j\in\mathbb{Z}}$ defined by

$$||q^n||_p = \left(\sum_{j\in\mathbb{Z}} |q_j^n|^p\right)^{1/p}, \quad p \in \mathbb{N}, \tag{2.14}$$

$$||q^n||_\infty = \sup_{j\in\mathbb{Z}} |q_j^n|. \tag{2.15}$$

Let us first consider the explicit upwind scheme with Euler time discretization ($a > 0$):

$$\frac{q_j^{n+1} - q_j^n}{\Delta t^n} + a\,\frac{q_j^n - q_{j-1}^n}{h} = 0. \tag{2.16}$$

2.3.1 ℓ_∞ stability

Generally, ℓ_∞ analysis is used to get some information on the extrema of the sequences.

Is is easy to show that (2.16) is ℓ^∞-stable under a condition of the time step Δt^n. This scheme can be rewritten in incremental form

$$q_j^{n+1} = \left(1 - a\frac{\Delta t^n}{h}\right) q_j^n + a\frac{\Delta t^n}{h} q_{j-1}^n. \tag{2.17}$$

It is observed that q_j^{n+1} is nothing else but a linear convex combination of q_j^n and q_{j-1}^n provided that Δt^n is such that

$$a\frac{\Delta t^n}{h} \leq 1. \tag{2.18}$$

The condition (2.18) is referred to as the Courant-Friedrichs-Lewy condition or simply CFL condition. It is usual to introduce the so-called local Courant number

$$\nu^n = a\frac{\Delta t^n}{h}. \tag{2.19}$$

Because convex combinations are stable in ℓ_∞ norm, the CFL condition (2.18) defines a sufficient condition of ℓ_∞ stability. Actually it is also a necessary condition. Consider the particular initial sequences $q_j^0 = 1_{(j\leq 0)}(j)$ with $\nu^n = \nu > 1$ for all $n \in \mathbb{N}$. Then it is easy to check that $q_1^1 = \nu$, $q_2^2 = \nu^2, \ldots, q_k^k = \nu^k\ \forall k \geq 1$ so that for any $M > 0$ there is always a rank n_0 such that for all $n \geq 0$, we have $||q^n||_\infty > M$.

Remark that the limit stability case $v^n = 1$ (Courant number exactly equal to one) leads to the numerical scheme

$$q_j^{n+1} = q_{j-1}^n, \tag{2.20}$$

which is compatible with the structure of the exact continuous solution

$$q(x, t^{n+1}) = q(x - a\Delta t^n, t^n) = q(x - h, t^n). \tag{2.21}$$

To summarize, let us enunciate the following theorem

Theorem 1. *The explicit upwind scheme (2.16) is conditionally ℓ_∞-stable*

$$||q^{n+1}||_\infty \leq ||q^n||_\infty \quad \forall n \in \mathbb{N}. \tag{2.22}$$

with the stability CFL condition (2.18) on the time step. For a Courant number $\nu = 1$, the scheme exactly propagates the discretized initial condition.

2.3.2 ℓ_1 stability

The ℓ_1-norms are practical to study the appearance of alternative oscillating discrete patterns in discrete solutions.
Let start again from the Euler upwind scheme using the Courant number :

$$q_j^{n+1} = (1 - \nu^n)q_j^n + \nu^n q_{j-1}^n. \tag{2.23}$$

Consider first a Courant number ν^n less than 1. Taking the absolute value of the expression and applying the triangular inequality gives

$$|q_j^{n+1}| \leq (1 - \nu^n)|q_j^n| + \nu^n |q_{j-1}^n|. \tag{2.24}$$

Suppose that $||q^n||_1 \leq +\infty$. Summing up (2.24) over the j then gives

$$\sum_{j\in\mathbb{Z}} |q_j^{n+1}| \leq (1 - \nu^n) \sum_{j\in\mathbb{Z}} |q_j^n| + \nu^n \sum_{j\in\mathbb{Z}} |q_{j-1}^n|.$$

and thus

$$||q^{n+1}||_1 \leq ||q^n||_1. \tag{2.25}$$

The ℓ_1-norm is decreasing during time iterations so the numerical method is ℓ_1-stable for Courant numbers less than one. On the other hand it is easy to build a counterexample thats shows that the condition $\nu^n = \nu > 1$ makes the numerical scheme unstable. Consider for example a discrete initial condition $(q_j^0) \in \ell_1(\mathbb{Z})$ such that

$$q_j^0 . q_{j+1}^0 < 0 \quad \forall j \in \mathbb{Z}. \tag{2.26}$$

(it is an alternate discrete function). Then for $\nu > 1$, it is easy to check that the alternate property is preserved during the iterations

$$q_j^n . q_{j+1}^n < 0 \quad \forall j \in \mathbb{Z},\ n \in \mathbb{N} \tag{2.27}$$

with also

$$q_j^n . q_j^{n+1} < 0 \quad \forall j \in \mathbb{Z},\ n \in \mathbb{N}. \tag{2.28}$$

From the scheme (1.24), one can notice that

$$q_j^{n+1} = sgn(q_{j-1}^n)|1 - \nu||q_j^n| + \nu q_{j-1}^n$$

so that

$$|q_j^{n+1}| > \nu |q_{j-1}^n| \quad \forall j \in \mathbb{Z}$$

or again

$$||q^n||_1 > \nu^n ||q^0||_1 \quad \forall n \in \mathbb{N} \tag{2.29}$$

making the numerical scheme unstable. We thus get the same stability results as in the previous ℓ_∞ case.

2.3.3 ℓ_2 stability, von Neumann stability

The von Neumann ℓ_2-stability gives information of the evolution of the energy of the solution and on the frequency spectrum. For that we use the Fourier transform

$$\mathscr{F}(f)(\xi) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x) e^{-ix\xi}\, dx \tag{2.30}$$

for $f \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. it is known that the Fourier transform is an isometry: $\mathscr{F}(f) \in L^2(\mathbb{R})$ and

$$||f||_{L^2} = ||\mathscr{F}(f)||_{L^2}. \tag{2.31}$$

We will also make use of the common properties of the Fourier transforms like

$$\mathscr{F}(\mathcal{T}_h f)(\xi) = e^{ih\xi} \mathscr{F}(f)(\xi) \tag{2.32}$$

where $\mathcal{T}_h f(x) = f(x-h)$.

To simplify the stability analysis in the sense of von Neumann, it is more convenient to deal with a semi-implicit version of the explicit Euler upwind scheme:

$$\frac{q^{n+1}(x) - q^n(x)}{\Delta t^n} + a\, \frac{q^n(x) - q^n(x-h)}{h} = 0, \quad \forall x \in \mathbb{R}. \tag{2.33}$$

The Fourier transform is a linear transform. Applying it on equation (2.33) and using (2.32) gives the expression

$$\mathscr{F}(q^{n+1})(\xi) - \mathscr{F}(q^n)(\xi) + \nu^n \left(\mathscr{F}(q^n)(\xi) - e^{ih\xi} \mathscr{F}(q^n)(\xi) \right) = 0, \quad \forall \xi \in \mathbb{R}. \tag{2.34}$$

or again

$$\mathscr{F}(F)(q^{n+1})(\xi) = \left[1 - \nu^n (1 - e^{ih\xi}) \right] \mathscr{F}(q^n)(\xi). \tag{2.35}$$

We have to check for which values of ν^n the modulus of the amplification factor $a(h\xi) = 1 - \nu^n(1 - e^{ih\xi})$ is less than one, for any frequency ξ. One finds

$$\begin{aligned} |a(h\xi)|^2 &= [1 - \nu^n(1 - e^{ih\xi})][1 - \nu^n(1 - e^{-ih\xi})] \\ &= 1 - 2\nu^n(1 - \cos(h\xi)) + (\nu^n)^2 |1 - e^{ih\xi}|^2 \\ &= 1 - 4\nu^n \sin^2(h\xi/2) + 4(\nu^n)^2 \sin^2(h\xi/2) \leq 1. \end{aligned}$$

That gives the condition

$$4 \sin^2(h\xi/2) \nu^n (\nu^n - 1) \leq 0 \quad \forall \xi \in \mathbb{R}$$

equivalent again to

$$\nu^n \leq 1. \tag{2.36}$$

Under condition (2.36) and due to the isometry property of the Fourier transform, we have

$$||q^{n+1}||_{L^2} = ||\mathscr{F}(q^{n+1})_{L^2}|| \leq ||\mathscr{F}(q^n)||_{L^2} = ||q^n||_{L^2}. \tag{2.37}$$

As exercise, we let the reader show that the explicit downwind Euler scheme as well as the centered scheme with Euler time discrete formula are unconditionally unstable in the sense of von Neumann. We summarize by the following theorem:

Theorem 2. *The explicit upwind scheme is conditionally stable in the sense of von Neumann under Courant numbers less than one. The explicit downwind scheme as well as the centered schemes are unconditionally unstable.*

2.4 Consistency properties

Another important feature of a discretization scheme is the consistency property and the order of accuracy. Let us analyze the order of accuracy the the explicit upwind scheme. For the sake of simplicity, we will restrict our analysis again to the one-dimensional case. Let us once again recall the expression of the explicit upwind scheme:

$$\frac{q_j^{n+1} - q_j^n}{\Delta t^n} + a \frac{q_j^n - q_{j-1}^n}{h} = 0. \tag{2.38}$$

The error of consistency consists in measuring the residual computed from the exact solution of the continuous problem

$$R_j^n(h, \Delta t^n) = \frac{q(x_j, t^{n+1}) - q(x_j, t^n)}{\Delta t^n} + a \frac{q(x_j, t^n) - q(x_{j-1}, t^n)}{h}. \tag{2.39}$$

If it is possible to write Taylor expansions of the exact solution near points (x_j, t^n) (this is possible with enough regularity), there exists $\theta^n \in (t^n, t^{n+1})$ such that

$$q(x_j, t^{n+1}) = q(x_j, t^n) + \Delta t^n \partial_t q(x_j, t^n) + \frac{1}{2}(\Delta t^n)^2 \partial_{tt}^2 (x_j, \theta^n)$$

and $y_j \in (x_{j-1}, x_j)$ such that

$$q(x_{j-1}, t^n) = q(x_j, t^n) - h\, \partial_t q(x_j, t^n) + \frac{1}{2} h^2 \partial_{xx}^2 q(y_n, t^n)$$

Then we get the residual

$$R_j^n(h, \Delta t^n) = \frac{1}{2} \Delta t^n \partial_{tt}^2 (x_j, \theta^n) - \frac{1}{2} a h \partial_{xx}^2 q(y_n, t^n). \tag{2.40}$$

If both $\partial_{tt}^2 q$ and ∂_{xx}^2 are bounded functions, then equation (2.40) shows that

$$R_j^n(h, \Delta t^n) = O(\Delta t^n) + O(h) \quad \forall j \in \mathbb{Z}, n \in \mathbb{N}. \tag{2.41}$$

The explicit upwind scheme is said to be first order accurate (in both space and time). It is possible to build counterexamples to show that this cannot be better than (2.41).

There are some numerical schemes that provide a smaller rate or error of consistency or equivalently a higher order of accuracy. Let us give the definition

Definition 1 (order of accuracy)**.** *A numerical scheme is said to be p-th order accurate in space and q-th order accurate in time if its residual is in the form*

$$R_j^n(h, \Delta t^n) = O(h^p) + O((\Delta t^n)^q) \quad \forall j \in \mathbb{Z}, n \in \mathbb{N}. \tag{2.42}$$

2.4.1 Lax-Wendroff scheme

It is of course of interest to define a methodology of construction of higher order accurate scheme. We shall construct now a second-order accurate scheme in both space and time. The solutions of the transport are supposed to be smooth enough to write high-order derivatives. Derivating

$$\partial_t q + a \partial_x q = 0,$$

with respect to t allows us to write:

$$\partial_{tt}^2 q = a^2 \partial_{xx}^2 q.$$

By writing a Taylor expansion in time up to the third order, we have

$$q(x_j, t^{n+1}) = q(x_j, t^n) + \Delta t^n \partial_t q(x_j, t^n) + \frac{1}{2}(\Delta t^n)^2 a^2 \partial_{xx}^2 (x_j, t^n) + O(\Delta t^3).$$

So it appears interesting to define a discrete time derivative in the form

$$\partial_t q(x_j, t^n) \approx \frac{q(x_j, t^{n+1}) - q(x_j, t^n)}{\Delta t^n} - \frac{1}{2} a^2 \Delta t^n \frac{-q(x_{j+1}, t^n) + 2q(x_j, t^n) - q(x_{j-1}, t^n)}{h^2}. \quad (2.43)$$

whose error of consistency is

$$O((\Delta t^n)^2) + O(h\, \Delta t^n)$$

or again

$$O((\Delta t^n)^2) + O(h^2)$$

(indeed $ab \leq \frac{1}{2}(a^2 + b^2)$). For the spatial derivative, we simply use the centered finite difference which is known to be second-order accurate:

$$(\partial_x q)(x_j, t^n) \approx \frac{q(x_{j+1}, t^n) - q(x_{j-1}, t^n)}{2h} \quad (2.44)$$

with consistency error

$$O(h^2).$$

The two discretization schemes (2.43) and (2.44) lead to the so-called Lax-Wendroff scheme (exercise):

Definition 2. *The Lax-Wendroff scheme for the transport equation (1.2) is the following second order accurate scheme:*

$$\frac{q_j^{n+1} - q_j^n}{\Delta t^n} + \frac{\Phi_{j+1/2}^{LW} - \Phi_{j-1/2}^{LW}}{h} = 0 \quad (2.45)$$

where the so-called Lax-Wendroff flux is

$$\Phi_{j+1/2}^{LW} = \frac{a q_j^n + a q_{j+1}^n}{2} - \frac{1}{2} \nu^n a (q_{j+1}^n - q_j^n). \quad (2.46)$$

still with Courant number

$$\nu^n = \frac{a \Delta t^n}{h}. \quad (2.47)$$

The script (2.46),(2.47) is interesting because it show that the numerical scheme has the conservation form (2.46). The numerical flux (2.47) is made of a centered flux and an artificial spatial viscosity term. The numerical flux if second-order consistent with

$$\Phi = aq - \frac{1}{2} \nu^n a h \partial_x q$$

Remark 1. *Remark that the upwind explicit scheme can also be written in conservation form*

$$\frac{q_j^{n+1} - q_j^n}{\Delta t^n} + \frac{\Phi_{j+1/2}^{up} - \Phi_{j-1/2}^{up}}{h} = 0 \quad (2.48)$$

with upwind numerical flux

$$\Phi_{j+1/2}^{up} = \frac{a q_j^n + a q_{j+1}^n}{2} - \frac{1}{2} |a| (q_{j+1}^n - q_j^n) \quad (2.49)$$

(exercise).

2.4.2 Von-Neumann stability of the Lax-Wendroff scheme

We shall study the ℓ_2-stability of the Lax-Wendroff scheme. For that, as mentioned above we consider a continuous-in-space version of that numerical scheme:

$$q^{n+1}(x) - q^n(x) + \frac{\Delta t^n}{h}\left(\Phi^{LW}(x, x+h) - \Phi^{LW}(x-h, x)\right) = 0 \tag{2.50}$$

with

$$\Phi^{LW}(x, x+h) = \frac{aq^n(x) + aq^n(x+h)}{2} - \frac{1}{2}a\nu^n(q^n(x+h) - q^n(x)).$$

By taking the Fourier transform of this expression, we get

$$\begin{aligned}\hat{q}^{n+1}(\xi) - \hat{q}^n(\xi) \quad &+ \quad \nu^n\left\{\frac{1+e^{-ih\xi}}{2} - \frac{1}{2}\nu^n(e^{-ih\xi}-1)\right\}\hat{q}^n(\xi) \\ &- \quad \nu^n\left\{\frac{1+e^{ih\xi}}{2} - \frac{1}{2}\nu^n(1-e^{ih\xi})\right\}\hat{q}^n(\xi) = 0\end{aligned}$$

that simplifies into

$$\hat{q}^{n+1}(\xi) = \left[1 - \nu^n\{\nu^n(1-\cos(h\xi)) - i\sin(h\xi)\}\right]\hat{q}^n(\xi), \quad \forall \xi \in \mathbb{R}. \tag{2.51}$$

The amplification factor is

$$a(h\xi) = 1 - \nu^n\{\nu^n(1-\cos(h\xi)) - i\sin(h\xi)\}.$$

To find the stability condition, the inequality

$$|a(h\xi)|^2 \leq 1.$$

has to be solved. Using the identity $\sin^2(h\xi) = 1 - \cos^2(h\xi)$, it is found that the condition is equivalent to

$$(1-\cos^2(h\xi))^2\,(\nu^n)^2((\nu^n)^2 - 1) \leq 0,\ \forall \xi \in \mathbb{R}.$$

That shows that the Lax-Wendroff scheme is ℓ_2-stable for Courant numbers less than one.

2.5 Lax-Friedrichs scheme

The Lax-Friedrichs scheme is another stable candidate based on the following approximation scheme:

$$\partial_x q(x_j, t^n) \approx \frac{q^n_{j+1} - q^n_{j-1}}{2h} \quad \text{(second order accurate)},$$

$$\partial_t q(x_j, t^n) \approx \frac{q^{n+1}_j - \frac{q^n_{j-1}+q^n_{j+1}}{2}}{\Delta t^n} \quad \text{(first order accurate)}.$$

It is easy to check that the residual of the discrete time derivative can be written

$$\frac{q(x_j, t^{n+1}) - \frac{q(x_{j-1},t^n)+q(x_{j+1},t^n)}{2}}{\Delta t^n} - \partial_t q(x_j, t^n) = \frac{\Delta t^n}{2}\partial^2_{tt} q(x_j, \theta^n) + \frac{1}{2}\frac{h^2}{\Delta t^n}\frac{\partial^2_{xx} q(\xi_j^-, t^n) + \partial^2_{xx} q(\xi_j^+, t^n)}{2} \tag{2.52}$$

for $\theta^n \in (t^n, t^{n+1})$, $\xi_j^- \in (x_{j-1,x_j})$ and $\xi_j^+ \in (x_j, x_{j+1})$. Expression (2.52) shows that the approximation stays first order accurate provided that h and Δt^n are of the same order, or more exactly

$$\left| \frac{h}{\Delta t^n} \right| \leq C, \tag{2.53}$$

where C is a constant which is independent from h and Δt^n. Remark that this could not be the case for possibly restrictive stability conditions like for example

$$\frac{\Delta t^n}{h^2} \leq C'.$$

In the next subsection, we will do the von Neumann stability analysis of the Lax-Friedrichs scheme. So the Lax-Friedrichs scheme is written as

$$\frac{q_j^{n+1} - \frac{q_{j-1}^n + q_{j+1}^n}{2}}{\Delta t^n} + a\,\frac{q_{j+1}^n - q_{j-1}^n}{2h} = 0. \tag{2.54}$$

It is easy to check that is can written in conservation form

$$q_j^{n+1} = q_j^n - \frac{\Delta t^n}{h}\left(\Phi_{j+1/2}^{FL} - \Phi_{j-1/2}^{LF}\right) \tag{2.55}$$

where the Lax-Friedrichs numerical flux is

$$\Phi_{j+1/2}^{LF} = \frac{aq_j^n + aq_{j+1}^n}{2} - \frac{1}{2\nu^n}\,a(q_{j+1}^n - q_j^n). \tag{2.56}$$

We again recognize a centered flux plus an artificial viscosity term.

2.5.1 Von Neumann stability analysis of the Lax-Friedrichs scheme

We consider a continuous-in-space version of the Lax-Friedrichs scheme:

$$q^{n+1}(x) - q^n(x) + \frac{\Delta t^n}{h}\left(\Phi^{LF}(x, x+h) - \Phi^{LF}(x-h, x)\right) = 0 \tag{2.57}$$

with

$$\Phi^{LF}(x, x+h) = \frac{aq^n(x) + aq^n(x+h)}{2} - \frac{1}{2\nu^n}a(q^n(x+h) - q^n(x)).$$

By taking the Fourier transform of this expression, we get

$$\begin{aligned}\hat{q}^{n+1}(\xi) - \hat{q}^n(\xi) \quad &+ \quad \nu^n\left\{\frac{1+e^{-ih\xi}}{2} - \frac{1}{2\nu^n}(e^{-ih\xi} - 1)\right\}\hat{q}^n(\xi) \\ &- \quad \nu^n\left\{\frac{1+e^{ih\xi}}{2} - \frac{1}{2\nu^n}(1 - e^{ih\xi})\right\}\hat{q}^n(\xi) = 0\end{aligned}$$

that simplifies into

$$\hat{q}^{n+1}(\xi) = \left[\cos(h\xi) - i\nu^n \sin(h\xi)\}\right]\hat{q}^n(\xi), \quad \forall \xi \in \mathbb{R}. \tag{2.58}$$

The amplification factor is

$$a(h\xi) = \cos(h\xi)) - i\nu^n \sin(h\xi)\}.$$

The stability condition inequality $|a(h\xi)|^2 \leq 1$ writes

$$\cos^2(h\xi) + (\nu^n)^2 \sin^2(h\xi) \leq 1, \tag{2.59}$$

clearly showing that the Lax-Friedrichs is stable in the von Neumann sense for Courant numbers less than one.

2.6 Hybrid interpolated fluxes

So far, we have seen three different stable schemes, namely the upwind (UP) scheme, the Lax-Wendroff (LW) scheme and the Lax-Friedrichs (LF) scheme. All three can be written in conservation form

$$q_j^{n+1} = q_j^n - \frac{\Delta t^n}{h}\left(\Phi_{j+1/2}^n - \Phi_{j-1/2}^n\right), \tag{2.60}$$

with different numerical flux $\Phi_{j+1/2}^n$ for each of them. Let us recall the three numerical fluxes written for any a (can be positive or negative):

$$\Phi_{j+1/2}^{UP} = \Phi_{j+1/2}^c - \frac{1}{2}|a|(q_{j+1}^n - q_j^n), \tag{2.61}$$

$$\Phi_{j+1/2}^{LW} = \Phi_{j+1/2}^c - \frac{1}{2}\nu^n|a|(q_{j+1}^n - q_j^n), \tag{2.62}$$

$$\Phi_{j+1/2}^{LF} = \Phi_{j+1/2}^c - \frac{1}{2\nu^n}|a|(q_{j+1}^n - q_j^n), \tag{2.63}$$

where

$$\Phi_{j+1/2}^c = \frac{aq_j^n + aq_{j+1}^n}{2}$$

is the second order centered flux. It is interesting so see that all three share the same structure and can even be written under the generic form

$$\Phi_{j+1/2}^n(\theta) = \Phi_{j+1/2}^c - \frac{1}{2}(\nu^n)^\theta|a|(q_{j+1}^n - q_j^n) \tag{2.64}$$

using a parameter $\theta \in [-1, 1]$. The UP scheme corresponds to $\theta = 0$ whereas $\theta = 1$ gives the LW scheme and $\theta = -1$ gives the LF scheme. To implement the three numerical scheme, we only need to implement the numerical method with the free parameter $\theta \in [-1, 1]$ and use it as a configuration parameter. For parameters θ different from $-1, 0, 1$, the numerical method (2.60),(2.64) defines a natural interpolation of the three common schemes.

2.6.1 Stability analysis of the hybrid interpolated scheme

For the hybrid interpolated scheme, the amplification relation then becomes

$$\hat{q}^{n+1}(\xi) = \left[1 - \nu^n\left((\nu^n)^\theta(1 - \cos(h\xi)) - i\sin(h\xi)\right)\right]\hat{q}^n(\xi). \tag{2.65}$$

The stability condition

$$|a(h\xi)|^2 \leq 1 \quad \forall \xi \in \mathbb{R}$$

reads (denoting $\omega = h\xi$ and $\nu^n = \nu$ for simplicity)

$$\nu^2 \sin^2\omega + \nu^{2(1+\theta)}(1 - \cos\omega)^2 - 2\nu^{1+\theta}(1 - \cos\omega) \leq 0.$$

By denoting $y = \cos\omega$, this can be simplified into

$$\nu^2(1 + y) + \nu^{2(1+\theta)}(1 - y) - 2\nu^{1+\theta} \leq 0 \quad \forall y \in [-1, 1) \tag{2.66}$$

(the case $y = 1$ is automatically satisfied). The analysis can be finalized by a simple study of function (as exercise).

2.7 Equivalent equation

It is of interest to exhibit what kind of equation a first order scheme solves at second order accuracy. This is exactly the definition of the equivalent equation. Let us write the equivalent equation for the hybrid interpolated scheme. Using Taylor expansion, it is an easy matter of fact to show that

$$\frac{q_j^{n+1}-q_j^n}{\Delta t^n} \approx \partial_t q(x_j,t^n) + \frac{\Delta t^n}{2}\partial_{tt}^2 q(q_j,t^n) + O((\Delta t^n)^2)$$

and

$$\frac{\Phi_{j+1/2}^n - \Phi_{j-1/2}^n}{h} \approx a\partial_x q(x_j,t^n) - \frac{h}{2}(\nu^n)^\theta |a| \partial_{xx}^2 q(x_j,t^n) + O(h^2).$$

Consequently, truncating up to second order terms gives the equivalent equation

$$\partial_t q + a\partial_x q = \frac{h}{2}(\nu^n)^\theta |a| \partial_{xx}^2 q - \frac{\Delta t^n}{2}\partial_{tt}^2 q. \tag{2.67}$$

From $\partial_{tt}^2 q = a^2 \partial_{xx}^2 q$, one can also write the convection-diffusion-like equation

$$\partial_t q + a\partial_x q - \left(\frac{h}{2}(\nu^n)^\theta |a| - \frac{\Delta t^n}{2}a^2\right)\partial_{xx}^2 q = 0. \tag{2.68}$$

In order to have a positive diffusion (necessary for stability at the continuous level), this requires

$$\frac{h}{2}(\nu^n)^\theta |a| \geq \frac{\Delta t^n}{2}a^2$$

which can also be written

$$(\nu^n)^{1-\theta} \leq 1. \tag{2.69}$$

and defines a necessary condition of stability. Notice that the second order term vanishes for $\theta = 1$, showing once again that the Lax-Wendroff scheme is second order accurate (in both space and time).

2.8 Numerical experiments

2.8.1 `Scilab` source code

```
// Advect.sce (Scilab)
// Numerical schemes for the pure advection equation
// Interpolation parameter theta in [-1,1];
// Courant number nu<=1;
theta = -1;
N = 1000;
h = 1 / N;
x = h/2 : h: 1-h/2;
key = 3;
// Initial condition
if (key==1) then
// Step solution
u = 0 + ((1/4-abs(x-1/2))>0);
//
```

```
elseif (key==2) then
// Pyramid-shaped function
u = max(0, 1-4*abs(x-1/2));
//
else
// Sine function
u = sin(2*%pi*x);
end;
plot(x, u, '.-'); xgrid();
nu = 0.5;
a = 1;
time = 0;
lambda = nu / abs(a);
dt = h * lambda;
//
for it=1:2e3
  time = time + dt;
  Phi = 0.5*a*([u, u(1)] + [u(N), u]) ...
     - 0.5*nu^theta*abs(a)*([u, u(1)] - [u(N), u]);
  u = u - lambda * (Phi(2:N+1) - Phi(1:N));
  //
  // Comparison with exact solution
  //
  xmodulo = modulo(a*time,1);
  if (key==1) then
  // Step solution
    uex = 0 + ((1/4-abs(x-xmodulo-1/2))>0) ...
       + ((1/4-abs(x-xmodulo+1/2))>0);
  elseif (key==2)
  // Pyramid function
    uex = max(0, 1-4*abs(x-xmodulo-1/2)) ...
       + max(0, 1-4*abs(x-xmodulo+1/2));
  else
  // Sine function
    uex = sin(2*%pi*(x-xmodulo)) ;
  end;
  if ~modulo(it, 50) then
    drawlater();
    clf();
    subplot(1,2,1), plot(x, u, '.-', x, uex, '-'); xgrid();
    xtitle('Discrete solution (uh)');
    subplot(1,2,2), plot(x, u-uex, '.-'); xgrid();
    xtitle('Error (uh-u_ex)');
    drawnow();
  end;
end;
```

2.8.2 Numerical results

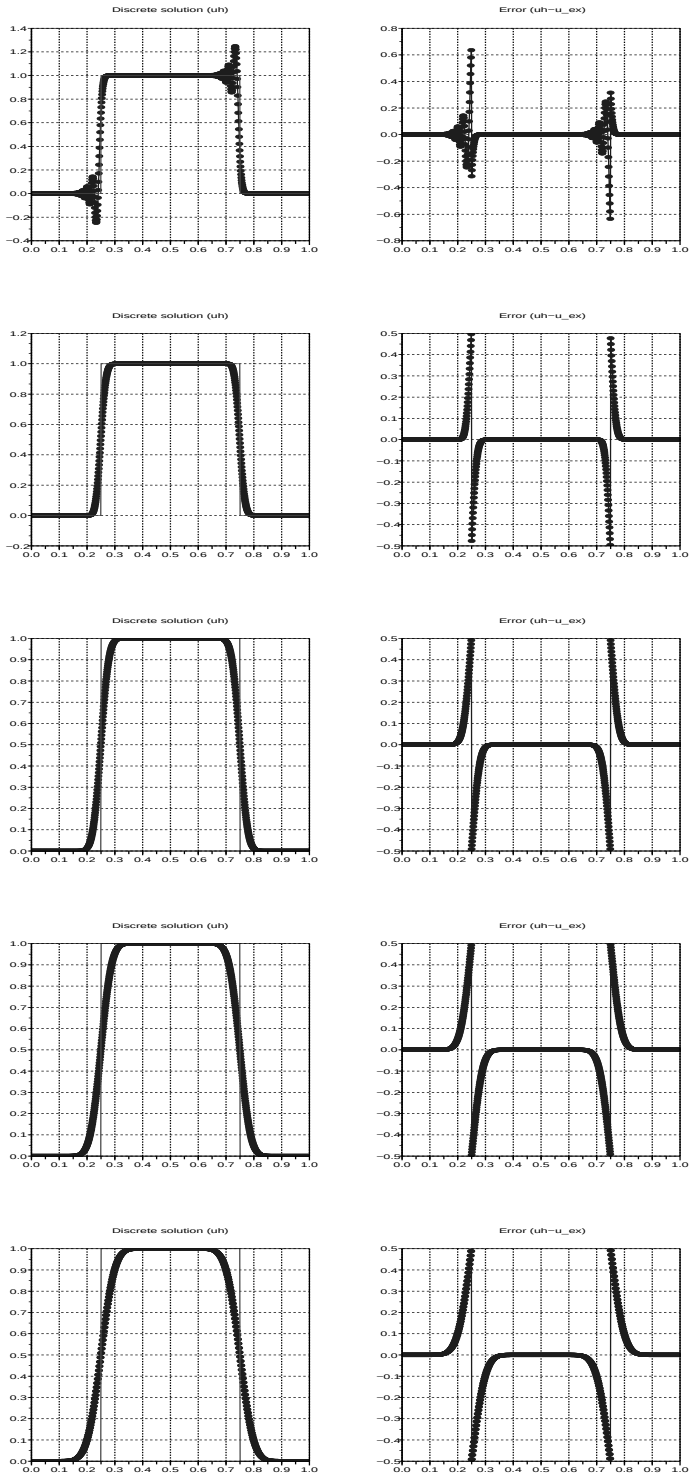


Figure 2.1: Numerical solution with the step function as initial condition. Respective discrete solutions and errors for $\theta = 1,\ \frac{1}{2},\ 0,\ -\frac{1}{2},\ -1$ ($\nu^n = \frac{1}{2}$ is used here, final time is $T = 1$).

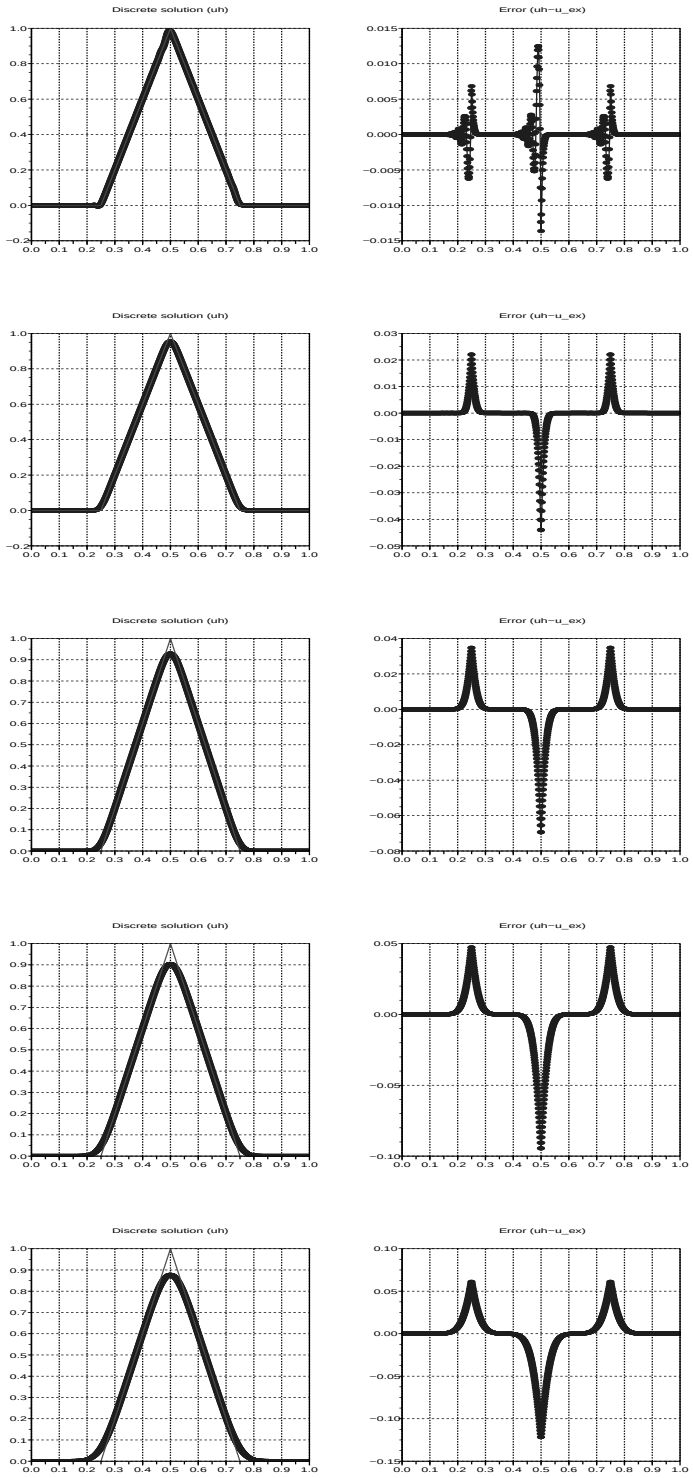


Figure 2.2: Numerical solution with the pyramid-shaped function as initial condition. Respective discrete solutions and errors for $\theta = 1,\ \frac{1}{2},\ 0,\ -\frac{1}{2},\ -1$ ($\nu^n = \frac{1}{2}$ is used here, final time is $T = 1$).

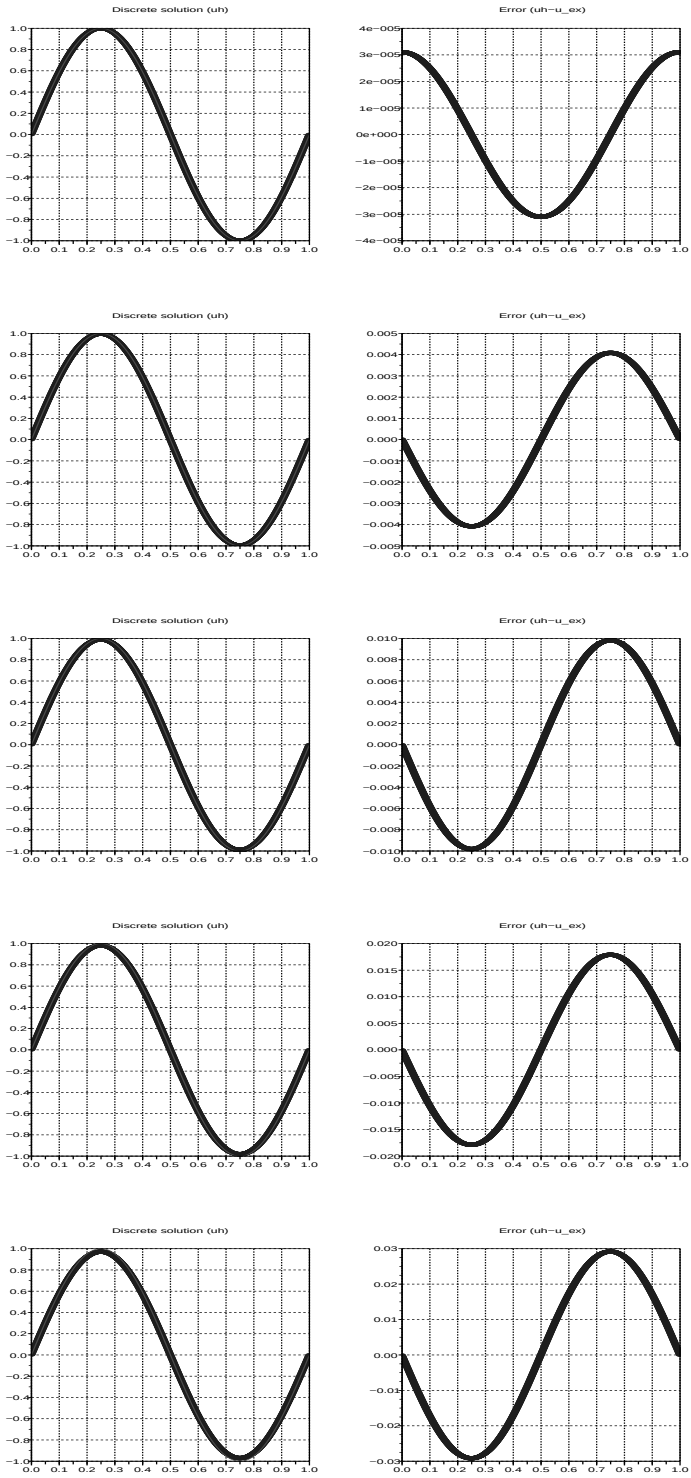


Figure 2.3: Numerical solution with the sine function as initial condition. Respective discrete solutions and errors for $\theta = 1,\ \frac{1}{2},\ 0,\ -\frac{1}{2},\ -1$ ($\nu^n = \frac{1}{2}$ is used here, final time is $T = 1$).

Chapter 3

Introduction to freefem++

`freefem++` is a language and an integrated development environment with visualization facilities for the Finite Element Method (FEM). It is dedicated to the simulation of two-dimensional or three-dimensional, steady or unsteady, linear or nonlinear Partial Differential problems defined from their variational formulation. `freefem++` is written in C++ and the `freefem++` language is a C++ idiom. The software can be downloaded at

`http://www.freefem.org/ff++/`.

The software is well-documented. The user will find the `freefem++` manual and various examples at

`http://www.freefem.org/ff++/ftp/freefem++doc.pdf`

3.1 Stationary elliptic problem

`freefem++` is able to solve Laplace problems and general elliptic problems in only a few lines. The problem has to be set from its discrete variational formulation. As example, consider the elliptic problem with Robin boundary conditions ($\delta, \varepsilon > 0$)

$$-\Delta u + \delta u = 1 \quad \text{in } \Omega, \tag{3.1}$$

$$\frac{\partial u}{\partial \boldsymbol{n}} + \varepsilon u = 0 \quad \text{on } \partial\Omega. \tag{3.2}$$

To write the variational formulation of the problem, first multiply (3.1) by a smooth test function v and then integrate it over the bounded domain Ω. Applying Green's formula gives

$$\int_\Omega \nabla u \cdot \nabla v \, d\boldsymbol{x} - \int_{\partial\Omega} \frac{\partial u}{\partial \boldsymbol{n}} v \, d\sigma + \delta \int_\Omega uv \, d\boldsymbol{x} = \int_\Omega v \, d\boldsymbol{x} \tag{3.3}$$

Now, due to the boundary condition (3.2), it can be rewritten

$$\int_\Omega \nabla u \cdot \nabla v \, d\boldsymbol{x} + \delta \int_\Omega uv \, d\boldsymbol{x} + \varepsilon \int_{\partial\Omega} uv \, d\sigma = \int_\Omega v \, d\boldsymbol{x} \tag{3.4}$$

or again written in the abstract form

$$a(u, v) = \ell(v) \tag{3.5}$$

where

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, d\boldsymbol{x} + \delta \int_\Omega uv \, d\boldsymbol{x} + \varepsilon \int_{\partial\Omega} uv \, d\sigma \tag{3.6}$$

and

$$\ell(v) = \int_\Omega v \, d\boldsymbol{x}. \tag{3.7}$$

Of course, we would like to find a unique solution of (3.6),(3.7) if it is possible. The Lax-Milgram Theorem (see for example [1]) guides us to find the proper space in which the solution u and the test functions v must live for a well-posed problem. Let us recall the theorem:

Theorem 3 (Lax-Milgram)**.** *Let $\mathcal{H}$ be a Hilbert space with scalar product $(.,.)$ and associated norm $||.||$. Let $\ell(.)$ be a linear form, continuous on $\mathcal{H}$ and $a(.,.)$ a bilinear form which is*

- *continuous on $\mathcal{H} \times \mathcal{H}$: $\exists c > 0$ /*

$$|a(u,v)| \leq C \, ||u||.||v|| \quad \forall u, v \in \mathcal{H}^2, \tag{3.8}$$

- *coercive (or $\mathcal{H}-elliptic$): $\exists a > 0$ /*

$$a(u,v) \geq \alpha \, ||u||^2. \tag{3.9}$$

Then there exist a unique u in $\mathcal{H}$ such that

$$a(u,v) = \ell(v) \quad \forall v \in \mathcal{H}. \tag{3.10}$$

In our example, the bilinear form a in (3.6) involves gradients of u and v. Thus we need to find a Hilbert space for which gradients exist in some sense. The Sobolev space $H^1(\Omega)$

$$H^1(\Omega) = \{v \in L^2(\Omega), \ \partial_i v \in L^2(\Omega), \ i = 1, \ldots, d\} \tag{3.11}$$

is proved to be a Hilbert space for the scalar product

$$\begin{aligned} (u,v)_{H^1} &= \int_\Omega uv \, d\boldsymbol{x} + \int_\Omega \partial_i u \partial_i v \, d\boldsymbol{x} && (3.12)\\ &= (u,v)_{L^2} + (\nabla u, \nabla v)_{L^2}. && (3.13) \end{aligned}$$

The H^1-scalar defines the associated H^1-norm

$$||u||_{H^1} = \sqrt{||u||^2_{L^2} + ||\nabla u||^2_{L^2}}. \tag{3.14}$$

Let us verify that all the assumptions of the Lax-Milgram Theorem are satisfied when $\mathcal{H} = H^1(\Omega)$. It is clear that the form a is bilinear. It is also continuous thanks to the Cauchy-Schwarz inequality:

$$\begin{aligned} |a(u,v)| &\leq ||\nabla u||_{L^2}.||\nabla v||_{L^2} + (\delta + \varepsilon) \, (||u||_{L^2}.||v||_{L^2}) \\ &\leq (1 + \delta + \varepsilon) \, ||u||_{H^1}.||v||_{H^1}. \end{aligned}$$

It is clearly H^1-elliptic:

$$\begin{aligned} a(u,u) &= \int_\Omega |\nabla u|^2 \, d\boldsymbol{x} + \delta \int_\Omega u^2 \, d\boldsymbol{x} + \int_{\partial\Omega} u^2 d\sigma \\ &\geq \min(1,\delta) \left(||\nabla u||_{L^2}^2 + ||u||_{L^2}^2 \right) \\ &= \min(1,\delta) \, ||u||_{H^1}^2. \end{aligned}$$

Finally, the form ℓ is clearly linear and continuous in H^1 thanks to Cauchy-Schwarz inequality:

$$|\ell(v)| \leq \int_\Omega 1.|v| \, d\boldsymbol{x} \leq |\Omega| ||v||_{L^2} \leq |\Omega| ||v||_{H^1}.$$

From the Lax-Milgram theorem , we have proved that the problem: *find* $u \in H^1(\Omega)$ *such that*

$$\int_\Omega \nabla u \cdot \nabla v \, d\boldsymbol{x} + \delta \int_\Omega uv \, d\boldsymbol{x} + \varepsilon \int_{\partial\Omega} uv \, d\sigma = \int_\Omega v \, d\boldsymbol{x} \quad \forall v \in H^1(\Omega) \tag{3.15}$$

has a unique solution in $H^1(\Omega)$. Actually, it defines what we call a weak solution of the initial PDE problem (3.1),(3.2).

3.1.1 Finite element method

The `freefem++` computational approach is based on (some convenient) discretization of the continuous problem (3.15). For that, one can use a conformal discrete Finite Element space V^h, meaning that V^h it embedded into the continuous space $H^1(\Omega)$. `freefem++` includes most of the common finite elements (P1, P2, P1-bubble, etc.). Let us for example consider a continuous piecewise polynomial of degree one P^1 Finite Element discretization on a triangulation $\mathscr{T}^h$ of Ω^h:

$$V^h = \left\{ v^h \in H^1(\Omega^h), \; v^h \in \mathscr{C}^0(\Omega^h), v^h_{|K} \in P^1(K) \, \forall K \in \mathscr{T}_h \right\}. \tag{3.16}$$

It is easy to check that

$$V^h \subset H^1(\Omega^h).$$

The discrete Finite Element problem then becomes: *find* $u^h \in V^h$ *such that*

$$\int_{\Omega^h} \nabla u^h \cdot \nabla v^h \, d\boldsymbol{x} + \delta \int_{\Omega^h} u^h v^h \, d\boldsymbol{x} + \varepsilon \int_{\partial\Omega^h} u^h v^h \, d\sigma = \int_{\Omega^h} v^h \, d\boldsymbol{x} \quad \forall v^h \in V^h. \tag{3.17}$$

Because the continuous bilinear form in (3.17) has still the ellipticity property in $\mathcal{H} = V^h$ provided with the H^1-scalar product, the Lax-Milgram theorem can be applied. This shows that the discrete problem (3.17) has a unique solution $u^h \in V^h$. Practically, the FE method requires the building of a large sparse matrix and the solution of a large sparse linear system. `freefem++` includes a large set of both direct and iterative solvers (LU, Choleski, UMFPACK, conjugate gradient, bicgstab, GMRES, etc...). See the user manual about more details on these solvers and the way to call them.

3.1.2 Practical implementation

`freefem++` language is a high-level interpreted language dedicated to FEM. There are primitives for geometry CAD definition, triangulation and mesh, finite element space definition, variable definition, variational problem definition, linear system solution and visualization facilities.

The following program written in `freefem++` language implements the standard Lagrangian P^1 FE and the variational formulation (3.17) on the unit disk. Important `freefem++` commands are `border` for border geometry definition, `mesh` for mesh generation, `fespace` for the definition of the Finite element space, `problem` for defining a variational problem and `solver` for its solution, and finally `plot` for visualization.

```
1  // Laplace.edp
2  // Laplace problem with robin boundary conditions
3  real epsilon = 1e-2;
4  border domega(t=0,2*pi){x=cos(t); y=sin(t);}
5  mesh Th = buildmesh (domega(100));
6  plot(Th, wait=1, ps="ThLaplace.eps");
7  fespace Vh(Th,P1);
8  Vh uh, vh;
9  func f=1;
10 real cpu=clock();
11 solve Poisson(uh, vh, solver=LU) = // defines the PDE
12   int2d(Th)(dx(uh)*dx(vh) + dy(uh)*dy(vh)) // bilinear part
13   + int1d(Th, domega)(epsilon*uh*vh)
14   - int2d(Th)( f*vh); // right hand side
15 cout << " CPU time = " << clock()-cpu << endl;
16 plot(uh, nbiso=50, fill=0, value=1, wait=1, ps="res.eps");
17 // Done !
```

3.2 Heat problem

`freefem++` is able to solve time-dependent problems. As example, let us consider the following heat problem. The boundary $\partial\Omega$ is partitioned into two boundaries Γ_1 and Γ_2:

$$\begin{aligned}
&\frac{\partial u}{\partial t} - \nabla\cdot(\kappa\nabla u) = 0 \quad \text{in } (0,T)\times\Omega,\\
&u(.,t=0) = \theta \quad \text{in } \Omega,\\
&\frac{\partial u}{\partial \boldsymbol{n}} = -1 \quad \text{on } (0,T)\times\Gamma_1,\\
&u = \theta \quad \text{on } [0,T)\times\Gamma_2.
\end{aligned}$$

As first step, consider a time-discretization of the equation, let say by the backward implicit Euler quadrature. The PDE in its semi-discrete form reads

$$\frac{u^{n+1}-u^n}{\Delta t^n} - \nabla\cdot(\kappa\nabla u^{n+1}) = 0. \tag{3.18}$$

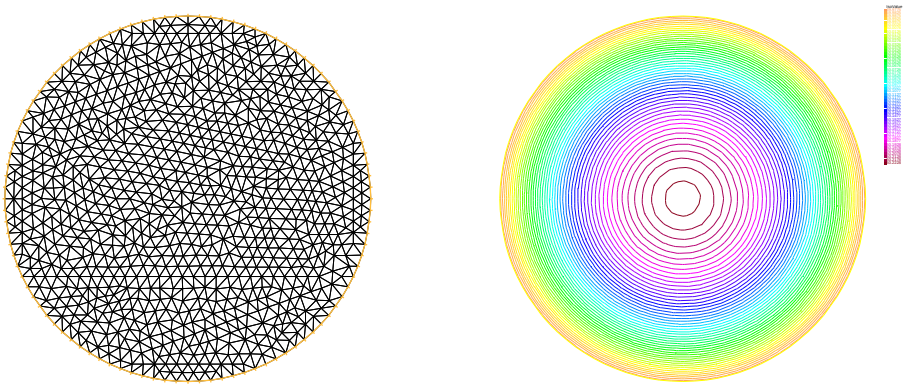


Figure 3.1: The problem (3.16) solved with `freefem++` on the unit disk. Automatic meshing and plot of the isocontours of the solution.

For spatial discretization, we shall use the Finite Element method. Multiplying (3.18) by a smooth test function v, integrating over Ω and applying Green's formula leads to

$$\frac{1}{\Delta t^n}\int_\Omega u^{n+1}v\,d\boldsymbol{x} + \int_\Omega \kappa\nabla u^{n+1}\cdot\nabla v\,d\boldsymbol{x} - \int_{\partial\Omega}\kappa\frac{\partial u^{n+1}}{\partial \boldsymbol{n}}v\,d\sigma = \frac{1}{\Delta t^n}\int_\Omega u^n v\,d\boldsymbol{x}. \tag{3.19}$$

The finite element space to consider for u here is the H^1 Sobolev space made of functions with trace equal to θ on Γ_2. Defining for any $g \in H^{1/2}(\Gamma_2)$

$$V_g = \left\{v \in H^1(\Omega), v = g \text{ on } \Gamma_2\right\}, \tag{3.20}$$

one considers a test function $v \in V_0$ and the variational formulation

$$\frac{1}{\Delta t^n}\int_\Omega u^{n+1}v\,d\boldsymbol{x} + \int_\Omega \kappa\nabla u^{n+1}\cdot\nabla v\,d\boldsymbol{x} + \int_{\Gamma_1}\kappa v\,d\sigma = \frac{1}{\Delta t^n}\int_\Omega u^n v\,d\boldsymbol{x}. \tag{3.21}$$

It is once again possible to apply the Lax-Milgram Theorem and show that the problem to find $u^{n+1} \in V_\theta$ such that (3.21) holds for all $v \in V_0$ has a unique solution.

The fully discretized problem is based on a discretization of the Sobolev spaces

$$V^h = \left\{v^h \in H^1(\Omega^h),\ v^h \in \mathscr{C}^0(\Omega^h), v^h_{|K} \in P^1(K)\ \forall K \in \mathscr{T}_h\right\}, \tag{3.22}$$

$$V^h_g = \left\{v^h \in V^h, v^h = g^h \text{ on } \Gamma_2^h\right\} \quad \text{for } g^h \in H^{1/2}(\Gamma_2^h), \tag{3.23}$$

and the variational formulation set into a finite dimensional linear space: *find* $u^h \in V^h_\theta$ such that

$$\frac{1}{\Delta t^n}\int_\Omega u^{h,n+1}v^h\,d\boldsymbol{x} + \int_\Omega \kappa\nabla u^{h,n+1}\cdot\nabla v\,d\boldsymbol{x} + \int_{\Gamma_1}\kappa v^h\,d\sigma = \frac{1}{\Delta t^n}\int_\Omega u^{h,n}v^h\,d\boldsymbol{x} \quad \forall v^h \in V^h_0. \tag{3.24}$$

3.2.1 Implementation in `freefem++`

```
// Heat.edp (freefem++)
// Linear Heat problem with inhomogeneous Dirichlet boundary
// conditions and Neuman nonzero flux boundary conditions.
//
real theta = 20; // initial and boundary temperature
real kappa = 0.1; // thermal conductivity
real dt = 0.5; // time step size
//
border Gamma1(t=0,2*pi){x=cos(t); y=sin(t);}
border Gamma2(t=0, 1){x=-0.5+t; y=0.3;}
mesh Th = buildmesh (Gamma1(100)+Gamma2(60));
plot(Th, wait=1, ps="Th.eps");
//
fespace Vh(Th,P1);
Vh uh, uold, vh;
// Initialization
uh = theta; uold = uh;
problem heatstep(uh, vh, solver=LU) = // defines the problem
    int2d(Th)(uh*vh/dt)
  - int2d(Th)(uold*vh/dt)
  + int2d(Th)(kappa*dx(uh)*dx(vh) + kappa*dy(uh)*dy(vh))
  + int1d(Th, Gamma1)(kappa*vh)
  + on(Gamma2, uh=theta); // Dirichlet BC
// Performs 6 time steps
for (int it=0;it<6;it++) {
// Update temperature field at previous time step
uold = uh;
// Then perform a time step
heatstep;
plot(uh, nbiso=50, fill=1, value=1, wait=1, ps="res"+it+".eps");

}
// Done !
```

3.3 A problem of thermal engineering

Let us here consider a more realistic problem of thermal engineering design. From two different materials - one is cheap, the other is expensive - we would like to build a composite material that provides good thermal resistance properties. If V is a volume occupied by a wall made of that composite material, V_c is the volume occupied by the cheap material and $V_e = V - V_c$ the volume occupied by the expensive one, the ratio

$$\mu = \frac{V_e}{V} = \frac{V_e}{V_c + V_e} \tag{3.25}$$

is fixed for economical purposes. We consider the geometry presented in figure 3.3. A rectangular room Ω_r has a size $L_x \times L_y$. At the center of the left wall, we have a radiator that locally keeps the temperature at T_r. We will denote that part of the wall Γ_r. The remainder of the left wall plus the upper and lower walls will be denoted Γ_a. We will suppose zero thermal flux boundary conditions on Γ_a. The domain Ω_w of the wall of the right part of the room is modeled. It will be made of the composite

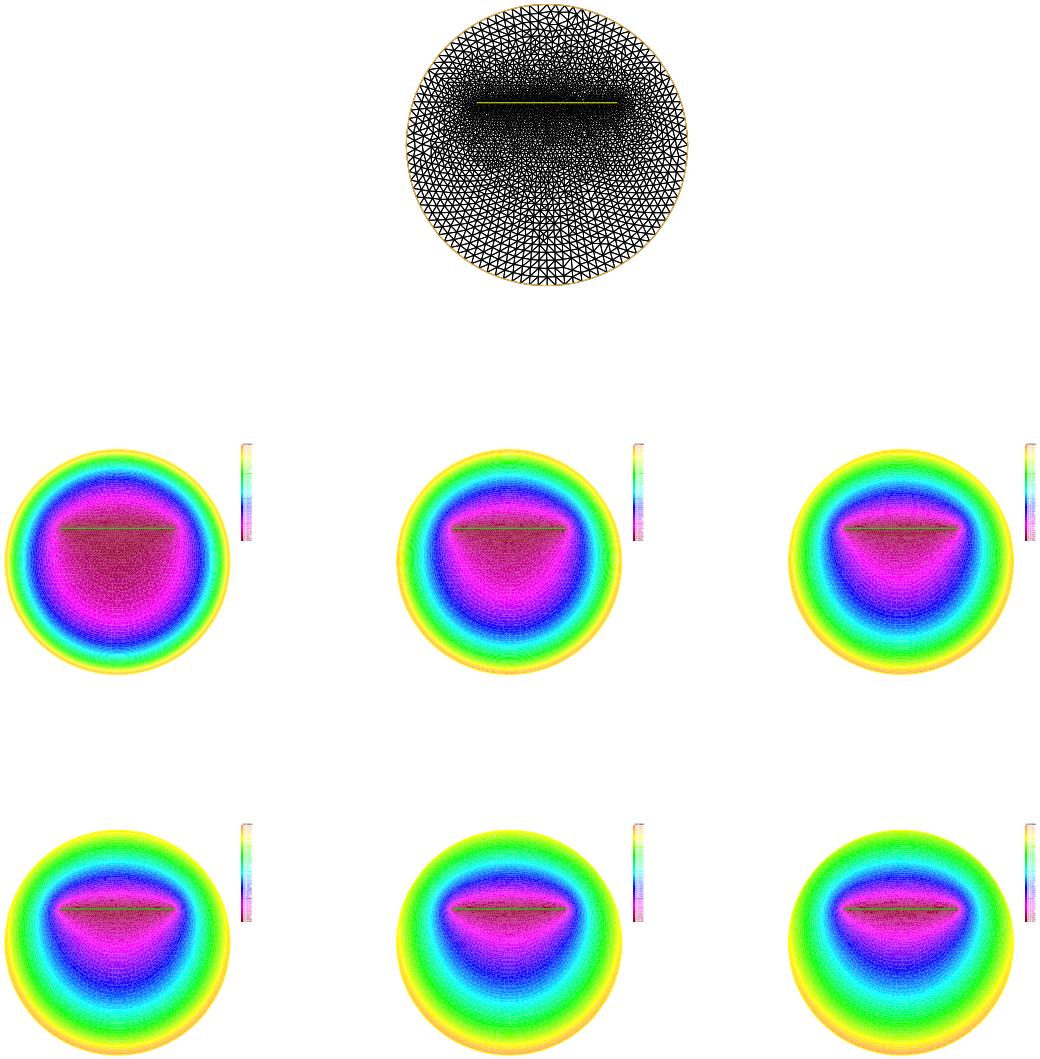


Figure 3.2: The problem (3.24) solved with `freefem++` . The mesh and the fields at the six first discrete time steps are plotted.

material. The width of the wall is named ℓ_x, so that $V = \ell_x L_y$. The right part of the composite wall denoted by Γ_{ext} is in contact with the exterior. The will denote T_{ext} the exterior temperature and will impose the Dirichlet boundary condition $T = T_{ext}$ on Γ_{ext}. Finally, we will respectively denote by κ_a, κ_c and κ_e the thermal conductivity coefficients of the ambient air, the cheap material and the expensive material. From the computation, we will extract performance indicators like:

1. the mean room temperature

$$J_1 = \bar{T} = \frac{1}{|\Omega_a|} \int_{\Omega_a} T \, d\boldsymbol{x}; \tag{3.26}$$

2. the minimal temperature in the room

$$J_2 = \min_{\boldsymbol{x} \in \Omega_a} T(\boldsymbol{x}); \tag{3.27}$$

3. the standard deviation of temperature:

$$J_3 = \sqrt{\frac{1}{\Omega_a} \int_{\Omega_a} \left(T(\boldsymbol{x}) - \bar{T}\right)^2 \, d\boldsymbol{x}}. \tag{3.28}$$

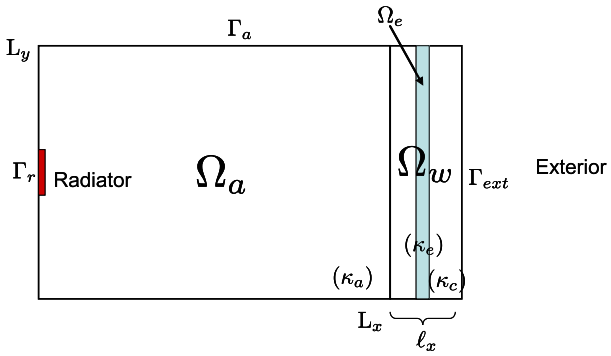


Figure 3.3: A problem of thermal engineering design. The room domain Ω_a, the wall domain Ω_w made of the expensive material domain Ω_e and the cheap material domain Ω_c.

Of course they are many ways to define the composite material: it can be organized by layers with one or several layers of expensive material. It can be designed as inclusion of spherical expensive material particulates, organized deterministically or randomly, etc. A complex engineering design would to be optimize the placement of the expensive material marked by the domain Ω_e, for example

$$\min_{\Omega_e} J_3(\Omega_e) \tag{3.29}$$

subject to

$$\frac{|\Omega_e|}{|\Omega_w|} = \mu. \tag{3.30}$$

This problem belongs to the general mathematical problem of shape optimization.

Let us denote Ω the whole interior domain of figure 3.3. The stationary thermal problem to consider

is

$$-\nabla \cdot (\kappa(.)\nabla T) = 0 \quad \text{in } \Omega, \tag{3.31}$$

$$\frac{\partial T}{\partial \boldsymbol{n}} = 0 \quad \text{on } \Gamma_a = \partial\Omega \backslash (\Gamma_r \cup \Gamma_{ext}), \tag{3.32}$$

$$T = T_r \quad \text{on } \Gamma_r, \tag{3.33}$$

$$T = T_{ext} \quad \text{on } \Gamma_{ext} \tag{3.34}$$

where $\kappa(.)$ is the piecewise constant thermal conductivity defined from κ_a, κ_c and κ_e:

$$\kappa(\boldsymbol{x}) = \kappa_a \, 1_{x\in\Omega_a}(\boldsymbol{x}) + \kappa_c \, 1_{x\in\Omega_c}(\boldsymbol{x}) + \kappa_e \, 1_{x\in\Omega_e}(\boldsymbol{x}). \tag{3.35}$$

```
// ThermalDesign.edp
// Thermal design engineeering problem
// Wall composite material made of two homogeneous materials,
// on is cheap, the other one is expensive.
real Lx = 4, Ly=5, lr=1, lx=0.3;
real Tr=20, Text = -5;
real hy = 0.1;
real hx = 0.1;
real mu = 0.2;
real kappaa = 10;
real kappac = 1;
real kappae = 0.008;
//
real V = lx*Ly;
real Ve = mu * V;
real xem = Lx + hx;
real deltaxe = Ve / (Ly-2*hy);
//
border c1(t=0,Lx){x=t; y=0;}
border c1w(t=Lx,Lx+lx){x=t; y=0;}
border cext(t=0,Ly){x=Lx+lx; y=t;}
border c3w(t=Lx+lx,Lx){x=t; y=Ly;}
border c3(t=Lx,0){x=t; y=Ly;}
border c4(t=Ly,Ly/2+lr/2){x=0; y=t;}
border cr(t=Ly/2+lr/2,Ly/2-lr/2){x=0; y=t;}
border c5(t=Ly/2-lr/2,0){x=0; y=t;}
border c0(t=0,Ly){x=Lx; y=t;}
//
border ce1(t=xem, xem+deltaxe){x=t; y=hy;}
border ce2(t=hy,Ly-hy){x=xem+deltaxe; y=t;}
border ce3(t=xem+deltaxe, xem){x=t; y=Ly-hy;}
border ce4(t=Ly-hy,hy){x=xem; y=t;}
//
mesh Th = buildmesh(c1(30)+c1w(10)+cext(100)
+c3w(10)+c3(30)+c4(15)+cr(15)+c5(15)+c0(100)
+ce1(15)+ce2(500)+ce3(15)+ce4(500));
plot(Th, ps="mesh.eps", wait=1);

```

```
39 mesh Th2=buildmesh( c1(30)+c0(100)+c3(30)+c4(15)
40 +cr(15)+c5(15) );
41 //
42 fespace Vh(Th, P1);
43 Vh kappa, th, vh;
44 real region1 = Th(Lx/2,Ly/2).region;
45 real region2 = Th(xem+deltaxe/2, Ly/2).region;
46 kappa = kappac + (kappaa-kappac)*(region==region1)
47   +(kappae-kappac)*(region==region2);
48   plot(kappa, nbiso=60, fill=1, value=1);
49 //
50 problem thermal(th, vh) =
51   int2d(Th)(kappa*dx(th)*dx(vh)+kappa*dy(th)*dy(vh))
52   +on(cext, th=Text)
53   +on(cr, th=Tr);
54 //
55 thermal;
56 real[int] colorhsv=[ // color hsv model
57 4./6., 1 , 0.5, // dark blue
58 4./6., 1 , 1, // blue
59 5./6., 1 , 1, // magenta
60 1, 1. , 1, // red
61 1, 0.5 , 1 // light red
62 ];
63 real[int] viso(26);
64 for (int i=0; i<viso.n; i++)
65   viso[i] = -5+i;
66 plot(th,viso=viso(0:viso.n-1), value=1, fill=1,
67   ps="tfield.eps");
68 // Performance indicators
69 fespace Vh2(Th2, P1);
70 Vh2 th2 = th;
71 real J1, J2, J3;
72 // Mean value
73 J1 = int2d(Th2)(th2)/Th2.area;
74 J2 = th2[].min;
75 J3 = sqrt(int2d(Th2)((th2-J1)^2)/Th2.area);
76 // Min value
77 // Standard Deviation
78 cout << "Mean temperature = " << J1 << endl;
79 cout << "Min temperature = " << J2 << endl;
80 cout << "Standard deviation = " << J3 << endl;
81 cout << "Done." << endl;
```

```
Mean temperature = 16.9214
Min temperature = 13.5943
Standard deviation = 0.98602
Done.
times: compile 0.187s, execution 1.282s,  mpirank:0
```

Figure 3.4: A snapshot of the standard output of the `freefem++` program `ThermalDesign.edp`.

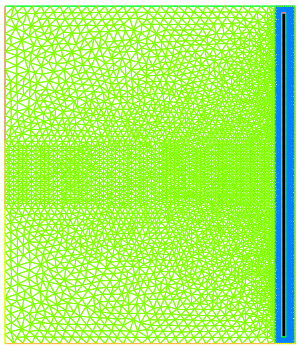


Figure 3.5: Mesh computed by `ThermalDesign.edp` for the thermal design problem. The composite wall here is organized in layers. One can notice the strong variation of diameter of the triangles due to the room/wall aspect ratio involving different spatial scales.

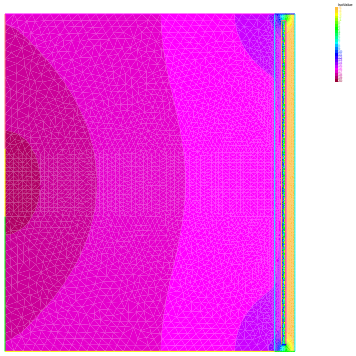


Figure 3.6: Temperature field computed by the `freefem++` program `ThermalDesign.edp`. Each band of color represents a temperature range of 1 degree.

Chapter 4

The Method of characteristics

4.1 Mathematical setting

In dimension $d = 1, 2, 3$ and for a domain $\Omega \subset \mathbb{R}^d$, the inhomogeneous convection equation is written

$$\partial_t q + \boldsymbol{u} \cdot \nabla q = f \quad \text{ on } \Omega \times (0, T) \tag{4.1}$$

where $\boldsymbol{u}(\boldsymbol{x}, t) \in \mathbb{R}^d$. Recall that the equation (4.1) can be discretized as

$$\frac{Dq}{Dt} = f$$

i.e.

$$\frac{dq}{dt}(\boldsymbol{X}(t), t) = f(\boldsymbol{X}(t), t), \ \frac{d\boldsymbol{X}}{dt}(t) = \boldsymbol{u}(\boldsymbol{X}(t), t)$$

where D is the particle derivative (also called the total derivative operator). So a good time discretization scheme is one step of backward convection by the so-called method of Characteristics:

$$\frac{q^{n+1}(\boldsymbol{x}) - q^n(\boldsymbol{X}^n(\boldsymbol{x}))}{\Delta t^n} = f^n(\boldsymbol{x}) \tag{4.2}$$

where $\boldsymbol{X}^n(\boldsymbol{x})$ is an approximation of the solution at time t^n ($t^{n+1} = t^n + \Delta t^n$) of the ordinary differential equation

$$\frac{d\boldsymbol{X}}{dt}(t) = \boldsymbol{u}^n(\boldsymbol{X}(t)), \ \boldsymbol{X}(t^{n+1}) = \boldsymbol{x}$$

where $\boldsymbol{u}^n(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}, t^n)$. Because, by Taylor's expansion, we have

$$\begin{aligned} q^n(\boldsymbol{X}(t^n)) &= q^n(\boldsymbol{X}(t^{n+1})) - \Delta t^n \sum_{i=1}^{d} \frac{\partial q}{\partial x_i}(\boldsymbol{X}(t^{n+1})) \frac{\partial X_i}{\partial t}(t^{n+1}) + o(\Delta t^n) \\ & q^n(\boldsymbol{x}) - \Delta t^n \, \boldsymbol{u}^n(\boldsymbol{x}) \cdot \nabla q^n(\boldsymbol{x}) + o(\Delta t^n) \end{aligned} \tag{4.3}$$

where $X_i(t)$ are the i-th components of $\boldsymbol{X}(t)$, $q^n(\boldsymbol{x}) = q(\boldsymbol{x}, t^n)$ and $\boldsymbol{x} = \boldsymbol{X}(t^{n+1})$. From (4.3), it follows that

$$q^n(\boldsymbol{X}^n(\boldsymbol{x})) = q^n(\boldsymbol{x}) - \Delta t^n \, \boldsymbol{u}^n(\boldsymbol{x}) \cdot \nabla q^n(\boldsymbol{x}) + o(\Delta t^n). \tag{4.4}$$

Also if we apply Taylor's expansion for $t \mapsto q^n(\boldsymbol{x} - \boldsymbol{u}^n(\boldsymbol{x})t)$, $0 \le t \le \Delta t^n$, then

$$q^n(\boldsymbol{x} - \boldsymbol{u}^n(\boldsymbol{x})\Delta t^n) = q^n(\boldsymbol{x}) - \Delta t^n\, \boldsymbol{u}^n(\boldsymbol{x}) \cdot \nabla q^n(\boldsymbol{x}) + o(\Delta t^n).$$

Denoting the `freefem++` function

$$\texttt{convect}(\boldsymbol{u}^n, -\Delta t^n, q^n) \approx q^n(\boldsymbol{x} - \boldsymbol{u}^n \Delta t^n)$$

we can get the approximation

$$q^n(\boldsymbol{X}^n(\boldsymbol{x})) \approx \texttt{convect}(\boldsymbol{u}^n, -\Delta t^n, q^n)$$

by

$$\boldsymbol{X}^n \approx \boldsymbol{x} \mapsto \boldsymbol{x} - \Delta t^n \boldsymbol{u}^n(\boldsymbol{x}).$$

4.1.1 `freefem++` source code of the pure transport problem

```
// Convection.edp (Freefem++)
real Lx = 6;
real Ly = 4;
real dt = 3;
real dtsnap=100, time=0, tsnap=dt;
int itmax=300;
//
real[int] A(2), B(2), C(2), D(2);
real[int] E(2), F(2), G(2), H(2);
real[int] I(2), J(2), K(2), L(2);
real[int] O(2), P(2), Q(2), R(2);
real[int] itplot(itmax), masse(itmax);
A = [0,0];  B=[Lx,0];
C= [Lx,Ly]; D=[0,Ly];
E=[Lx/2,Ly/4]; F=[Lx/2,3*Ly/4];
G=[Lx/6,Ly/4]; H=[5*Lx/6,Ly/4];
I=[Lx/6,3*Ly/4]; J=[5*Lx/6,3*Ly/4];
O=[0, Ly/2]; P=[Lx/3,Ly/2];
Q=[2*Lx/3, Ly/2]; R=[Lx,Ly/2];
border c1(t=0,1){x=(1-t)*A[0]+t*B[0]; y=(1-t)*A[1]+t*B[1];}
border c2(t=0,1){x=(1-t)*B[0]+t*C[0]; y=(1-t)*B[1]+t*C[1];}
border c3(t=0,1){x=(1-t)*C[0]+t*D[0]; y=(1-t)*C[1]+t*D[1];}
border c4(t=0,1){x=(1-t)*D[0]+t*A[0]; y=(1-t)*D[1]+t*A[1];}
border c5(t=0,1){x=(1-t)*E[0]+t*F[0]; y=(1-t)*E[1]+t*F[1];}
border c6(t=0,1){x=(1-t)*G[0]+t*H[0]; y=(1-t)*G[1]+t*H[1];}
border c7(t=0,1){x=(1-t)*I[0]+t*J[0]; y=(1-t)*I[1]+t*J[1];}
border c8(t=0,1){x=(1-t)*O[0]+t*P[0]; y=(1-t)*O[1]+t*P[1];}
border c9(t=0,1){x=(1-t)*Q[0]+t*R[0]; y=(1-t)*Q[1]+t*R[1];}
int nn=20;
mesh Th = buildmesh( c1(8*nn)+c2(6*nn)+c3(8*nn)+c4(6*nn)
+c5(4*nn)+c6(5*nn)+c7(5*nn)+c8(3*nn)+c9(3*nn) );
plot (Th, ps="mesh.eps");
//
func fy=x-Lx/2;
```

```
fespace Uh(Th, P1b);
fespace Vh(Th, P1);
Uh u1, u2, u1h, u2h;
Vh p, ph, q;
//
problem Stokes([u1, u2, p],[u1h, u2h,ph]) =
  int2d(Th)(dx(u1)*dx(u1h)+dy(u1)*dy(u1h))
  +int2d(Th)(dx(u2)*dx(u2h)+dy(u2)*dy(u2h))
  +int2d(Th)(dx(p)*u1h+dy(p)*u2h)
  +int2d(Th)(dx(u1)*ph+dy(u2)*ph)
  -int2d(Th)(fy*u2h)
  +on(c1,c2,c3,c4,c5,c6,c7,c8,c9, u1=0, u2=0);
//
Stokes; plot([u1, u2], ps="velocity.eps");
//
q = (sqrt((x-Lx/2)^2+(y-Ly/8)^2)<0.2);
real masse0 = int2d(Th)(q);
//
plot(q, nbiso=40, fill=1);
//
for (int it=0; it<itmax; it++){
  itplot[it] = it;
  q = convect([u1,u2], -dt, q);
  masse[it] = int2d(Th)(q);
  time += dt;
  plot(q, nbiso=40, fill=1);
  cout << "Masse0 = " << masse0 << "Masse(t) = "
  << masse[it] << endl;
  if (time >= tsnap) {
    tsnap += dtsnap;
    plot(q, nbiso=60, fill=1, ps="q_time="+time+".eps");
  }
}
plot([itplot, masse], ps="histomasse.eps", value=1);
ofstream of("histomasse.txt");
for (int it=0; it<itmax; it++) {
  of << itplot[it] << " " << masse[it] << endl;
}
```

4.1.2 Numerical Results

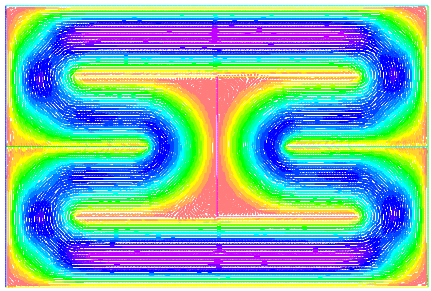


Figure 4.1: Independent stationary velocity field computed with the Stokes equations

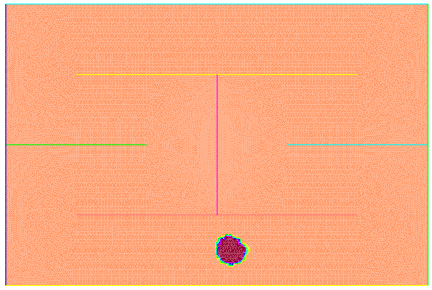


Figure 4.2: Numerical solution of the pure convection problem. Numerical solution at time $t = 3$.

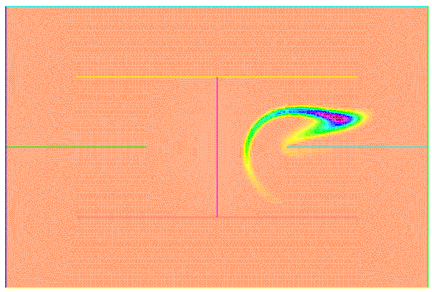


Figure 4.3: Numerical solution of the pure convection problem. Numerical solution at time $t = 105$.

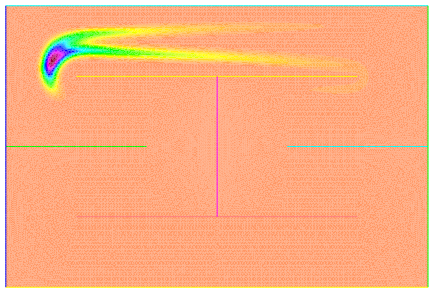


Figure 4.4: Numerical solution of the pure convection problem. Numerical solution at time $t = 204$.

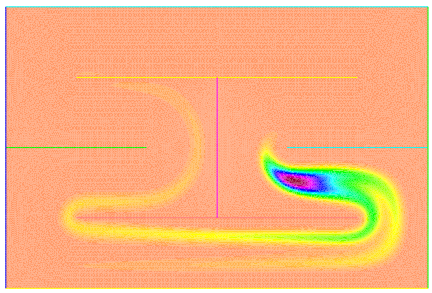


Figure 4.5: Numerical solution of the pure convection problem. Numerical solution at time $t = 405$.

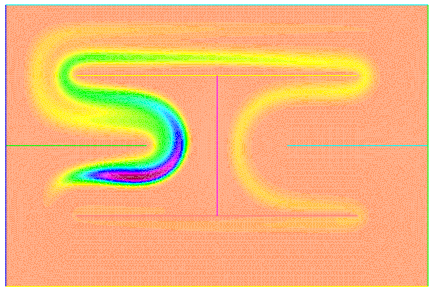


Figure 4.6: Numerical solution of the pure convection problem. Numerical solution at time $t = 603$.

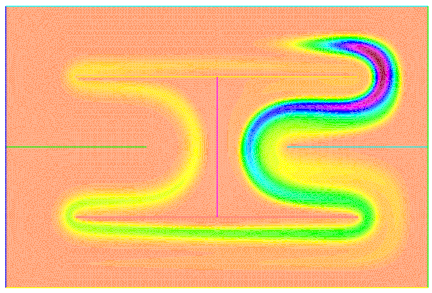


Figure 4.7: Numerical solution of the pure convection problem. Numerical solution at time $t = 804$.

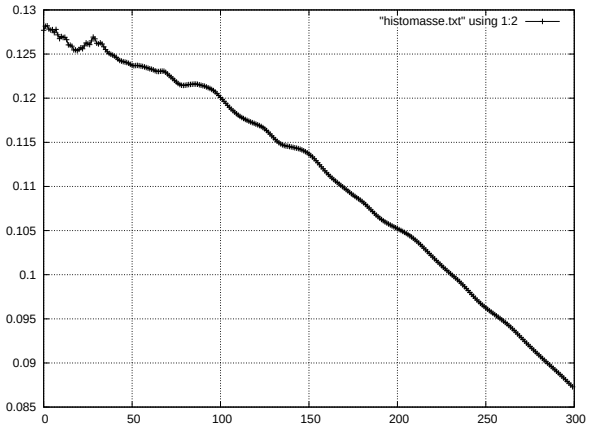


Figure 4.8: History of the total mass of the solution $t \mapsto ||q(.,t)||_{L^1}$. One can observe an increasing error of mass conservation.

Chapter 5

Stokes equations and Navier-Stokes equations

5.1 Setting of the equations

Let us consider a Newtonian fluid with density ρ and velocity $\boldsymbol{u}$. For an incompressible fluid (constant density), the continuity equation of mass conservation

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{5.1}$$

simply writes

$$\nabla \cdot \boldsymbol{u} = 0. \tag{5.2}$$

Let p be the pressure of the fluid, μ the dynamic viscosity and ρ a volume external force. The balance equation of momentum

$$\partial_t(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) - \nabla \cdot (\mu \nabla \boldsymbol{u}) + \nabla p = \rho \boldsymbol{f} \tag{5.3}$$

also simplifies as

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}. \tag{5.4}$$

where $\nu = \mu/\rho$ denotes the static viscosity. The two equations (5.2) and (5.3) form the well-known Navier-Stokes equations. Those equations are nonlinear because of the inertial term $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$. The existence and uniqueness of solutions of these equations in the 3D case in still an open problem. It is known that the structure of the solutions can be quite complex especially for small viscosities ν where turbulence occurs.

Dimensionless Navier-Stokes equations make appear the Reynolds number

$$Re = \frac{U_0 \, L_0}{\nu} \tag{5.5}$$

where U_0 and L_0 are respectively velocity and length characteristic scales. The dimensionless Navier-Stokes equations then write

$$\nabla \cdot \boldsymbol{u} = 0, \ \partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \frac{1}{Re} \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}. \tag{5.6}$$

Small Reynolds numbers lead to a laminar flow dominated by “Stokes effects" whereas large Reynolds numbers lead to a turbulent flow. For intermediate Reynolds, the flow is said to be in transition regime with appearance of unsteady instabilities. When the flow is dominated by viscous effects (small Reynolds numbers), the initial term $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$ is rather small compred to the viscous term. Thus, the Navier-Stokes equations can be approximated by the Stokes equations

$$\nabla \cdot \boldsymbol{u} = 0, \tag{5.7}$$

$$\partial_t \boldsymbol{u} - \frac{1}{Re}\Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}. \tag{5.8}$$

When the flow is closed to a steady state, the unsteady Stokes equation can be approximated by so-called stationary Stokes equations

$$\nabla \cdot \boldsymbol{u} = 0, \tag{5.9}$$

$$-\frac{1}{Re}\Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}. \tag{5.10}$$

5.2 Analysis of the stationary Stokes problem

Let us consider a smooth spatial domain $\Omega \in \mathbb{R}^2$. For reasons that will appear later, we shall consider an approximate model of the Stokes equations, namely the pseudo-compressible approximation where a pressure term is added to the continuity equation:

$$-\nu\Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}, \tag{5.11}$$

$$\nabla \cdot \boldsymbol{u} + \varepsilon\, p = 0. \tag{5.12}$$

As an example, let us consider a boundary $\partial\Omega$ splitted up into three borders $\partial\Omega = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ with distinct boundary conditions. On Γ_1, we will consider Dirichlet boundary condition for the velocity:

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \Gamma_1 \tag{5.13}$$

(for example a given velocity profile for inlet flow). On Γ_2, we will consider wall boundary no-slip conditions:

$$\boldsymbol{u} = 0 \quad \text{on } \Gamma_2. \tag{5.14}$$

On Γ_3, let us consider homogeneous Neumann conditions on velocity and a zero pressure condition

$$\frac{\partial \boldsymbol{u}}{\partial n} = 0,\ p = 0 \quad \text{on } \Gamma_3. \tag{5.15}$$

(usual for outlet boundary conditions for example) Let us now write the variational formulation of the problem. The first equation (5.11) also writes component-by-component

$$-\nu\Delta u_j + \partial_j p = f_j,\ j = 1, 2.$$

By multiplying by a smooth test function v_j and integrating over Ω, we get

$$-\nu \int_\Omega \Delta u_j v_j \, d\boldsymbol{x} + \int_\Omega \partial_j p\, v_j \, d\boldsymbol{x} = \int_\Omega f_j v_j \, d\boldsymbol{x}.$$

Applying Green's formula gives

$$-\nu \int_{\partial\Omega} \frac{\partial u_j}{\partial n} v_j \, d\gamma + \nu \int_\Omega \nabla u_j \cdot \nabla v_j \, d\boldsymbol{x} + \int_{\partial\Omega} p n_j \, v_j \, d\gamma - \int_\Omega p \partial_j v_j \, d\boldsymbol{x} = \int_\Omega f_j v_j \, d\boldsymbol{x}.$$

This can be written in vector form

$$-\nu \int_{\partial\Omega} \frac{\partial \boldsymbol{u}}{\partial n} \cdot \boldsymbol{v} \, d\gamma + \nu \int_\Omega \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{v} \, d\boldsymbol{x} + \int_{\partial\Omega} p \, \boldsymbol{v} \cdot \boldsymbol{n} \, d\gamma - \int_\Omega p \, \nabla \cdot \boldsymbol{v} \, d\boldsymbol{x} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, d\boldsymbol{x} \tag{5.16}$$

with $\boldsymbol{v} = (v_1, v_2)$. We also multiply equation (5.12) by a smooth test function q and integrate over Ω:

$$\int_\Omega \nabla \cdot \boldsymbol{u} \, q \, d\boldsymbol{x} + \varepsilon \int_\Omega p \, q \, d\boldsymbol{x} = 0. \tag{5.17}$$

Now, we have to find the proper functional spaces for both the unknowns $(\boldsymbol{u}, p)$ and the test functions $(\boldsymbol{v}, q)$ according to the boundary conditions (5.13)-(5.15). For any $\varphi \in H^{1/2}(\Gamma_1)$, let us consider the product space

$$V_\varphi = \left\{ (\boldsymbol{v}, q) \in [H^1(\Omega)]^2 \times L^2(\Omega), \ \boldsymbol{v} = \varphi \text{ on } \Gamma_1, \ \boldsymbol{v} = 0 \ \text{ on } \Gamma_2 \right\}. \tag{5.18}$$

It is clear that $V = V_0$ is a Hilbert space for the scalar product

$$((\boldsymbol{u}, p), (\boldsymbol{v}, q))_V = (\boldsymbol{u}, \boldsymbol{v})_{H^1(\Omega)} + (p, q)_{L^2(\Omega)} \tag{5.19}$$

and the associated norm

$$||(\boldsymbol{u}, p)||_V = \left(||\boldsymbol{u}||^2_{H^1(\Omega)} + ||p||^2_{L^2(\Omega)} \right)^{1/2}.$$

By choosing functions $(\boldsymbol{u}, p) \in V_g$ and $(\boldsymbol{v}, q) \in V$, all the terms in (5.18),(5.19) have a sense. From boundary conditions (5.13)-(5.15), the variational formulation (5.16) reduces to

$$\nu \int_\Omega \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{v} \, d\boldsymbol{x} - \int_\Omega p \, \nabla \cdot \boldsymbol{v} \, d\boldsymbol{x} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, d\boldsymbol{x}. \tag{5.20}$$

So the problem is to find a solution $(\boldsymbol{u}, p) \in V_g$ such that (5.20) and (5.17) hold for all $(\boldsymbol{v}, q) \in V$. This is also equivalent to find $(\boldsymbol{u}, p) \in V_g$ such that

$$\nu \int_\Omega \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{v} \, d\boldsymbol{x} - \int_\Omega p \, \nabla \cdot \boldsymbol{v} \, d\boldsymbol{x} + \int_\Omega q \, \nabla \cdot \boldsymbol{u} \, d\boldsymbol{x} + \varepsilon \int_\Omega p \, q \, d\boldsymbol{x} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, d\boldsymbol{x} \tag{5.21}$$

for all $(\boldsymbol{v}, q) \in V$. Equation (5.21) is in the form

$$a\left((\boldsymbol{u}, p), (\boldsymbol{v}, q)\right) = \ell\left(\boldsymbol{v}, q\right) \tag{5.22}$$

with

$$a\left((\boldsymbol{u}, p), (\boldsymbol{v}, q)\right) = \nu \int_\Omega \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{v} \, d\boldsymbol{x} - \int_\Omega p \, \nabla \cdot \boldsymbol{v} \, d\boldsymbol{x} + \int_\Omega q \, \nabla \cdot \boldsymbol{u} \, d\boldsymbol{x} + \varepsilon \int_\Omega p \, q \, d\boldsymbol{x} \tag{5.23}$$

and

$$\ell\left(\boldsymbol{v}, q\right) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, d\boldsymbol{x}. \tag{5.24}$$

We are going to verify that the assumptions on the Lax-Milgram theorem are satisfied. First, remember that V his a Hilbert space with scalar product (5.19). Clearly, a is a bilinear form in $V \times V$ and ℓ is a linear form in V. Let us show the continuity property with respect to the V-norm. First by the Cauchy-Schwarz inqequality we have

$$\begin{aligned} |\ell(\boldsymbol{v}, q)| &\leq ||\boldsymbol{f}||_{[L^2(\Omega)]^2} \, ||\boldsymbol{v}||_{[L^2(\Omega)]^2} \\ &\leq ||\boldsymbol{f}||_{[L^2(\Omega)]^2} \, ||\boldsymbol{v}||_{[H^1(\Omega)]^2} \\ &\leq ||\boldsymbol{f}||_{[L^2(\Omega)]^2} \, ||(\boldsymbol{v}, q)||_V . \end{aligned}$$

Secondly, we have also

$$\begin{aligned} |a\left((\boldsymbol{u}, p), (\boldsymbol{v}, q)\right)| &\leq \nu ||\boldsymbol{u}||_{H^1(\Omega)} \, ||\boldsymbol{v}||_{H^1(\Omega)} + ||p||_{L^2(\Omega)} ||\nabla \cdot \boldsymbol{v}||_{L^2(\Omega)} + ||q||_{L^2(\Omega)} ||\nabla \cdot \boldsymbol{u}||_{L^2(\Omega)} \\ &\quad + \varepsilon \, ||p||_{L^2(\Omega)} \, ||q||_{L^2(\Omega)} \\ &\leq (\nu + 2 + \varepsilon) \, ||(\boldsymbol{u}, p)||_V \, ||(\boldsymbol{v}, q)||_V \end{aligned}$$

so that a is a bilinear form, continuous in $V \times V$. The last property to verify is the V-ellipticity of the bilinear form a. From (5.23) we have

$$\begin{aligned} a\left((\boldsymbol{u}, p), (\boldsymbol{u}, p)\right) &= \int_\Omega |\nabla \boldsymbol{u}|^2 \, d\boldsymbol{x} + \varepsilon \int_\Omega p^2 \, d\boldsymbol{x} \\ &= |\boldsymbol{u}|^2_{H^1(\Omega)} + \varepsilon \, ||p||^2_{L^2(\Omega)} \\ &\geq \min(1, \varepsilon) ||(\boldsymbol{u}, p)||_V . \end{aligned}$$

This last inequality shows the V-ellipticity property as soon as $\varepsilon > 0$. Thus, the Lax-Milgram theorem states that the solution $(\boldsymbol{u}, p)$ of the problem (5.22) exists and is unique. For a Finite Element method which is conformal in V, we also have this result of existence and uniqueness, showing that the discrete problem

$$a((\boldsymbol{u}^h, p^h), (\boldsymbol{v}^h, q^h)) = \ell(\boldsymbol{v}^h, q^h) \quad \forall (\boldsymbol{v}^h, q^h) \in V^h \subset V \tag{5.25}$$

is well-posed. Remark that the pseudo-compressibility assumption is important to get the V-ellipticity property. Under the true incompressibility hypothesis ($\varepsilon = 0$), more theoretical developments and a deeper analysis are required. The existence and uniqueness comes from a property of ellipticity called the inf-sup condition or also referred to as LLB (Ladyzenskaya-Babuska-Brezzi) condition (see[1]). From the discrete point of view, stable Finite Element method for the true incompressible Stokes equations are method that satisfy a discrete version of the inf-sup condition (the so-called discrete inf-sup condition). Discrete inf-sup conditions are proved for example for P1-bubble/P1 approximations (P1-bubble in velocity and P1 in pressure) or P2/P1 approximation (P2 in velocity and P1 in pressure). The P1/P1 approximation violates the discrete inf-sup condition. In that case, some parasite modes appear in the discrete solution. This topic is beyond the scope of this course. The interested reader can refer to the important literature on this topic (see[1, **?**]).

5.3 Numerical method for the Navier-Stokes equations

Let us consider now the time-dependent Navier-Stokes equations with the pseudo compressibility approximation:

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}, \tag{5.26}$$

$$\nabla \cdot \boldsymbol{u} + \varepsilon\, p = 0. \tag{5.27}$$

The system of equations becomes nonlinear due to the convection term. The simplest way to approximate these equations is to use the methods of characteristics for the convection term and standard Finite Elements for the other term. A semi-discrete time discretization of (5.26),(5.27) gives

$$\frac{\boldsymbol{u}^{n+1}(\boldsymbol{x}) - \boldsymbol{u}^n \circ X^n(\boldsymbol{x})}{\Delta t} - \nu \Delta \boldsymbol{u}^{n+1}(\boldsymbol{x}) + \nabla p^{n+1}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}), \tag{5.28}$$

$$\nabla \cdot \boldsymbol{u}^{n+1}(\boldsymbol{x}) + \varepsilon\, p^{n+1}(\boldsymbol{x}) = 0. \tag{5.29}$$

For a given field $\boldsymbol{u}^n$, considering $(\boldsymbol{u}^{n+1}, p^{n+1})$ as the unknowns, the system (5.28),(5.29) is seen as the Stokes equations. Then we can apply the results obtained for the Stokes problem. Considering the same boundary conditions as the previous section, the variational problem of (5.28),(5.29) is to find $(\boldsymbol{u}^{n+1}, p^{n+1}) \in V_g$ such that

$$\int_\Omega \frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n \circ X^n}{\Delta t} \cdot \boldsymbol{v}\, d\boldsymbol{x} + \nu \int_\Omega \nabla \boldsymbol{u}^{n+1} \cdot \nabla \boldsymbol{v}\, d\boldsymbol{x} - \int_\Omega p^{n+1} \nabla \cdot \boldsymbol{v}\, d\boldsymbol{x} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\boldsymbol{x}, \tag{5.30}$$

$$\int_\Omega \nabla \cdot \boldsymbol{u}^{n+1}\, q\, d\boldsymbol{x} + \varepsilon \int_\Omega p^{n+1}\, q\, d\boldsymbol{x} = 0. \tag{5.31}$$

for all $(\boldsymbol{v}, q) \in V$. As exercise, we reader will show the continuity and ellipticity constants of the underlying bilinear and linear forms.

5.4 Numerical experiments

5.4.1 `freefem++` source code

```
//
real Re = 600.0;
real nu = 1.0/Re;
real Lx = 12;
real Ly = 5;
real dt = 0.5;
border c1(t=0,1){x=t*Lx; y=0;}
border c2(t=0,1){x=Lx; y=t*Ly;}
border c3(t=1,0){x=t*Lx; y=Ly;}
border c4(t=1,0){x=0; y=t*Ly;}
border c5(t=2*pi,0){x=4+0.2*cos(t); y=Ly/2+0.2*sin(t);}
border c6(t=0,1){x=5+Lx/2*t; y=Ly/2+0.4;}
border c7(t=0,1){x=5+Lx/2*t; y=Ly/2-0.4;}
//
```

```
15 mesh Th = buildmesh(c1(80)+c2(40)+c3(80)+c4(20)
16         +c5(60)+c6(100)+c7(100));
17 plot(Th);
18 //
19 fespace Uh(Th, P2);
20 fespace Vh(Th, P1);
21 Uh u, v, uh, vh, uold, vold;
22 Vh p, ph;
23 Vh uplot, vplot, vort;
24 //
25 // Velocity field is initialized by steady state solution
26 //
27 real eps = 1e-10;
28 problem steadystokes([u,v,p], [uh,vh,ph]) =
29   int2d(Th)( nu*dx(u)*dx(uh) + nu*dy(u)*dy(uh) )
30   +int2d(Th)(nu*dx(v)*dx(vh) + nu*dy(v)*dy(vh) )
31   -int2d(Th)(p*dx(uh))
32   -int2d(Th)(p*dy(vh))
33   -int1d(Th, c2)(nu*dx(u)*N.x*uh+nu*dy(u)*N.y*uh)
34   -int1d(Th, c2)(nu*dx(v)*N.x*vh+nu*dy(v)*N.y*vh)
35   +int2d(Th)(dx(u)*ph + dy(v)*ph)
36   +int2d(Th)(eps*p*ph)
37   +on(c1, c3, c5, u=0, v=0)
38   +on(c4, u=4.0 * y/Ly * (1-y/Ly), v=0);
39 //
40 steadystokes;
41 uplot = u;
42 vplot = v;
43 plot(Th, [uplot,vplot], nbiso=40, value=1);
44 uold = u;
45 vold = v;
46 //
47 // Now go for unsteady Navier-Stokes equations.
48 //
49 int it = 0;
50 problem navierstokes([u,v,p], [uh,vh,ph], init=it,
51   solver=sparsesolver) =
52   int2d(Th)(u*uh/dt)
53   -int2d(Th)(convect([uold,vold], -dt, uold)*uh/dt)
54   +int2d(Th)(v*vh/dt)
55   -int2d(Th)(convect([uold,vold], -dt, vold)*vh/dt)
56   +int2d(Th)( nu*dx(u)*dx(uh) + nu*dy(u)*dy(uh) )
57   +int2d(Th)(nu*dx(v)*dx(vh) + nu*dy(v)*dy(vh) )
58   -int2d(Th)(p*dx(uh))
59   -int2d(Th)(p*dy(vh))
60   -int1d(Th, c2)(nu*dx(u)*N.x*uh+nu*dy(u)*N.y*uh)
61   -int1d(Th, c2)(nu*dx(v)*N.x*vh+nu*dy(v)*N.y*vh)
62   +int2d(Th)(dx(u)*ph + dy(v)*ph)
63   +int2d(Th)(eps*p*ph)
64   +on(c1, c3, c5, u=0, v=0)
65   +on(c4, u=4.0 * y/Ly * (1-y/Ly), v=0);
66 //
```

```
67 for (it=0; it<20; it++) {
68   for (int subit=0; subit<5; subit++) {
69     navierstokes;
70     // Th = adaptmesh(Th, [u,v]); u = u; v = v;
71     uold = u;
72     vold = v;
73   }
74   uplot = u;
75   vplot = v;
76   plot(Th, [uplot, vplot], nbiso=60, ps="u_ns_it="+it+".eps");
77   vort = dy(u)-dx(v);
78   plot(vort, nbiso=60, fill=0, ps="vort_ns_it="+it+".eps");
79 }
```

5.4.2 Numerical results

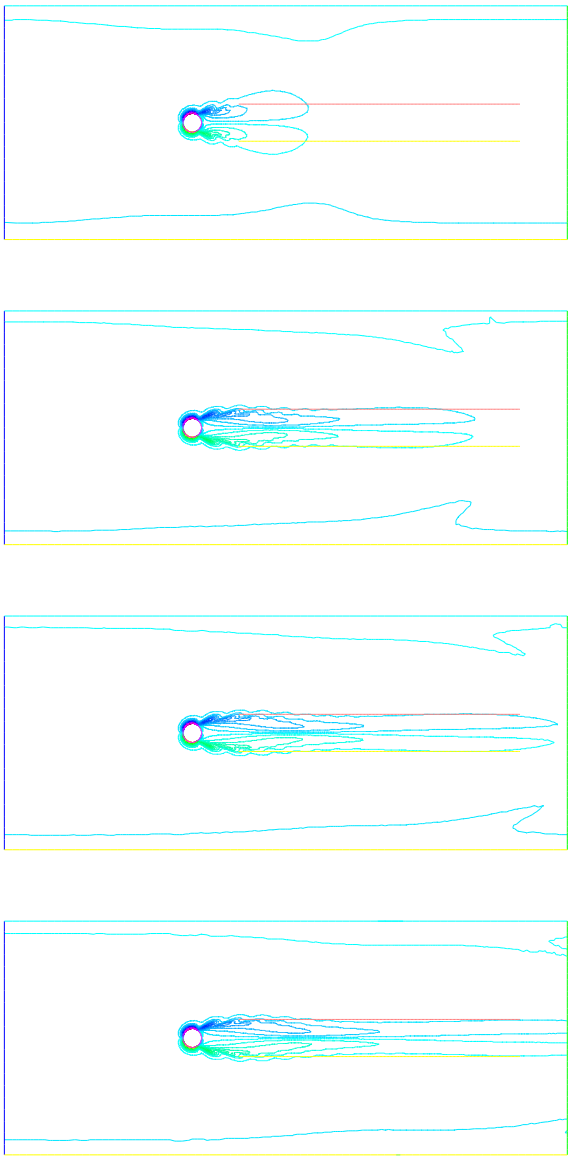


Figure 5.1: Vorticity contours during simulation.

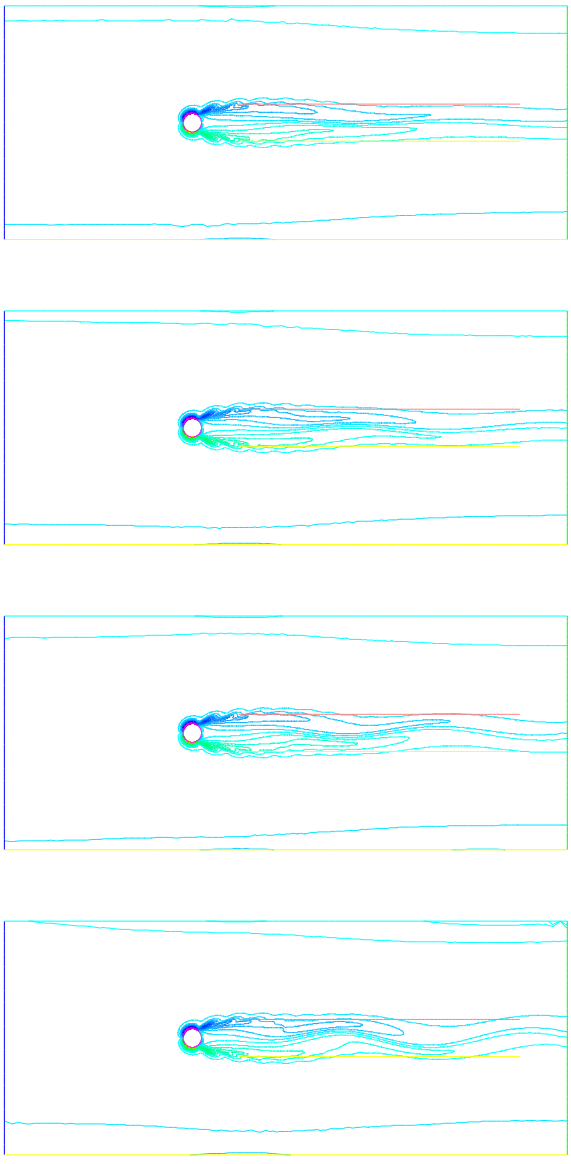


Figure 5.2: Vorticity contours during simulation. Von Karman instabilities develop.

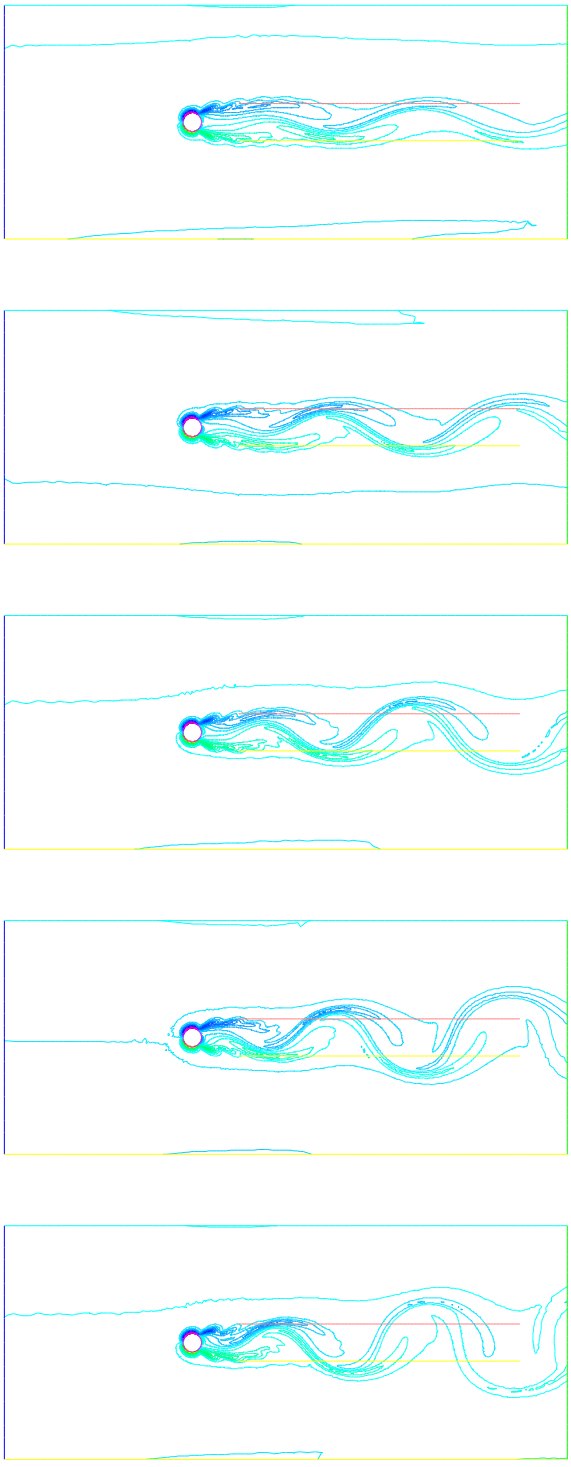


Figure 5.3: Vorticity contours during simulation. Von Karman instabilities develop.

Chapter 6

Fractional step methods

Fractional step methods also known as operator splitting methods are time advance schemes where partial differential equations are splitted up into different simpler (well-posed) partial differential equations. The different PDE problems are solved sequentially. Those are interesting computational approaches especially when the Physics is complex. When the whole Physics is a combination of different effects (convection, diffusion, reaction, exchanges, etc.), fractional steps method solve each physical effect independently and sequentially. Fractional step methods can also be seen as a way to couple component codes where each code solves a particular Physics.

6.1 Introduction

Let us consider an ordinary differential system of equations

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}_1(\boldsymbol{u}) + \boldsymbol{f}_2(\boldsymbol{u}) \tag{6.1}$$

where $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ both are continuously differentiable functions. With an initial data

$$\boldsymbol{u}(0) = \boldsymbol{u}^0 \in \mathbb{R}^p \tag{6.2}$$

the problem has a unique maximal solution in a time interval I that includes 0. Let Δt be a small time step such that $[0, \Delta t] \subset I$. Then a Taylor expansion gives

$$\begin{aligned} \boldsymbol{u}(\Delta t) &= \boldsymbol{u}^0 + \Delta t \, \frac{d\boldsymbol{u}}{dt}(\boldsymbol{u}^0) + o(\Delta t) \\ &= \boldsymbol{u}^0 + \Delta t \, \left(\boldsymbol{f}_1(\boldsymbol{u}^0) + \boldsymbol{f}_2(\boldsymbol{u}^0)\right) + o(\Delta t). \end{aligned} \tag{6.3}$$

A fractional step method for (6.1),(6.2) can be the following one:

1. First solve the differential problem

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}_1(\boldsymbol{u}), \tag{6.4}$$

$$\boldsymbol{u}(0) = \boldsymbol{u}^0 \tag{6.5}$$

   over a time step. Let $\hat{\boldsymbol{u}}^0 = \boldsymbol{u}(\Delta t)$.

2. Then solve the differential problem

$$\frac{d\hat{\boldsymbol{u}}}{dt} = \boldsymbol{f}_2(\hat{\boldsymbol{u}}), \tag{6.6}$$

$$\hat{\boldsymbol{u}}(0) = \hat{\boldsymbol{u}}^0 \tag{6.7}$$

over a time step Δt.

Then it is easy to check that the solution $\hat{\boldsymbol{u}}(\Delta t)$ is an approximation of order 1 in Δt of the true solution $u(\Delta t)$ of (6.1),(6.2) at time Δt. Indeed, from Taylor expansions we have

$$\begin{aligned}
\hat{\boldsymbol{u}}(\Delta t) &= \hat{\boldsymbol{u}}^0 + \Delta t \frac{d\hat{\boldsymbol{u}}}{dt}(\hat{\boldsymbol{u}}^0) + o(\Delta t) \\
&= \boldsymbol{u}^0 + \Delta t\, \boldsymbol{f}_2(\hat{\boldsymbol{u}}^0) + o(\Delta t) \\
&= \boldsymbol{u}^0 + \Delta t\, \boldsymbol{f}_1(\boldsymbol{u}^0) + \Delta t\, \boldsymbol{f}_2(\boldsymbol{u}^0 + \Delta t\, \boldsymbol{f}_1(\boldsymbol{u}^0) + o(\Delta t)) + o(\Delta t) \\
&= \boldsymbol{u}^0 + \Delta t\, \left(\boldsymbol{f}_1(\boldsymbol{u}^0) + \boldsymbol{f}_2(\boldsymbol{u}^0)\right) + o(\Delta t).
\end{aligned} \tag{6.8}$$

Thus Taylor expansions (6.3) and (6.8) are identical and $\hat{\boldsymbol{u}}(\Delta t)$ is a first order approximation in Δt of $\boldsymbol{u}(\Delta t)$.

As we will see, there are higher order fractional steps methods. We have also to check that the splitting approach is stable in time.

6.2 Continuous analysis, case of a linear system

Consider now the linear problem

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{B}\boldsymbol{u}, \tag{6.9}$$

$$\boldsymbol{u}(0) = \boldsymbol{u}^0 \tag{6.10}$$

for some square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The analytical solution of (6.9) and (6.10) is

$$\boldsymbol{u}(t) = \exp\left((\boldsymbol{A} + \boldsymbol{B})t\right)\, \boldsymbol{u}^0,\ t \geq 0. \tag{6.11}$$

Proceeding in two steps by splitting up the problem into

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{A}\boldsymbol{u}, \tag{6.12}$$

$$\boldsymbol{u}(0) = \boldsymbol{u}^0 \tag{6.13}$$

whose solution is $\boldsymbol{u}(t) = \exp(\boldsymbol{A}t)\, \boldsymbol{u}^0$, and

$$\frac{d\hat{\boldsymbol{u}}}{dt} = \boldsymbol{B}\hat{\boldsymbol{u}}, \tag{6.14}$$

$$\hat{\boldsymbol{u}}(0) = \hat{\boldsymbol{u}}^0 = \exp(\boldsymbol{A}t)\, \boldsymbol{u}^0, \tag{6.15}$$

the approximate solution is

$$\hat{\boldsymbol{u}}(t) = \exp(\boldsymbol{B}t)\exp(\boldsymbol{A}t)\,\boldsymbol{u}^0 \tag{6.16}$$

which is in general different from (6.11). At a small instant $t = \Delta t$, we have by from Taylor expansions

$$\begin{aligned}
\boldsymbol{u}(\Delta t) &= \exp((\boldsymbol{A}+\boldsymbol{B})\Delta t)\,\boldsymbol{u}^0 \\
&= \left(\boldsymbol{I} + \Delta t(\boldsymbol{A}+\boldsymbol{B}) + \frac{\Delta t^2}{2}(\boldsymbol{A}+\boldsymbol{B})^2\right)\boldsymbol{u}^0 + o(\Delta t^2) \\
&= \left(\boldsymbol{I} + \Delta t(\boldsymbol{A}+\boldsymbol{B}) + \frac{\Delta t^2}{2}(\boldsymbol{A}^2+\boldsymbol{A}\boldsymbol{B}+\boldsymbol{B}\boldsymbol{A}+\boldsymbol{B}^2)\right)\boldsymbol{u}^0 + o(\Delta t^2)
\end{aligned} \tag{6.17}$$

on one side and

$$\begin{aligned}
\hat{\boldsymbol{u}}(\Delta t) &= \exp(\boldsymbol{B}\Delta t)\exp(\boldsymbol{A}\Delta t)\,\boldsymbol{u}^0 \\
&= \left(\boldsymbol{I} + \Delta t\boldsymbol{B} + \frac{\Delta t^2}{2}\boldsymbol{B}^2\right)\left(\boldsymbol{I} + \Delta t\boldsymbol{A} + \frac{\Delta t^2}{2}\boldsymbol{A}^2\right)\boldsymbol{u}^0 + o(\Delta t^2) \\
&= \left(\boldsymbol{I} + \Delta t(\boldsymbol{A}+\boldsymbol{B}) + \frac{\Delta t^2}{2}(\boldsymbol{A}^2+2\boldsymbol{B}\boldsymbol{A}+\boldsymbol{B}^2)\right)\boldsymbol{u}^0 + o(\Delta t^2)
\end{aligned} \tag{6.18}$$

on the other side. From (6.17) and (6.18), one can conclude that the fractional step method generally leads to a first order approximate solution. In the particular case where matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ commute, i.e.

$$\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}\boldsymbol{A}, \tag{6.19}$$

then the approximation is second order accurate.

6.3 Strang second-order symmetric splitting

We shall prove that the following symmetric three-step fractional step method provides second-order accuracy. By denoting

$$\mathcal{P}_A^{\Delta t}\,\boldsymbol{u}^0 \tag{6.20}$$

the solution of

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{A}\,\boldsymbol{u},\ \boldsymbol{u}(0) = \boldsymbol{u}^0, \tag{6.21}$$

at time Δt and

$$\mathcal{P}_B^{\Delta t}\boldsymbol{u}^0 \tag{6.22}$$

the solution of

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{B}\,\boldsymbol{u},\ \boldsymbol{u}(0) = \boldsymbol{u}^0, \tag{6.23}$$

at time Δt, the Strang second-order symmetric splitting consists in approximating the exact solution $\boldsymbol{u}(\Delta t)$ by

$$\hat{\boldsymbol{u}}(\Delta t) = \mathcal{P}_A^{\Delta t/2}\,\mathcal{P}_B^{\Delta t}\,\mathcal{P}_A^{\Delta t/2}\,\boldsymbol{u}^0. \tag{6.24}$$

Let us show that the approximation (6.24) is second order accurate in time. We have also

$$\hat{\boldsymbol{u}}(\Delta t) = \exp(\boldsymbol{A}\frac{\Delta t}{2})\exp(\boldsymbol{B}\Delta t)\exp(\boldsymbol{A}\frac{\Delta t}{2})\,\boldsymbol{u}^0. \tag{6.25}$$

By a Taylor expansion of order in Δt in (6.24), we get

$$\begin{aligned}
\hat{\boldsymbol{u}}(\Delta t) &= \left(\boldsymbol{I}+\frac{\Delta t}{2}\boldsymbol{A}+\frac{\Delta t^2}{8}\boldsymbol{A}^2\right)\left(\boldsymbol{I}+\Delta t\boldsymbol{B}+\frac{\Delta t^2}{2}\boldsymbol{B}^2\right)\left(\boldsymbol{I}+\frac{\Delta t}{2}\boldsymbol{A}+\frac{\Delta t^2}{8}\boldsymbol{A}^2\right)\boldsymbol{u}^0+o(\Delta t^2)\\
&= \left(\boldsymbol{I}+\Delta t\boldsymbol{A}+\Delta t\boldsymbol{B}+\frac{\Delta t}{2}(\boldsymbol{A}^2+\boldsymbol{A}\boldsymbol{B}+\boldsymbol{B}\boldsymbol{A}+\boldsymbol{B}^2)\right)\boldsymbol{u}^0+o(\Delta t^2)\\
&= \exp((\boldsymbol{A}+\boldsymbol{B})\Delta t)+o(\Delta t^2)
\end{aligned}$$

which proves the assertion.

6.4 Discrete time advance schemes

Let us now consider discrete times advances schemes combined with an operator splitting approach.

6.4.1 First order scheme

Consider again the nonlinear differential problem (6.1),(6.2). Let $\boldsymbol{u}^n$ be an approximation of order 1 of $\boldsymbol{u}(t^n)$, Δt a time step and $t^{n+1} = t^n + \Delta t$ the next discrete time. Using for example forward Euler schemes for the integration of each step, we get the following scheme corresponding to the time iteration n:

1. Compute first
$$\hat{\boldsymbol{u}}^{n+1} = \boldsymbol{u}^n + \Delta t\,\boldsymbol{f}_1(\boldsymbol{u}^n); \tag{6.26}$$
2. Then compute
$$\boldsymbol{u}^{n+1} = \hat{\boldsymbol{u}}^{n+1} + \Delta t\,\boldsymbol{f}_2(\hat{\boldsymbol{u}}^{n+1}). \tag{6.27}$$

It is an easy matter of fact to show that the fractional step metho (6.26),(6.27) is first order accurate. By using (6.26) into (6.27) we have

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t\,\boldsymbol{f}_1(\boldsymbol{u}^n) + \Delta t\,\boldsymbol{f}_2\left(\boldsymbol{u}^n + \Delta t\,\boldsymbol{f}_1(\boldsymbol{u}^n)\right). \tag{6.28}$$

The consistency error of expression (6.28) is

$$\begin{aligned}
\varepsilon^n &= \frac{\boldsymbol{u}(t^{n+1})-\boldsymbol{u}(t^n)}{\Delta t} - \boldsymbol{f}_1(\boldsymbol{u}(t^n)) - \boldsymbol{f}_2\left(\boldsymbol{u}(t^n)+\Delta t\,\boldsymbol{f}_1(\boldsymbol{u}(t^n))\right)\\
&= \boldsymbol{f}_1(\boldsymbol{u}(t^n)) + \boldsymbol{f}_2(\boldsymbol{u}(t^n)) + o(1) - \boldsymbol{f}_1(\boldsymbol{u}(t^n)) - \boldsymbol{f}_2(\boldsymbol{u}(t^n)) - \Delta t\,\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{u}}(\boldsymbol{u}(t^n))\boldsymbol{f}_1(\boldsymbol{u}(t^n)) + o(\Delta t)\\
&= o(1).
\end{aligned}$$

6.4.2 Second order schemes

For many applications where dynamical effects or transient phases are important, first order schemes are not sufficiently accurate to compute the dynamics. Therefore second order fractional step schemes are needed. From the second order symmetric Strang splitting scheme, let us define a discrete time advance scheme.

Each step of the operator splitting has to solved with a second order scheme too in order to get global second order accuracy.

Recall first that the following two-step Runge-Kutta RK2 time-advance scheme (also known as the Heun scheme) is second order accurate: for the solution of $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u})$ between t^n and t^{n+1}, it is written

$$\hat{\boldsymbol{u}}^{n+1} = \boldsymbol{u}^n + \Delta t \boldsymbol{f}(\boldsymbol{u}^n), \tag{6.29}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \frac{\Delta t}{2}\left(\boldsymbol{f}(\boldsymbol{u}^n) + \boldsymbol{f}(\hat{\boldsymbol{u}}^{n+1})\right). \tag{6.30}$$

The consistency error is

$$\begin{aligned}
\varepsilon^n &= \frac{\boldsymbol{u}(t^{n+1}) - \boldsymbol{u}(t^n)}{\Delta t} - \frac{1}{2}\left[\boldsymbol{f}(\boldsymbol{u}(t^n)) + \Delta t \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \boldsymbol{f}(\boldsymbol{u}(t^n))(\mathbf{u}(\mathbf{t^n})) + \mathbf{f}(\mathbf{u}(\mathbf{t^n}))\right] + o(\Delta t) \\
&= \boldsymbol{f}(\boldsymbol{u}(t^n)) + \frac{\Delta t}{2}\frac{d^2\boldsymbol{u}}{dt^2}(\boldsymbol{u}(t^n)) - \frac{1}{2}\left[\boldsymbol{f}(\boldsymbol{u}(t^n)) + \Delta t \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \boldsymbol{f}(\boldsymbol{u}(t^n))(\mathbf{u}(\mathbf{t^n})) + \mathbf{f}(\mathbf{u}(\mathbf{t^n}))\right] + o(\Delta t) \\
&= o(\Delta t)
\end{aligned}$$

because

$$\frac{d^2\boldsymbol{u}}{dt^2} = \frac{d}{dt}\boldsymbol{f}(\boldsymbol{u}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{dt} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\boldsymbol{u}(t))\boldsymbol{f}(\boldsymbol{u}(t)).$$

Let us now define the discrete Strang fractional step scheme. For that, let us denote $\mathcal{P}_1^{\Delta t/2}$ the time propagator operator over a time step $\frac{\Delta t}{2}$ of the solution of the problem

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}_1(\boldsymbol{u}(t)),$$

$$\boldsymbol{u}(0) = \boldsymbol{u}^0$$

i.e.

$$\mathcal{P}_1^{\Delta t/2}\,\boldsymbol{u}^0 = \boldsymbol{u}(\frac{\Delta t}{2}) \tag{6.31}$$

and $\mathcal{P}_2^{\Delta t}$ the time propagator operator over a time step Δt of the solution of the problem

$$\frac{d\boldsymbol{v}}{dt} = \boldsymbol{f}_2(\boldsymbol{v}(t)),$$

$$\boldsymbol{v}(0) = \boldsymbol{u}^0$$

i.e.

$$\mathcal{P}_2^{\Delta t}\,\boldsymbol{u}^0 = \boldsymbol{v}(\Delta t). \tag{6.32}$$

Then the discrete Strang splitting scheme using the Heun scheme as second time advance scheme is defined by the three following steps:

1. Approximation of the state $\mathcal{P}_1^{\Delta t/2}\,\boldsymbol{u}^n$:

$$\hat{\boldsymbol{u}}^{1,n+1} = \boldsymbol{u}^n + \frac{\Delta t}{2}\boldsymbol{f}_1(\boldsymbol{u}^n), \tag{6.33}$$

$$\boldsymbol{u}^{1,n+1} = \boldsymbol{u}^n + \frac{\Delta t}{4}\left(\boldsymbol{f}_1(\boldsymbol{u}^n) + \boldsymbol{f}_1(\hat{\boldsymbol{u}}^{1,n+1})\right); \tag{6.34}$$

2. Approximation of the state $\mathcal{P}_2^{\Delta t}\,\boldsymbol{u}^{1,n+1}$:

$$\hat{\boldsymbol{u}}^{2,n+1} = \boldsymbol{u}^{1,n+1} + \Delta t\boldsymbol{f}_2(\boldsymbol{u}^{1,n+1}), \tag{6.35}$$

$$\boldsymbol{u}^{2,n+1} = \boldsymbol{u}^{1,n+1} + \frac{\Delta t}{2}\left(\boldsymbol{f}_2(\boldsymbol{u}^{1,n+1}) + \boldsymbol{f}_2(\hat{\boldsymbol{u}}^{2,n+1})\right); \tag{6.36}$$

3. Approximation of the state $\mathcal{P}_1^{\Delta t/2}\,\boldsymbol{u}^{2,n+1}$:

$$\hat{\boldsymbol{u}}^{n+1} = \boldsymbol{u}^{2,n+1} + \frac{\Delta t}{2}\boldsymbol{f}_1(\boldsymbol{u}^{2,n+1}), \tag{6.37}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{2,n+1} + \frac{\Delta t}{4}\left(\boldsymbol{f}_1(\boldsymbol{u}^{2,n+1}) + \boldsymbol{f}_1(\hat{\boldsymbol{u}}^{n+1})\right). \tag{6.38}$$

As exercise, the proof of second order consistency is let to the reader.

6.5 Chorin-Temam fractional step method for the Navier-Stokes equations

In this section, we will consider true incompressible time-dependent Navier-Stokes equations.

With standard discretizations, both velocity $\boldsymbol{u}$ and pressure p variables are solved simultaneously. It is said that the velocity and pressure are coupled in the solution. From the numerical point of view, this leads of course to a large system to solve at each time step. In three dimensions for example, considering P1 approximations and N degrees of freedom, the size of the system to solve is $4N$.

Because the solution of 4 linear systems of size N has lower complexity than the solution of a linear system of size $4N$, techniques of variable decoupling have been investigated. The so-called Chorin-Temam fractional step method is based on a splitting of the Navier-Stokes equation into two parts. Each time step is made of two substeps:

1. Step 1. First solve the equation by "forgetting" the pressure term over a time interval Δt:

$$\frac{\boldsymbol{u}^\star(\boldsymbol{x}) - \boldsymbol{u}^n \circ X^n(\boldsymbol{x})}{\Delta t} - \nu\Delta\boldsymbol{u}^{n+1}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}). \tag{6.39}$$

2. Step 2. Solve the remaining par of the system (pressure term) over a time interval Δt:

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^\star}{\Delta t} + \nabla p^{n+1} = 0 \tag{6.40}$$

in order to project the velocity $\boldsymbol{u}^{n+1}$ on a divergence-free velocity field. We want $\nabla \cdot \boldsymbol{u}^{n+1} = 0$. By taking the divergence of the equation (6.40), one gets the Poisson pressure equation

$$-\Delta p^{n+1} = -\frac{\nabla \cdot \boldsymbol{u}^{\star}}{\Delta t}. \tag{6.41}$$

Once the Poisson equation is solve, the velocity field is updated according to the rule

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{\star} - \Delta t\, \nabla p^{n+1}. \tag{6.42}$$

A remarkable property of Chorin-Temam's method is that the solution $\boldsymbol{u}^{n+1}$ at time t^{n+1} is consistent with the Navier-Stokes solution with order of accuracy $O(\Delta t)$.

By using this fractional step approach, the velocity and pressure fields are indeed weakly coupled. During step 1, only the velocity is changed. More over, because the Laplacian is a "diagonal" operator, each component of the velocity can be solved independently. Step 1 requires the solution of d independent convection-diffusion problems. In step 2, a Poisson problem on the pressure alone has to be solved.

6.5.1 Boundary conditions

We have also to take care of the boundary conditions in this fractional step approach. For step 1, one can use the prescribed boundary conditions on the velocity, i.e.

$$\boldsymbol{u}^{\star} = \boldsymbol{g} \text{ on } \Gamma_1,\ \boldsymbol{u}^{\star} = 0 \text{ on } \Gamma_2,\ \frac{\partial \boldsymbol{u}^{\star}}{\partial n} = 0 \text{ on } \Gamma_3.$$

For the second step, it is a little more tricky because we have to add artificial boundary conditions on for the unknown pressure to get a boundary value problem. Velocity boundary conditions are already taken into account in the first step, so we would like that the velocity correction step does not affect the boundary velocity too much. From (6.42), we have at any boundary

$$\frac{\partial \boldsymbol{u}^{n+1}}{\partial t} = \frac{\partial \boldsymbol{u}^{\star}}{\partial n} - \Delta t\, \frac{\partial p^{n+1}}{\partial n} = 0.$$

So the natural boundary condition for the Poisson problem is

$$\frac{\partial p^{n+1}}{\partial n} = 0 \text{ on } \partial\Omega. \tag{6.43}$$

Unfortunately, there is no control on the tangential derivative of the pressure on the boundary. The Chorin-Temam fractional step method is known to have a lack of accuracy near boundaries. The second difficulty is that the solution of a Poisson problem with purely homogeneous Neumann boundary conditions is not unique and defined up to a constant. A perturbed "regularized" Poisson problem

$$-\Delta p + \varepsilon\, p = -\frac{\nabla \cdot \boldsymbol{u}^{\star}}{\Delta t} \tag{6.44}$$

should be considered numerically in order to get the ellipticity property.

Chapter 7

Case study. Population dynamics and migration flux analysis

Population dynamics are typically convection-reaction-diffusion equations which model the migration process of two kinds of population with interaction between both populations. Such models are able to explain the migration dynamics of predators and preys ecosystems or for instance the human migration flux into a large city.

7.1 Lokta-Volterra equations

Suppose that two animal species are living in the same territory, a family of predators (in number u in the sequel) and a family of preys (in number v). To write a model, some assumptions are done. It is supposed that preys without predators are proliferating, predators without preys vanish, and that preys are kept by predators at a fixed rate. In this section, it is also assumed that the spatial distribution of the populations is homogeneous, so that there is no spatial effect.

With these assumptions, the dynamical system is clearly

$$\frac{du}{dt} = a\,uv - b\,u, \tag{7.1}$$

$$\frac{dv}{dt} = -c\,uv + d\,v, \tag{7.2}$$

where the positive constant a corresponds to the growth rate of predators by prey capture and the positive constant b is the natural death rate. Also, constants c and d respectively represent the loss rate of preys by capture and their natural proliferation rate.

By a time scale change, it is possible to write the system (7.1),(7.2) by the normalized one

$$\frac{du}{dt} = \alpha\,u(v-1), \tag{7.3}$$

$$\frac{dv}{dt} = v(1-u), \tag{7.4}$$

for a constant $\alpha \geq 0$. To (7.3),(7.4), an initial condition is added

$$u(0) = u^0,\ v(0) = v^0. \tag{7.5}$$

The following Scilab code below is able to plot different trajectory solutions for different initial conditions. The figure 7.1 shows a graphical result of the Scilab code. One can observe that solutions are periodical in time. They create a closed trajectory solution in the state space (u, v), $u, v \geq 0$.

```
// odelokta.sce (Scilab file) - florian de vuyst
//
// This file is a modification of the initial file
//"ode_lotka.dem.sce", part of the Scilab project.
//
// Sharks and sardins: Lotka-Volterra ODE
//


text = ["Lotka-Volterra:"; ..
       "du/dt = u(v-1)/2"; ..
       "dv/dt = v(1-u)"; ..
       ""; ..
       "A trajectory is plotted by clicking on the"; ..
       " LEFT button of the mouse."; ..
       " The trajectory is updated as you move the mouse."; ..
       " To fix the trajectory, click again on the LEFT button."; ..
       "You can start over by clicking on the LEFT button again"; ..
       " or stop everything by clicking on the RIGHT button." ];

x_message(text);
my_handle = scf(100001);
clf(my_handle,"reset");
demo_viewCode("odelokta.sce");

function yprim=f(t,y)
  yprim=[alpha*y(1)*(y(2)-1) y(2)*(1-y(1))]
endfunction

alpha = 0.5;
xmin = 0; xmax = 4.0; ymin = 0; ymax = 6.0;
fx = xmin:0.5:xmax; fy = ymin:0.5:ymax;
fchamp(f,1,fx,fy);
xlabel('u (normalized nb. of predators)','fontsize',3)
ylabel('v (normalized nb. of preys)','fontsize',3)
a=gca();a.margins(3)=0.2;
title([_("Lokta-Volterra vector field")
      "du/dt = u(v-1)/2";
      "dv/dt = v(1-u)"],'fontsize',3)

t0     = 0;          tmax   = 20;
t      = t0:0.05:tmax;
oldx0  = 10*xmax;    oldy0  = 10*ymax;
dx     = 0.1;        dy     = 0.1;
rtol   = 0.0001;     atol   = rtol;

while (%t)
```

```
48   [b,x0,y0]=xclick();
49   if or(b==[2 5 -1000]) then break end;
50   if or(b==[0 3]) & xmin<x0 & x0<xmax & ymin<y0 & y0<ymax then
51     sol=ode([x0;y0],t0,t,rtol,atol,f);
52     xpoly(sol(1,:)',sol(2,:)');
53     p=gce();p.thickness=2;p.foreground=5;
54     rep=[x0,y0,-1];
55     while rep(3)==-1 then
56       rep=xgetmouse();
57       x0=rep(1); y0=rep(2);
58       if (xmin<x0 & x0<xmax & ymin<y0 & y0<ymax) & (abs(x0-oldx0)>=dx | abs
             (y0-oldy0)>=dy) then
59         sol=ode([x0;y0],t0,t,rtol,atol,f);
60         p.data=[sol(1,:)' sol(2,:)'];
61         oldx0=x0; oldy0=y0;
62       end
63     end
64   end
65 end
```

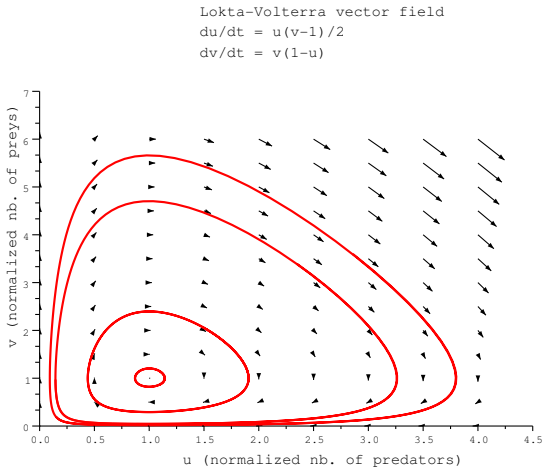


Figure 7.1: The vector field and some trajectories (with different initial conditions) for the Lokta-Volterra system (7.3),(7.4) with $\alpha\frac{1}{2}$ in the state space (u, v). One can numerically observe a periodical behaviour of the solutions.

7.1.1 Analysis of some qualitative properties of the solutions

It can be remarked that any solution of the system (7.3),(7.4) such that $u \neq 1$ and $v \neq 0$ at any time, verifies the equation

$$\frac{du}{dv} = \alpha \frac{u(v-1)}{v(1-u)} \tag{7.6}$$

which can be easily integrated as

$$\alpha v + u - \log(v^\alpha u) = \alpha v^0 + u^0 - \log((v^0)^\alpha u^0).$$

From the study of the function $f(u,v) = \alpha v + u - \log(v^\alpha u)$ in $\mathbb{R}^+_\star$, one can prove the periodical feature of the differential system.

Let us now consider the equilibrium states of the system (7.3),(7.4). The constants states solutions of $\dot{u} = \dot{v} = 0$ are $(u,v) = (0,0)$ and $(u,v) = (1,1)$. Linearizing the system (7.3),(7.4) in the vicinity of $(0,0)$ gives

$$\frac{du}{dt} = -\alpha u,$$

$$\frac{dv}{dt} = v.$$

In particular $v(t) = v^0\, e^t$ and thus $(0,0)$ is an unstable equilibrium. Let us now consider the second equilibrium $(u,v) = (1,1)$. By considering the change of variable $u = 1+w$, $v = 1+z$, the linearized system in variables (w,z) in the vicinity of $(0,0)$ gives

$$\frac{dw}{dt} = \alpha z,$$

$$\frac{dz}{dt} = -w$$

in the form

$$\frac{d}{dt}(w,z)^T = \boldsymbol{A}\,(w,z)^T \tag{7.7}$$

with

$$\boldsymbol{A} = \begin{pmatrix} 0 & \alpha \\ -1 & 0 \end{pmatrix} \tag{7.8}$$

with solution $(w,z)^T(t) = \exp(\boldsymbol{A}t)(w^0,z^0)^T$, $t \geq 0$. The stability of the equilibrium depends of the eigenstructure of $\boldsymbol{A}$. By denoting λ_1 and λ_2 the two eigenvalues of $\boldsymbol{A}$ in $\mathbb{C}$, we clearly have $tr(\boldsymbol{A}) = 0 = \lambda_1 + \lambda_2$ and $det(\boldsymbol{A}) = \lambda_1\lambda_2 = \alpha > 0$. This shows that $\lambda^1 = \bar{\lambda}^2 = i\sqrt{\alpha}$. Both eigenvalues are pure imaginary complexes. This shows that the equilibrium $(1,1)$ is a center in the theory of dynamical systems. In particular, solutions are periodical, oscillating toward the center with a period $\frac{2\pi}{\sqrt{\alpha}}$.

7.2 A fractional step approach to solve the Lokta-Volterra equations.

The Strang splitting presented in the previous chapter can be used here as a "numerical solver" of the nonlinear Lokta-Volterra equations. Let Δt be a small time step. A second-order fractional step approach can be we following:

1. First solve the following problem over a time step $\frac{\Delta t}{2}$:

$$\frac{du}{dt} = \alpha u(v-1), \tag{7.9}$$

$$\frac{dv}{dt} = 0, \tag{7.10}$$

$$u(0) = u^0, \tag{7.11}$$

$$v(0) = v^0. \tag{7.12}$$

The analytical solution of (7.9)-(7.12) at time $\Delta t/2$ is

$$v^{\star,1} = v^0,\ u^{\star,1} = u_0\, e^{\alpha(v^0-1)\frac{\Delta t}{2}}. \tag{7.13}$$

2. Second, solve the following problem over a time step Δt:

$$\frac{du}{dt} = 0, \tag{7.14}$$

$$\frac{dv}{dt} = v(1-u) \tag{7.15}$$

$$u(0) = u^{\star,1}, \tag{7.16}$$

$$v(0) = v^{\star,1}. \tag{7.17}$$

The analytical solution of (7.14)-(7.17) at time Δt is

$$u^{\star,2} = u^{\star,1},\ v^{\star,2} = v^{\star,1}\, e^{(1-u^{\star,1})\Delta t}. \tag{7.18}$$

3. Finaly solve the following problem over a time step $\frac{\Delta t}{2}$:

$$\frac{du}{dt} = \alpha u(v-1), \tag{7.19}$$

$$\frac{dv}{dt} = 0, \tag{7.20}$$

$$u(0) = u^{\star,2}, \tag{7.21}$$

$$v(0) = v^{\star,2}. \tag{7.22}$$

The analytical solution of (7.19)-(7.21) at time $\Delta t/2$ is

$$v^{\star,3} = v^{\star,2},\ u^{\star,3} = u_{\star,2}\, e^{\alpha(v^{\star,2}-1)\frac{\Delta t}{2}}. \tag{7.23}$$

To summarize, a second order approximation $(u(\Delta t), v(\Delta t))$ for the Lokta-Volterra equations at time Δt is

$$u^\star = u_0\, e^{\alpha(v^0-1)\frac{\Delta t}{2}}, \tag{7.24}$$

$$v(\Delta t) = v_0\, e^{(1-u^\star)\Delta t}, \tag{7.25}$$

$$u(\Delta t) = u^\star\, e^{\alpha(v(\Delta t)-1)\frac{\Delta t}{2}}. \tag{7.26}$$

The following `Scilab` program implements the integration scheme (7.24)-(7.26) and solves the Lokta-Volterra problem with $(u^0, v^0) = (2, 4)$. Figure 7.2 is the graphical output the of `Scilab` program `loktastrang.sce`.

```
1 // Loktastrang.sce (Scilab)
2 // Solution of the Lokta-Volterra equations
3 // using a second order Strang splitting as
4 // integration scheme
```

```
//
clear;
alpha = 0.5;
u(1,1) = 2;
v(1,1) = 4;
t(1,1) = 0;
dt    = 0.1;
//
for n=1:400
  ustar = u(n,1) *exp(alpha*(v(n,1)-1)*dt/2);
  v(n+1,1) = v(n,1) * exp((1-ustar)*dt);
  u(n+1,1) = ustar*exp(alpha*(v(n+1,1)-1)*dt/2);
  t(n+1,1) = t(n,1) + dt;
end; //for n
//
clf();
subplot(1,2,1), plot(u, v, '.-');xlabel("u");ylabel("v");xgrid();
subplot(1,2,2), plot(t, u, 'x-', t, v, '+-');
xlabel("Time t"); ylabel("u(t) and v(t)"); xgrid();
```

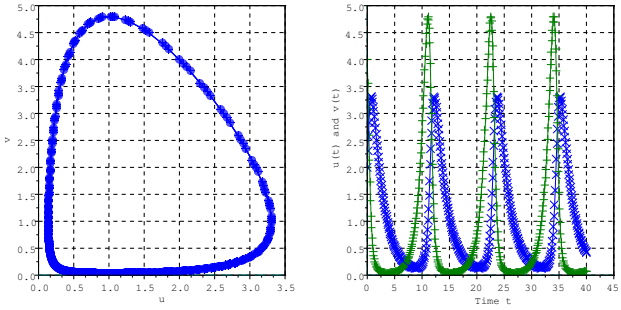


Figure 7.2: Numerical solution of the Lokta-Volterra solver (7.24)-(7.26) plotted in the spate space first and secondly as a time series (for both u and v).

7.3 Introducing spatial effects, population diffusion phenomenon

Suppose now that the spatial distribution of predators and preys is no more homogeneous. The individual random walk of both predators and preys introduces at the macroscopic scale a spatial diffusion operator. Assuming that the diffusion is isotropic in any direction, the dynamic system then becomes the PDE reaction-diffusion system

$$\frac{\partial u}{\partial t} - \nu \Delta u = \alpha\, u(v-1), \tag{7.27}$$

$$\frac{\partial v}{\partial t} - \mu \Delta v = v(1-u) \tag{7.28}$$

from some diffusion coefficients $\nu, \mu > 0$. The following `Freefem++` program `LoktaVolterra.edp` implements the Strang fractional step method with successive solutions of reactions and diffusion

problems, a spatial P2 Finite Element approximation is used. In this numerical example, $\nu = 10^{-3}$, $\mu = 10^{-4}$ and the following initial data is used:

$$u^0(\boldsymbol{x}) = 0.1, \quad v^0(\boldsymbol{x}) = 2\,1_{\left(0<x,y<\frac{1}{2}\right)}(\boldsymbol{x}).$$

Zero-flux homogeneous Neumann boundary conditions are used.

```
// LoktaVolterra.edp (Freefem++)
// Solve the Rreaction-Diffusion equation
// of the Lokta-Volterra Prey-predator model
// by the Strang splitting fractional step method.
// The differential reaction system is itself solved
// by a Strang splitting . Square spatial domain
//
mesh Th = square(40, 40);
fespace Vh(Th, P2);
fespace Wh(Th, P1);
real alpha = 0.5;
real t = 0.0;
real dt = 0.2;
real nu = 0.001;
real mu = 0.0001;
real theta = 0.49;
int it=0;
//
Vh u, uh, uold;
Vh v, vh, vold;
Wh dxu, dyu, dxv, dyv;
//
problem heatu (u, uh, init=it) =
  int2d(Th)(u*uh/dt)
  -int2d(Th)(uold*uh/dt)
  +int2d(Th)(nu*(1-theta)*dx(u)*dx(uh)
          +nu*(1-theta)*dy(u)*dy(uh))
  +int2d(Th)(nu*(theta)*dx(uold)*dx(uh)
          +nu*(theta)*dy(uold)*dy(uh));
//
problem heatv (v, vh, init=it) =
  int2d(Th)(v*vh/dt)
  -int2d(Th)(vold*vh/dt)
  +int2d(Th)(mu*(1-theta)*dx(v)*dx(vh)
          +mu*(1-theta)*dy(v)*dy(vh))
  +int2d(Th)(mu*(theta)*dx(vold)*dx(vh)
          +mu*(theta)*dy(vold)*dy(vh));
//
//  Initializing
u = 1 + 0.5*cos(3*pi*x)*sin(3*pi*y); uold = u;
v = 1 + 0.5*sin(3*pi*x)*cos(3*pi*x); vold = v;
//
// Big loop in time
for (it=0; it<200; it++) {
  for (int subit=0; subit<1; subit++){
```

```
46      t = t + dt;
47      // Fractional step method
48      // Step a. Solve the reaction system on (dt/2)
49      u = u * exp(alpha*(v-1)*dt*0.25);
50      v = v * exp((1-u)*dt*0.5);
51      u = u * exp(alpha*(v-1)*dt*0.25);
52      uold = u; vold = v;
53      //
54      // Step b. Solve the diffusion system on (dt)
55      heatu; uold = u;
56      heatv; vold = v;
57      //
58      // Step c. Solve the reaction system on (dt/2)
59      u = u * exp(alpha*(v-1)*dt*0.25);
60      v = v * exp((1-u)*dt*0.5);
61      u = u * exp(alpha*(v-1)*dt*0.25);
62      uold = u; vold = v;
63    }
64    plot(u, nbiso=40, value=1, fill=0);//, ps="crd_u_it="+it+".eps");
65    plot(v, nbiso=40, value=1, fill=0);//, ps="crd_v_it="+it+".eps");
66    //
67  } // for it
```

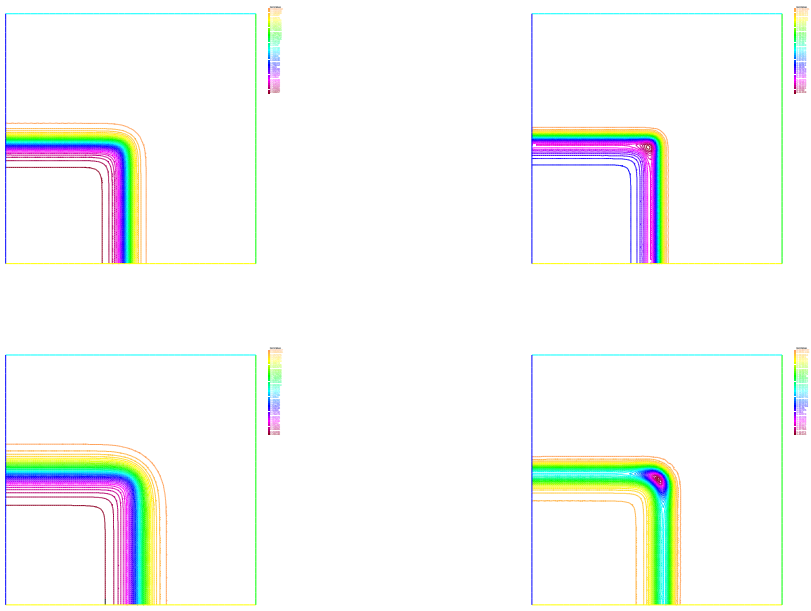


Figure 7.3: Lokta-Volterra numerical solutions at some instants. Iso-contours of u (left) and v (right).

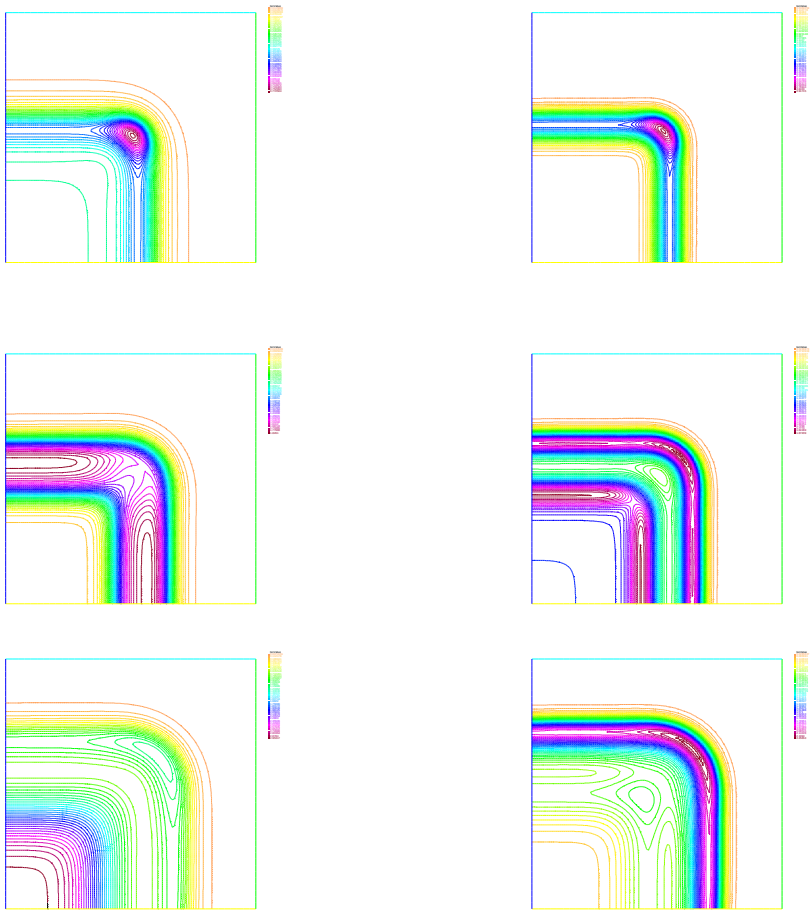


Figure 7.4: Lokta-Volterra numerical solutions at some instants. Iso-contours of u (left) and v (right).

7.4 Adding seasonal migration into the model

Preys usually migrate in order to find better places for food. In a seasonal migration, animals are moving from north to south and vice-versa. This can be coarsely modeled by an independent time-dependent vertical and periodic vector field

$$\boldsymbol{a}(\boldsymbol{x}, t) = (0, \sigma \sin(\omega t)) \, .$$

Then the equations of evolution of predator and prey density are

$$\frac{\partial u}{\partial t} - \nu \Delta u = \alpha \, u(v-1), \tag{7.29}$$

$$\frac{\partial v}{\partial t} + \boldsymbol{a} \cdot \nabla v - \mu \Delta v = v(1-u) \tag{7.30}$$

In this case, the equation splitting can be on one side a transport-diffusion equation

$$\frac{\partial u}{\partial t} - \nu \Delta u = 0,$$
$$D_t v - \mu \Delta v = 0,$$

with

$$D_t = \partial_t + \boldsymbol{a} \cdot \nabla,$$

and on the other side a (nonlinear) reaction equation

$$\frac{du}{dt} = \alpha u(v-1),$$
$$\frac{dv}{dt} = v(1-u).$$

The implementation is also the same as in a previous case. Here a Lagrangian derivative $D_t v$ has to be discretized in a stable way, for example using the method of characteristics. In `Freefem++` the `convect` function can be used for example. Numerical results are performed with the parameters $\nu = 5\,10^{-4}$, $\mu = 2\,10^{-4}$, $\Delta t = 0.08$, $\sigma = 0.4$, $\omega = \pi$ and $\alpha = 0.5$. The initial condition considered in this simulation is

$$u^0(\boldsymbol{x}) = 0.1, \quad v^0(\boldsymbol{x}) = 1_{\left((x-\frac{1}{2})^2+(y-\frac{1}{2})^2 \leq \frac{1}{32}\right)}(\boldsymbol{x}).$$

Periodic boundary conditions are used for the top and bottom borders while zero-flux boundary conditions are used for both left and right borders.

From figures 7.5 to 7.6, isocontours of predator and prey densities at successive instants. One can observe the complex dynamics of the migration for both two populations, with moving regions of both high and low density. Predators tend to "follow" preys during their seasonal migration.

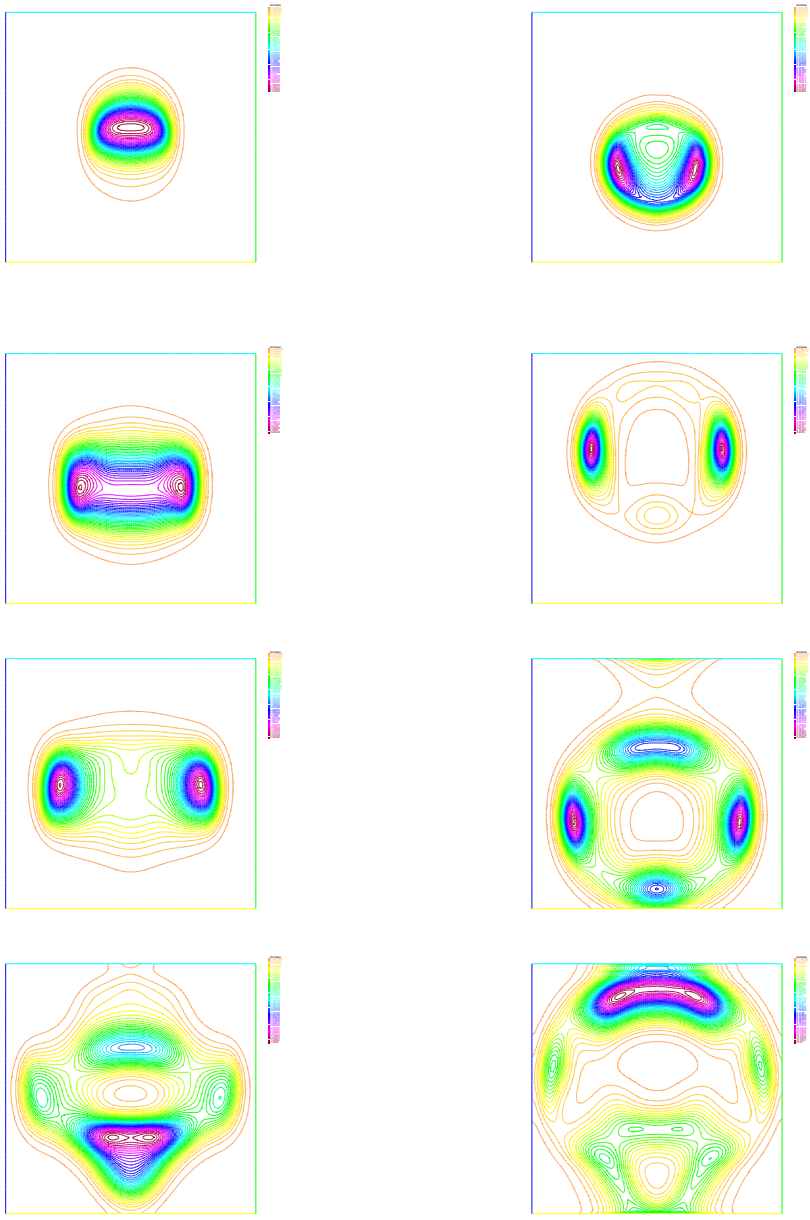


Figure 7.5: Convection-Reaction-Diffusion prey-predator Lokta-Volterra numerical solutions at successive instants. Iso-contours of predator density u (left) and prey density v (right).

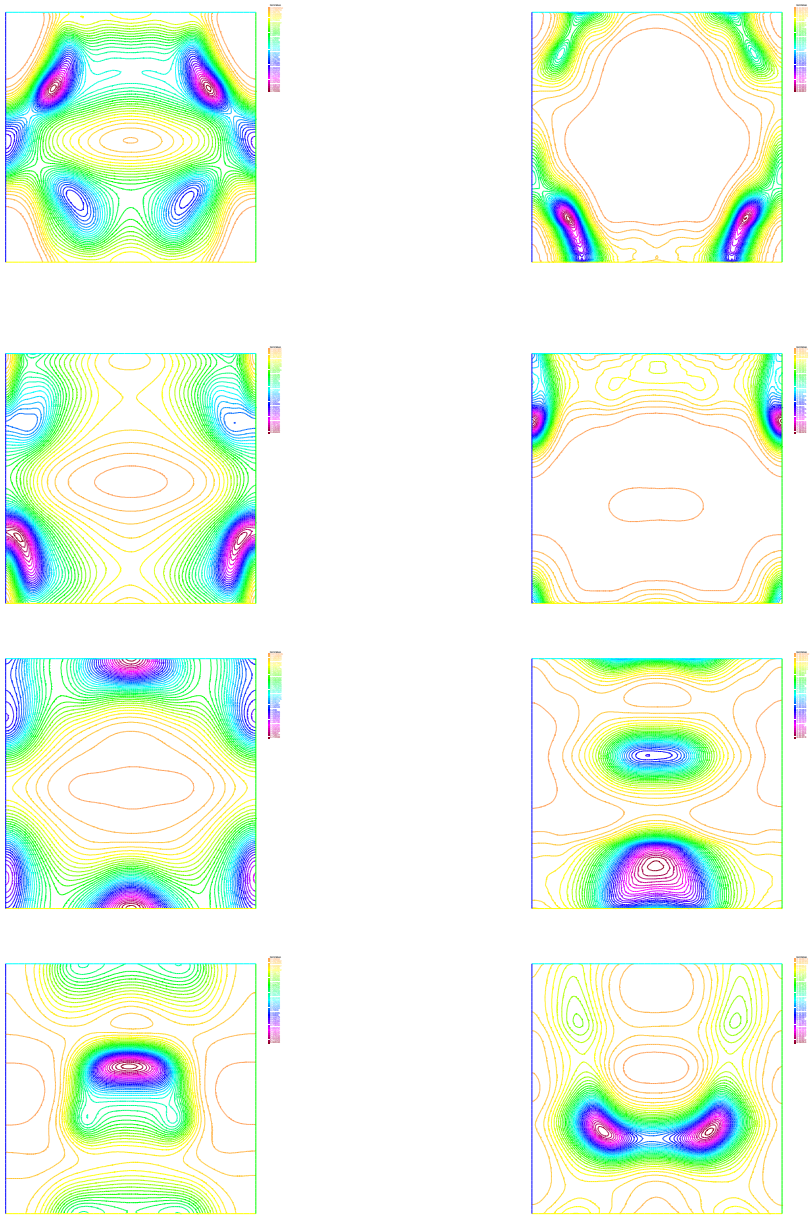


Figure 7.6: Convection-Reaction-Diffusion prey-predator Lokta-Volterra numerical solutions at successive instants. Iso-contours of predator density u (left) and prey density v (right).

Chapter 8

Model of biological spatial pigment pattern formation

During the development of an embryo, there is a rapid growth, not only in cell numbers, but also in specialization and complex organization among cells. Cells in the vertebrate embryo divide, migrate, differentiate and for the various organs of the body. Many of the structures have a regular pattern such as the vertebrae in the spine, the pattern of feather, etc.

Pigment patterns are generated by chromatophore cells which lie in the dermal or epidermal layers of the skin. There are several types of chromatophores each containing different pigments; the most common are melanin-bearing cells and melanophores which contain black, brown or yellow pigments. During development, pigment cell precursors - chromatoblasts - originate in the neural crest. These cells spread over the skin at a roughly uniform density. Whether or not the skin develops a pigmented patch depends on whether pigment cells produce pigment or remain quiescent. chromatophore interactions may result in pigmented cells and unpigmented cells gathering in different regions to produce stripes or spots.

The principle mathematical models for pigmentation to date have been mainly reaction-diffusion models, pioneered in the 1980s. These models hypothesize the existence of chemicals (morphogens) which react and diffuse and, under appropriate conditions, generate spatially heterogeneous patterns. This chemical landscape is viewed as a pre-pattern to which cells then respond in some genetically predetermined way and differentiate accordingly.

Oster and Murray (1989, []) proposed a simple cell-chemotaxis model for pattern formation which takes account of cell motility and chemotaxis, the chemical process by which cells migrate up a chemical gradient. Some of the developments of this chapter are pioneered by Maini et al. [] who have established that a cell-chemotaxis model can produce a wide variety of observed patterns.

8.1 Cell chemotaxis model

The model mechanism involves the cell density, $\rho(\boldsymbol{x}, t)$ and chemo-attractant concentration $c(\boldsymbol{x}, t)$, where $\boldsymbol{x}$ and t are the spatial coordinate and time respectively, and consists of equations which describe their motion and net production. The general form of the cell equation is the conservative balance

equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \boldsymbol{J}_\rho = R(\rho) \tag{8.1}$$

where $\boldsymbol{J}_\rho$ is the flux of cells and $R(\rho)$ is the local net cell production. From biological considerations, it is assumed that there is two contributions to the flux term, namely a random Fickian diffusion process with diffusive flux

$$\boldsymbol{J}^d = -D_\rho \nabla \rho \tag{8.2}$$

where D_ρ is the diffusion coefficient, and chemotaxis with chemotactic flux

$$\boldsymbol{J}^c = \alpha \rho \nabla c \tag{8.3}$$

where α is the chemotaxis coefficient. Remark that $\boldsymbol{J}^c$ is a nonlinear term. We take the cell production term to be adequately described by logistic growth of the form $R(\rho) = r\rho(\rho_\infty - \rho)$ where $r\rho_\infty$ is the linear mitotic growth rate with r and ρ_∞ both nonnegative constants. The logistic growth rate is the simplest way to describe the characteristic sigmoidal growth exhibited by several cell types. As summary, the equation for cell density is

$$\frac{\partial \rho}{\partial t} + \alpha \nabla \cdot (\rho \nabla c) - D_\rho \Delta \rho = r\rho(\rho_\infty - \rho). \tag{8.4}$$

It is a nonlinear convection-reaction-diffusion equation. We assume that the cell secretes its own chemoattractant according to a growth rate in the form

$$\frac{S\rho}{\beta + \rho}$$

It is also supposed to diffuse with diffusion coefficient D_c and to linearly degrade (apoptosis) at a rate γc. The equation for chemotactic concentration c is then

$$\frac{\partial c}{\partial t} - D_c \Delta c = \frac{S\rho}{\beta + \rho} - \gamma c \tag{8.5}$$

with positive constants S, β and γ. Equation (8.4) and (8.5) form a coupled system of nonlinear partial differential equations.

8.1.1 Dimensionless equations

To reduce the number of parameters, it is usual to write the model in non-dimensional terms. For any quantity q, let us denote by q_0 a dimensional constant and the non-dimensional quantity $\bar{q}$ defined as

$$\bar{q} = \frac{q}{q_0}.$$

By denoting $\bar{\nabla}$ and $\bar{\Delta}$ respectively the gradient and the Laplace operator with respect to the non-dimensional space variable $\bar{\boldsymbol{x}}$, the non-dimensional equations write

$$\frac{\rho_0}{t_0}\frac{\partial \bar{\rho}}{\partial \bar{t}} + \frac{\alpha \rho_0 c_0}{(x_0)^2}\alpha \bar{\nabla} \cdot (\bar{\rho}\bar{\nabla}\bar{c}) - \frac{D_\rho}{(x_0)^2}\bar{\Delta}\bar{\rho} = r(\rho_0)^2 \bar{\rho}(\frac{\rho_\infty}{\rho_0} - \bar{\rho})$$

and

$$\frac{c_0}{t_0}\frac{\partial \bar{c}}{\partial \bar{t}} - \frac{c_0}{(x_0)^2} D_c \bar{\Delta}\bar{c} = \frac{S\rho_0}{\beta}\frac{\bar{\rho}}{1 + \frac{\rho_0}{\beta}\bar{\rho}} - \gamma c_0 \bar{c}.$$

Introducing a scaling factor s, the following choice

$$x_0 = \sqrt{\frac{D_c s}{\gamma}},\ t_0 = \frac{s}{\gamma},\ \rho_0 = \beta,\ c_0 = \frac{S}{\gamma},$$

gives the non-dimensional equations become (omiting the bar symbols for the sake of simplicity)

$$\frac{\partial \rho}{\partial t} + \alpha \nabla \cdot (\rho \nabla c) - D\Delta\rho = sr\rho(\rho_\infty - \rho), \tag{8.6}$$

$$\frac{\partial c}{\partial t} - \Delta c = s\left(\frac{\rho}{1+\rho} - c\right). \tag{8.7}$$

For simplicity, the spatial domain Ω is supposed to be a rectangle of respective lengths L_x and L_y. Moreover we consider zero flux boundary conditions, meaning that no cell or chemoattractant migrates through the boundary:

$$\nabla\rho \cdot \boldsymbol{n} = \nabla c \cdot \boldsymbol{n} = 0, \quad \boldsymbol{x} \in \partial\Omega. \tag{8.8}$$

8.2 Zero-dimensional model

Forgetting the spatial terms, the so-called 0-dimensional model is

$$\frac{d\rho}{dt} = sr\rho(\rho_\infty - \rho), \tag{8.9}$$

$$\frac{dc}{dt} = s\left(\frac{\rho}{1+\rho} - c\right). \tag{8.10}$$

There are two equilibrium states $(\rho, c) = (0, 0)$ and $(\rho, c) = (\rho_\infty, \frac{\rho_\infty}{1+\rho_\infty})$. To know wether those equilibrium states are stable or unstable, one linearizes the dynamical system in the vicinity of the equilibrium. For (ρ, c) small enough, the system is equivalent to

$$\frac{d}{dt}\begin{pmatrix}\rho\\ c\end{pmatrix} = \begin{pmatrix} sr\rho_\infty & 0\\ s & -s\end{pmatrix}\begin{pmatrix}\rho\\ c\end{pmatrix}.$$

There are two real eigenvalues for the linearized system, one of them is positive so that the equilibrium state is unstable. For the second equilibrium $(\rho, c) = (\rho_\infty, \frac{\rho_\infty}{1+\rho_\infty})$, using the new variables $\rho' = \rho - \rho_\infty$, $c' = c - \frac{\rho_\infty}{1+\rho_\infty}$, for (ρ', c') small enough, the system is equivalent to the linearized one

$$\frac{d}{dt}\begin{pmatrix}\rho'\\ c'\end{pmatrix} = \begin{pmatrix} -sr\rho_\infty & 0\\ \frac{s}{(1+\rho_\infty)^2} & -s\end{pmatrix}\begin{pmatrix}\rho'\\ c'\end{pmatrix}.$$

According to the sign of the two eigenvalues, the second equilibrium state is stable.

8.3 Linear stability analysis of the complete model

Now we carry out the analysis of the system (8.6),(8.7). It is easy to check that the constant steady-states of (8.6),(8.7) are the equilibrium states of the dynamical system (8.9),(8.10). The steady-state $(0, 0)$ is always unstable by inspection so we only consider the non-zero steady state $(\rho, c) =$

$(\rho_\infty, \frac{\rho_\infty}{1+\rho_\infty})$ here.

We set $\rho = \rho_\infty + u$ and $c = \frac{\rho_\infty}{1+\rho_\infty} + v$ where $|u|, |v|$ are small, substitute into (8.6)-(8.8) and only retain linear terms. This gives the PDE problem of linear partial differential equations which governs the behaviour near the steady state:

$$\partial_t u - D\Delta u + \alpha\rho_\infty \Delta v = -rs\rho_\infty u \quad \text{in } \Omega, \tag{8.11}$$

$$\partial_t v - \Delta v = s\left(\frac{u}{(1+\rho_\infty)^2} - v\right) \quad \text{in } \Omega, \tag{8.12}$$

$$\boldsymbol{n}\cdot\nabla u = \boldsymbol{n}\cdot\nabla v = 0 \quad \text{on } \partial\Omega. \tag{8.13}$$

We look for planar wave solutions of (8.11)-(8.13) i.e. solutions in the form

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \exp(i\boldsymbol{k}\cdot\boldsymbol{x} + \lambda t) \tag{8.14}$$

where $\lambda = \lambda(\boldsymbol{k})$ determines the temporal growth rate of the disturbance with wave vector $\boldsymbol{k}$. Let us denote $k = |\boldsymbol{k}|$. Putting (8.14) into (8.11),(8.12) gives

$$\lambda u + Dk^2 u + \alpha\rho_\infty k^2 v = -r\rho_\infty s u,$$

$$\lambda v + k^2 v = s\left(\frac{u}{(1+\rho_\infty)^2} - v\right).$$

These inequalities have to be satisfied for $\boldsymbol{x}$ and any t. Then we have the compatibility linear system

$$\begin{pmatrix} Dk^2 - r\rho_\infty s & \alpha\rho_\infty k^2 v \\ \frac{s}{(1+\rho_\infty)^2} & k^2 + s \end{pmatrix} \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = -\lambda \begin{pmatrix} u_0 \\ v_0 \end{pmatrix}$$

This is an eigenvalue problem. Non-trivial solutions for u_0 and v_0 exist only if λ, the dispersion relation satisfies the characteristic polynomial

$$\lambda^2 + \left[(D+1)k^2 + r\rho_\infty + s\right]\lambda + \left[Dk^4 + \left\{r\rho_\infty s + Ds - \frac{s\rho_\infty\alpha}{(1+\rho_\infty)^2}\right\}k^2 + r\rho_\infty s^2\right] = 0. \tag{8.15}$$

Moreover, the wave with wavenumber $|\boldsymbol{k}|$ must satisfy the boundary conditions (8.13). Denoting $\boldsymbol{k} = (k_x, k_y)$, we get the compatibility conditions

$$k_x = \frac{m\pi}{L_x},\ k_y = \frac{\ell\pi}{L_y}, \quad m, \ell \in \mathbb{N}.$$

The linear spatial eigenmodes then are $\cos m\pi x/L_x \cos \ell\pi y/L_y$. In (8.15), $\lambda = \lambda(k^2)$ clearly depends on k^2. If $\lambda(k^2) < 0$, then a disturbance of wavevector $\boldsymbol{k}$ will decay in time. If $\lambda(k^2) > 0$ for some k^2 then the disturbance with wavenumbers will grow and the system will evolve to a nonuniform spatially structured solution. On the rectangular domain $[0, L_x] \times [0, L_y]$, the values of k^2 which produce a pattern are those where $\lambda(k^2) > 0$ with

$$k^2 = \boldsymbol{k}\cdot\boldsymbol{k} = \pi^2\left(\frac{m^2}{L_x^2} + \frac{\ell^2}{L_y^2}\right). \tag{8.16}$$

The critical value occur when $\lambda(k^2) = 0$, that is when k satisfies

$$Dk^4 + \left\{ r\rho_\infty s + Ds - \frac{s\rho_\infty \alpha}{(1+\rho_\infty)^2} \right\} k^2 + r\rho_\infty s^2 = 0. \tag{8.17}$$

In order to find at least one unstable mode, we require equation (8.17) to have only one double root solution for k^2, so we further impose the condition for equal roots, namely

$$\left[r\rho_\infty s + Ds - \frac{s\rho_\infty \alpha}{(1+\rho_\infty)^2} \right]^2 - 4Dr\rho_\infty s^2 = 0. \tag{8.18}$$

Hence the modulus of the critical wave vector is given by

$$k_c^2 = \frac{2s\sqrt{Dr\rho_\infty}}{2D} = s\left(\frac{r\rho_\infty}{D}\right)^{1/2}. \tag{8.19}$$

By choosing D, s, r and N appropriately, we can find a k^2 from (8.16) which satisfies equation (8.19), and then solve equation (8.18) for α (one can take the larger root for α so that k_c^2 is positive). This determines the point in $(\rho_\infty, D, r, s, \alpha)$ parameter space where the mode (8.19) is isolated.

8.3.1 Continuous variation of a single parameter

From equations (8.17)-(8.19), it is clear that by making appropriately vary any of the five parameters r, ρ_∞, s, D or α, the uniform steady state can evolve to a non-uniform steady-state. The chemotaxis parameter α is a key parameter so one can fix the others and make vary α to locate bifurcations in α and follow the corresponding solutions.

8.4 Numerical discretization

The problem is hard to solve because of strongly nonlinear terms and important convective effects. Recall the equation of cell density

$$\partial_t \rho + \nabla \cdot (\alpha \rho \nabla c) - D\Delta\rho = sr\rho(\rho_\infty - \rho). \tag{8.20}$$

First, let us emphasize the convection term. By denoting

$$\boldsymbol{u} = \alpha \nabla c, \tag{8.21}$$

the cell density equation can be rewritten

$$D_t \rho + \alpha\rho\Delta c - D\Delta\rho = sr\rho(\rho_\infty - \rho) \tag{8.22}$$

where D_t denotes the particle derivative

$$D_t = \partial_t + \boldsymbol{u} \cdot \nabla. \tag{8.23}$$

8.4.1 Fractional step method

The term $\alpha\rho\delta c$ is not easy to treat because of its nonconservative form and nonlinear nature. Thus it is appropriate to consider here a fractional step method. A fractional step method allows us to successively deal with each term of an operator in an ODE or a PDE. Consequently, a time step is a multi-step process made of several steps that "solve" each part of the equation treated separately. For equation (8.20), it is convenient first to solve

$$D_t\rho - D\Delta\rho = sr\rho(\rho_\infty - \rho) \tag{8.24}$$

which only depends on c by means of D_t, then solve the system

$$\partial_r\rho + \alpha\rho\Delta c = 0, \tag{8.25}$$

$$\partial_t c - \Delta c = s\left(\frac{\rho}{1+\rho} - c\right). \tag{8.26}$$

Remark that equation (8.25) can be rewritten in conservation form

$$\partial_t\sigma + \alpha\Delta c = 0 \tag{8.27}$$

with $\sigma = \log(\rho)$. A semi-discretization in time gives the following scheme:

1. Solve the semi-implicit linear scheme in $\rho^\star$:

$$\frac{\rho^\star - \rho^n \circ X^n}{\Delta t^n} - D\Delta\rho^\star = sr\rho^n(\rho_\infty - \rho^\star) \tag{8.28}$$

with boundary conditions

$$\frac{\partial\rho^\star}{\partial \boldsymbol{n}} = 0. \tag{8.29}$$

The function $\rho^\star$ will serve as "initial data" of cell density for the next step:

2. Solve the following linear problem ($\sigma^\star = \log(\rho^\star)$):

$$\frac{\sigma^{n+1} - \sigma^\star}{\Delta t^n} + \alpha\Delta c^{n+1} = 0 \quad \text{in } \Omega, \tag{8.30}$$

$$\frac{\partial\sigma^{n+1}}{\partial \boldsymbol{n}} = 0 \quad \text{on } \partial\Omega, \tag{8.31}$$

$$\frac{c^{n+1} - c^n}{\Delta t^n} - \Delta c^{n+1} = s\left(\frac{\rho^\star}{1+\rho^\star} - c^{n+1}\right) \quad \text{in } \Omega, \tag{8.32}$$

$$\frac{\partial c^{n+1}}{\partial \boldsymbol{n}} = 0 \quad \text{on } \partial\Omega, \tag{8.33}$$

then compute

$$\rho^{n+1} = \exp(\sigma^{n+1}) \tag{8.34}$$

to end the time iteration $t^n \to t^{n+1}$.

Of course, the global time advance scheme is not fully implicit and thus may not be unconditionally stable.

8.4.2 Full discretization

The full discretization of the problem will correspond to Finite Element space discretization of the semi-discretized scheme (8.28)-(8.33).

1. The variational formulation of (8.28),(8.29) leads to

$$\int_{\Omega^h} \frac{\rho^\star - \rho^n \circ X^n}{\Delta t^n} v^h \, d\boldsymbol{x} + \int_{\Omega^h} D\nabla\rho^\star \cdot \nabla v^h \, d\boldsymbol{x} = sr \int_{\Omega^h} \rho^n(\rho_\infty - \rho^\star) \, v^h \, d\boldsymbol{x} \quad \forall v^h \in V^h. \tag{8.35}$$

   Once $\rho^\star \in V^h$ is computed by (8.35) then compute

$$\sigma^\star = P^{W^h}(\log(\rho^\star)) \tag{8.36}$$

   as the projection on the σ-Finite Element space of the field $\log(\rho^\star)$.

2. The variational formulation of (8.30)-(8.33) leads to

$$\begin{aligned} &\int_{\Omega^h} \frac{\sigma^{n+1} - \sigma^\star}{\Delta t^n} \sigma^h \, d\boldsymbol{x} - \int_{\Omega^h} \alpha\nabla c^{n+1} \cdot \nabla\sigma^h \, d\boldsymbol{x} \\ &+ \int_{\Omega^h} \frac{c^{n+1} - c^n}{\Delta t^n} c^h \, d\boldsymbol{x} + \int_{\Omega^h} \nabla c^{n+1} \cdot \nabla c^h \, d\boldsymbol{x} - s \int_{\Omega^h} \left(\frac{\rho^\star}{1+\rho^\star} - c^{n+1} \right) c^h \, d\boldsymbol{x} = 0 \\ &\forall \sigma^h \in W^h, \; c^h \in X^h. \end{aligned} \tag{8.37}$$

   To finish, let compute

$$\rho^{n+1} = P^{V^h}(\exp(\sigma^{n+1})) \tag{8.38}$$

   as the projection on V^h of the field $\exp(\sigma^{n+1})$.

To completely define the numerical method, we have to choose some convenient Finite Element spaces. The classical P^1 is convenient for both ρ and σ. However, for c the P^2 Finite Element space is preferable because we need to compute the convective vector field

$$\boldsymbol{u} = \alpha\nabla c$$

from c. If c is piecewise P^2 in a triangle, then ∇c will be piecewise P^1.

8.4.3 `freefem++` source code of the numerical scheme and numerical results

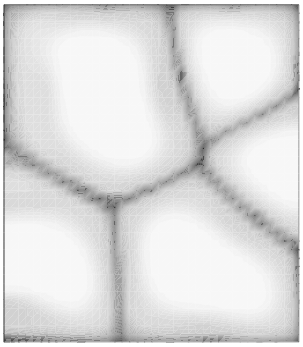
```
// Chemotaxis.edp (freefem++)
// Chemotaxis model for biological pattern generation
//
// Parameter definition
real r = 38.05;
real alpha = 285;
real rhoinf =1;
real D = 0.25;
real s = 1;
real Lx = 3.5;
real Ly = 4;
```

```
real dt = 0.05;
//
mesh Th=square(40, 40, [Lx*x, Ly*y]);
plot(Th, wait=0);
fespace Vh(Th, P1);
Vh rho, rhoold, rhotest, u1, u2, output;
Vh sigma, sigmaold, sigmatest;
fespace Wh(Th, P2);
Wh c, cold, ctest;
// Starting from the (unstable) constant state
rho = rhoinf;
rhoold=rho;
c   = rhoinf/(1+rhoinf);
cold=c;
u1 = alpha * dx(c);
u2 = alpha * dy(c);
// Then go to the PDE problem
problem step1(rho, rhotest) =
  int2d(Th)(rho*rhotest /dt)
  -int2d(Th)( convect([u1,u2], -dt, rhoold)*rhotest /dt)
  +int2d(Th)(D*dx(rho)*dx(rhotest)+D*dy(rho)*dy(rhotest))
  -int2d(Th)(r*s*rhoold*rhoinf*rhotest)
  +int2d(Th)(r*s*rhoold*rho*rhotest);

problem step2([sigma, c], [sigmatest, ctest]) =
   int2d(Th)( sigma*sigmatest /dt)
  -int2d(Th)( sigmaold*sigmatest /dt)
  -int2d(Th)(alpha*dx(c)*dx(sigmatest)+alpha*dy(c)*dy(sigmatest))
  +int2d(Th)(c*ctest /dt)
  - int2d(Th)(cold*ctest /dt)
  +int2d(Th)(dx(c)*dx(ctest) + dy(c)*dy(ctest))
  -int2d(Th)(s*rho/(1+rho)*ctest)
  +int2d(Th)(s*c*ctest);

for (int it=0; it<200; it++) {
cout << "it = " << it << endl;
// Step 1
step1;
rhoold = rho;
sigmaold = log(rhoold);
//
step2;
u1 = alpha * dx(c);
u2 = alpha * dy(c);
rho = exp(sigma);
rhoold = rho;
cold = c;
//
// plot(c, nbiso=60, value=1, wait=0);
plot( rho , nbiso=40, grey=1, fill=1, value=1, wait=0);
}
cout << "cmin = " << c[].min << " cmax = " << c[].max << endl;
```

```
64 cout << "rhomin = " << rho[].min << " rhomax = " << rho[].max << endl;
65 //
66 plot( rho, nbiso=60, grey=1, fill=1, value=0, wait=0, ps="rho.eps");
67 plot(c , nbiso=60, grey=1, fill=1, value=0, wait=0, ps="c.eps");
```



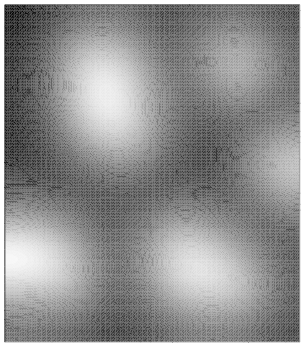
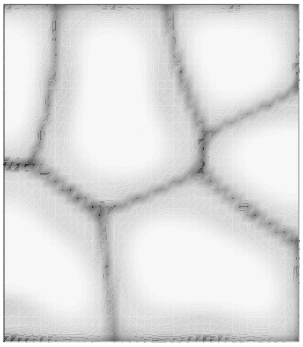


Figure 8.1: Isocontours of cell density (left) and chemoattractant concentration (right) in grey colors. The parameter alpha used here is $\alpha = 285$. Stable steady state starting from the constant unstable steady state.



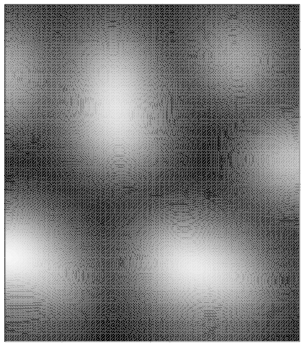


Figure 8.2: Isocontours of cell density (left) and chemoattractant concentration (right) in grey colors. The parameter alpha used here is $\alpha = 305$. Stable steady state starting from the constant unstable steady state.

Chapter 9

Vehicle traffic flow modeling

9.1 Setting of the problem

Although vehicle traffic flow is intrinsically a discrete dynamical process, it is possible to model it continuously at a certain level of description. If we are looking at a highway quite war from it, the flow can seen as a continuous flow with respect to macroscopic quantities like density ρ and flow rate q. Let us denote $S = [x_-, x_+]$ a bounded road section. Then the evolution of the quantity of vehicle on that section is governed by the gain-loss equation

$$\frac{dS}{dt} = q(x_-, t) - q(x_+, t). \tag{9.1}$$

By introducing a vehicle density $\rho = \rho(x, t)$ and assuming a continuously varying flow rate, then (9.58) can write

$$\frac{d}{dt}\int_S \rho(x, t)\, dx + \int_S \frac{\partial q}{\partial s} = 0. \tag{9.2}$$

This balance equation holds for any road section S. Then for an infinitesimal road section S with $|S| \to 0$, one gets almost everywhere the conservation equation

$$\partial_t \rho + \partial_x q = 0. \tag{9.3}$$

The class of partial differential equation (9.3) was introduced in the 1930s by Whitham. Today, the so-called Whitham first-order models are still used in transportation engineering, especially for traffic forecast and travel time estimation.

A closure is required for equation (9.3), in order to link density and flow rate. First, due to the dimension of q, it is natural to consider a flow rate as a vehicle mean velocity times the vehicle density:

$$q = \rho u. \tag{9.4}$$

Now the closure is then transfered onto the speed u. For a free highway with only few vehicles, it is natural to consider that the vehicle speed is the free speed u_f fixed by speed limitation. On the other hand, a car is "stopped" in a fully jammed lane of maximum critical density ρ_c, so its velocity can be considered as zero. For intermediate velocities, one can consider as first approximation a linear law

$$u = u(\rho) = u_f \left(1 - \frac{\rho}{\rho_c}\right). \tag{9.5}$$

The equation (9.5) is referred to as the fundamental diagram in the traffic theory. It appears that the fundamental law (9.5) is a rather good approximation of the reality. Statistics performed on real traffic measurements show that real data plotted in the density-velocity or density-occupancy state space can be linearly regressed as in (9.5). Remark that the function $\rho \mapsto q(\rho) = \rho u(\rho)$ is parabolic (see figure 9.8) with maximum flow rate

$$q_M = \frac{1}{4}\rho_c u_f \tag{9.6}$$

at density $\rho = \rho_c/2$. We fall into a nonlinear partial differential equation in the form

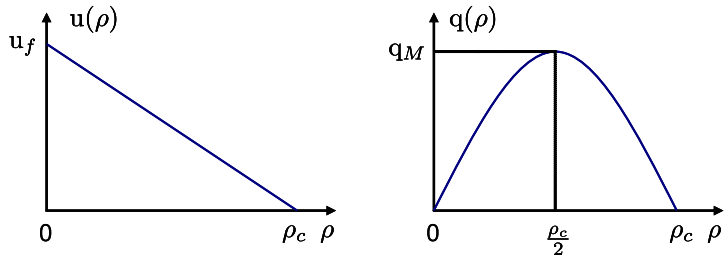


Figure 9.1: Fundamental diagram of traffic

$$\partial_t \rho + \partial_x(\rho\, u(\rho)) = 0 \tag{9.7}$$

with $u(\rho)$ given by (9.5).

9.2 Some mathematical aspects of nonlinear transport equations

Although (9.5) has a rather simple script, its mathematical analysis is not so easy and requires attention. In particular, we are going to see that solutions of equations (9.7) can develop discontinuities (also name shocks) during time, even if the initial data is smooth. When discontinuities exists, equation (9.7) is read in a weak sense, i.e. in the sense of distributions.

For smooth solutions, equation (9.7) can be written in nonconservative form. Denoting $a(\rho) = q'(\rho)$, it is equivalent to the nonlinear transport equation

$$\partial_t \rho + a(\rho)\partial_x \rho = 0 \tag{9.8}$$

where the characteristic velocity $a(\rho)$ depends itself on ρ. In the case of (9.5), we have

$$a(\rho) = u_f \left(1 - 2\,\frac{\rho}{\rho_c}\right)$$

which is different from the vehicle velocity. It is important to understand the difference between the vehicle velocity and the characteristic velocity. If we have a Lagrangian description of the flow, meaning there are (vehicle) particles flowing with differential equation for the vehicle position

$$\frac{dx}{dt} = u(\rho(x(t), t)).$$

Now consider the characteristic velocity $a(\rho) = q'(\rho)$. Suppose a density solution in the form

$$\rho(x,t) = \rho_0 + \varepsilon\rho_1(x,t) \tag{9.9}$$

for a constant ρ_0 and a small parameter $0 < \varepsilon \ll 1$, meaning that the density is almost constant up to a perturbation $\varepsilon\rho_1(x,t)$. Introducing expression (9.9) into (9.8) gives

$$\partial_t(\rho_0 + \varepsilon\rho_1) + \left(a(\rho_0) + \varepsilon\frac{da}{d\rho}(\rho_0)\rho_1\right)\partial_x(\rho_0 + \varepsilon\rho_1) = 0.$$

Then, homogeneous terms of degree 1 in ε give the linearized equation

$$\partial_t\rho_1 + a(\rho_0)\partial_x\rho_1 = 0. \tag{9.10}$$

Equation (9.10) shows that the fluctuations of the solutions are propagating at characteristic velocity $a(\rho_0)$ but not $u(\rho_0)$. The characteristic velocity is the speed of propagation of the information whereas the vehicle velocity is the material velocity.

9.2.1 Smooth autosimilar solutions

In this section, we are looking for continuous solutions of the variable $\xi = x/t$, i.e.

$$\rho(x,t) = \phi(\frac{x}{t}). \tag{9.11}$$

Putting (9.11) into (9.8) gives the differential equation

$$-\frac{x}{t^2}\phi'(\frac{x}{t}) + a(\phi(\frac{x}{t}))\frac{1}{t}\phi'(\frac{x}{t}) = 0$$

or again

$$(a(\phi(\xi) - \xi))\ \phi'(\xi) = 0, \quad \xi = \frac{x}{t}. \tag{9.12}$$

To (9.12), we will add the "initial condition"

$$\rho = \rho_L \quad \text{at point } \xi = \xi_L \tag{9.13}$$

(L means "left" for reasons that will appear later). There are different kinds of solutions for (9.12),(9.13). Either $\phi'(\xi) = 0$ and thus $\Phi(\xi)$ is locally constant. Or $\Phi'(\xi) \neq 0$ and we have the solution

$$\xi = a(\phi(\xi)). \tag{9.14}$$

In the case of (9.5), we have

$$a(\rho) = u_f\left(1 - 2\,\frac{\rho}{\rho_c}\right) \tag{9.15}$$

and thus from (9.14)

$$\phi(\xi) = \frac{\rho_c}{2}\left(1 - \frac{\xi}{u_f}\right). \tag{9.16}$$

This gives the compatibility condition between ρ_L and ξ_L:

$$\rho_L = \frac{\rho_c}{2}\left(1 - \frac{\xi_L}{u_f}\right).$$

For example, the function in figure 9.9 in the (x,t) half plane plotted with characteristics is solution of the equation (9.12). The ξ-varying part of the solution is delimited by two constant states ρ_L and ρ_R. Such a solution is called a rarefaction wave or a rarefaction fan. Because of

$$\phi(\xi_R) - \phi(\xi_L) = \int_{\xi_L}^{\xi_R} \phi'(\xi)\, d\xi,$$

we have the compatibility condition

$$\rho_R - \rho_L = -\frac{\rho_c}{2u_f}(\xi_R - \xi_L). \tag{9.17}$$

For $\xi_R > \xi_L$, we have $\rho_R < \rho_L$. The flow indeed tends to rarefy with accelerating vehicles within the fan.

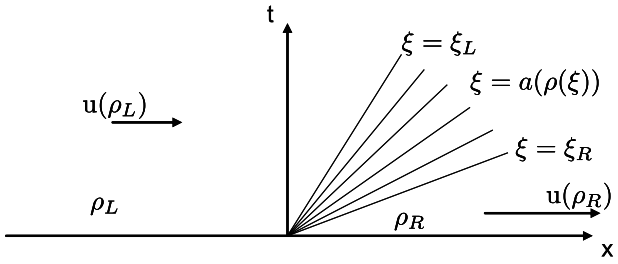


Figure 9.2: A rarefaction fan, autosimilar solution of the traffic equation.

9.2.2 Shock wave discontinuous solution

We are now looking for solutions in the form

$$\rho(x,t) = \phi(x - \sigma t). \tag{9.18}$$

The quantity σ is a wave propagation velocity. Putting the expression (9.18) into (9.7) gives

$$-\sigma\phi'(\xi) + \frac{d}{d\xi}[q(\phi(\xi))] = 0, \quad \xi = x - \sigma t. \tag{9.19}$$

In order to consider discontinuous functions ϕ, we write a weak formulation of (9.19). Let $\xi \mapsto v(\xi)$ a compactly supported smooth function. Then, using an integration by parts, we can write for any $\xi \in \mathbb{R}$,

$$\begin{aligned} 0 &= -\sigma \int_{-\infty}^{\xi} v(s)\phi'(s)\, ds + \int_{-\infty}^{\xi} v(s)\frac{d}{d\xi}[q(\phi(s))]\, ds \\ &= -[\sigma v(s)\phi(s)]_{\infty}^{\xi} + \sigma \int_{-\infty}^{\xi} v'(s)\phi(s)\, ds + [v(s)q(\phi(s))]_{-\infty}^{\xi} - \int_{-\infty}^{\xi} v'(s)q(\phi(s))\, ds \\ &= v(\xi)\left(-\sigma\phi(\xi) + q(\phi(\xi))\right) + \sigma \int_{-\infty}^{\xi} v'(s)\phi(s)\, ds - \int_{-\infty}^{\xi} v'(s)q(\phi(s))\, ds. \end{aligned} \tag{9.20}$$

Now consider a discontinuous weak solution in the form

$$\Phi(\xi) = \rho_L + (\rho_R - \rho_L)\, H(\xi) \tag{9.21}$$

where ρ_L and ρ_R and two constant states and $H(x) = 1_{(x\geq 0)}(x)$ denotes the Heaviside function. Let v be a compactly supported function whose support includes the origin. From the weak formulation we get

$$v(0)\left(-\sigma\phi(0^-) + q(\phi(0^-))\right) + \sigma\int_{-\infty}^{0} v'(s)\phi(s)\,ds - \int_{-\infty}^{0} v'(s)q(\phi(s))\,ds = 0. \tag{9.22}$$

In a similar way, integrating from $\xi = 0$ to $+\infty$ gives

$$-v(0)\left(-\sigma\phi(0^+) + q(\phi(0^+))\right) + \sigma\int_{0}^{\infty} v'(s)\phi(s)\,ds - \int_{0}^{\infty} v'(s)q(\phi(s))\,ds = 0. \tag{9.23}$$

Because we have also

$$-\sigma\int_{\mathbb{R}} v'(s)\phi(s)\,ds + \int_{\mathbb{R}} v'(s)q(\phi(s))\,ds = 0, \tag{9.24}$$

summing up (9.22), (9.23) and (9.24) gives the relations

$$v(0)\left\{-\sigma(\phi(0^+) - \phi(0^-)) + q(\phi(0^+)) - q(\phi(0^-))\right\} = 0 \quad \forall v \in \mathscr{D}(\mathbb{R}).$$

One obtains the well-known Rankine-Hugoniot jump compatibility conditions

$$\sigma(\rho_R - \rho_L) = q(\rho_R) - q(\rho_L) \tag{9.25}$$

often written

$$\sigma\,|[\rho]| = |[q(\rho)]|. \tag{9.26}$$

Remark that due to the parabolic form of q, one can have $q(\rho_L) = q(\rho_R)$ for $\rho_L \neq \rho_R$. In that case,

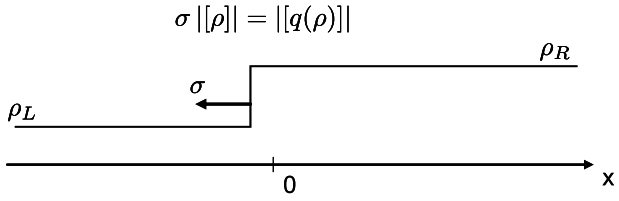


Figure 9.3: A discontinuous shock wave solution. Through the discontinuity, we must satisfy the Rankine-Hugoniot jump compatibility conditions.

the Rankine-Hugoniot relations give $\sigma = 0$. In other words, stationary shock waves do exist.

9.3 Transport equation of a vehicle fraction

There are different kinds of vehicles on a road: car, trucks, motobikes. It can be interesting to track a particular category to known the behavior among all the vehicles. Sometimes, vehicles are forming some platoons and one can be interested in knowing and following the concentration of the platoons. If c denotes the concentration of vehicle of interest, the quantity (ρc) refers to the partial density of vehicles of interest. If there is no distinction of velocity between all the kinds of vehicles, we have of course the conservation law

$$\partial_t(\rho c) + \partial_x(\rho c\, u(\rho)) = 0. \tag{9.27}$$

For solutions of class $\mathscr{C}^1$, equation (9.27) can be expanded as

$$\rho\left\{\partial_t c + u(\rho)\partial_c\right\} + c\left\{\partial_t\rho + \partial_x(\rho u(\rho))\right\} = 0.$$

Thus we get the transport equation on the concentration c:

$$\partial_t c + u(\rho)\,\partial_x c = 0. \tag{9.28}$$

9.4 System of conservation laws

The continuity equation (9.8) combined with the conservation law (9.27) form a system of conservation laws

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = 0 \tag{9.29}$$

with vector state $\boldsymbol{U} = (\rho, \rho c)$ and vector flux $\boldsymbol{F}(\boldsymbol{U}) = (\rho\, u(\rho), \rho c\, u(\rho))$. For smooth solution, it can be written in nonconservation form

$$\partial_t \boldsymbol{U} + \boldsymbol{A}(\boldsymbol{U})\partial_x \boldsymbol{U} = 0 \tag{9.30}$$

where $\boldsymbol{A}(\boldsymbol{U}) = D_U \boldsymbol{F}(\boldsymbol{U})$ is the Jacobian matrix of the flux. It is easy to check that

$$\frac{\partial \rho c u(\rho)}{\partial \rho} = \rho c\, u'(\rho),\ \frac{\partial(\rho c u(\rho))}{\partial(\rho c)} = u(\rho).$$

Thus $\boldsymbol{A}(\boldsymbol{U})$ is equal to

$$\boldsymbol{A}(\boldsymbol{U}) = \begin{pmatrix} a(\rho) & 0 \\ \rho c\, u'(\rho) & u(\rho) \end{pmatrix} \tag{9.31}$$

where $a(\rho) = q'(\rho) = u(\rho) + \rho u'(\rho)$. If $u'(\rho) = 0$, then $\boldsymbol{A}(\boldsymbol{U})$ is in diagonal form and $u(\rho)$ is an eigenvalue of multiplicity 2. Otherwise if $u'(\rho) \neq 0$, then $\boldsymbol{A}(\boldsymbol{U})$ has two distinct eigenvalues $a(\rho)$ and $u(\rho)$ so that $\boldsymbol{A}(\boldsymbol{U})$ is diagonalizable in $\mathbb{R}$. A system in the form (9.29) such that the Jacobian matrix is diagonalizable in $\mathbb{R}$ is called a hyperbolic system of conservation laws. The eigenvalues of the Jacobian matrix define the characteristic velocities of the system. Here, we have $\lambda_1(\boldsymbol{U}) = a(\rho)$ and $\lambda_2(U) = u(\rho)$. The first eigenvector is equal to $\boldsymbol{r}_1(\boldsymbol{U}) = (1, c)$ and the second one is $\boldsymbol{r}_2(\boldsymbol{U}) = (0, 1)$.

Denoting by $\boldsymbol{R}(\boldsymbol{U})$ the eigenvector matrix, i.e. $\boldsymbol{R}(\boldsymbol{U}) = col(\boldsymbol{r}_1(\boldsymbol{U}), \boldsymbol{r}_2(\boldsymbol{U}))$ and

$$\boldsymbol{\Lambda}(\boldsymbol{U}) = diag(\lambda_1(\boldsymbol{U}), \lambda_2(\boldsymbol{U})),$$

we have

$$\boldsymbol{A}(\boldsymbol{U}) = \boldsymbol{R}(\boldsymbol{U})\boldsymbol{\Lambda}(\boldsymbol{U})\boldsymbol{R}^{-1}(\boldsymbol{U}). \tag{9.32}$$

9.5 Finite difference methods for nonlinear transport equations

Here, we need to design both robust and stable numerical methods, able to converge to (possibly) discontinuous weak solutions. In Chapter 2, we have introduced the upwind scheme for linear transport equations. A good candidate here would be extend the upwind conservative schemes to the case of nonlinear transport equations.

For that, we are going to locally linearize the traffic equation at each computational cell interface. Let consider a uniform spatial discretization with nodes $x_j = jh$, $j \in \mathbb{Z}$ and cells $I_j = (x_{j-1/2}, x_{j+1/2})$, $x_{j+1/2} = (j+1/2)h$. If on the interval (x_j, x_{j+1}), between instants t^n and t^{n+1} the traffic equation is linearized into

$$\partial_t \rho + a^n_{j+1/2}\, \partial_x \rho = 0, \quad x \in (x_j, x_{j+1}),\ t \in (t^n, t^{n+1}), \tag{9.33}$$

for some mean propagation velocity, then the upwind scheme naturally writes

$$\rho^{n+1}_j = \rho^n_j - \frac{\Delta t^n}{h}\left((a^n_{j-1/2})^+(\rho^n_j - \rho^n_{j-1}) + (a^n_{j+1/2})^-(\rho^n_{j+1} - \rho^n_j)\right) \tag{9.34}$$

with the notation $x^+ = \max(x, 0)$ and $x^- = \min(x, 0)$. Unfortunately, the nonlinear upwind scheme (9.34) is not in conservation form, making its irrelevant in most cases for discontinuous solutions with incapability to correctly predict the shock propagation velocity at the discrete level. There is only a particular choice where the upwind scheme (9.34) can be written in conservation form. The following theorem enlightens the right choice of averages:

Theorem 4. *If the average $a^n_{j+1/2}$ is chosen such that*

$$a^n_{j+1/2} = \begin{cases} a(\rho^n_j) & \text{if } \rho^n_j = \rho^n_{j+1}, \\ \dfrac{q(\rho^n_{j+1}) - q(\rho^n_j)}{\rho^n_{j+1} - \rho^n_j} & \text{otherwise.} \end{cases} \tag{9.35}$$

then the upwind scheme (9.34) has the conservative form

$$\rho^{n+1}_j = \rho^n_j - \frac{\Delta t^n}{h}\left(\Phi^n_{j+1/2} - \Phi^n_{j-1/2}\right) \tag{9.36}$$

with consistent numerical flux

$$\Phi^n_{j+1/2} = \frac{q(\rho^n_j) + q(\rho^n_{j+1})}{2} - \frac{1}{2}|a^n_{j+1/2}|\,(\rho^n_{j+1} - \rho^n_j). \tag{9.37}$$

The average in (9.35) is called a Roe average and the numerical scheme (9.37) is called the Roe scheme. The Roe scheme has ℓ^1, ℓ^2 and ℓ^∞ stability properties under the Courant-Friedrichs-Lewy condition (CFL condition)

$$\frac{\Delta t^n}{h} \sup_{j\in\mathbb{Z}} |a^n_{j+1/2}| \le 1. \tag{9.38}$$

The proof is let to the reader as exercise. Remark that the Roe average formula can be put in mirror with the Rankine-Hugoniot jump conditions. Indeed from (9.35) we have

$$a^n_{j+1/2}\,(\rho^n_{j+1} - \rho^n_j) = q(\rho^n_{j+1}) - q(\rho^n_j).$$

All happens as if the linearized problem were adapt its local propagation velocity to a shock wave velocity and as if all the waves were replaced by shock waves in the linearized problem.

9.5.1 Nonlinear extension of the Lax-Wendroff scheme

As in the linear case, it appears that the Roe scheme is only first order accurate. It is natural to look for a second order accurate conservative scheme. To reach second order accuracy in time, we start from the Taylor expansion

$$\rho(x_j, t^{n+1}) = \rho(x_j, t^n) + \Delta t^n \partial_t \rho(x_j, t^n) + \frac{(\Delta t^n)^2}{2} \partial_{tt}^2 \rho(x_j, t^n) + O((\Delta t^n)^3), \tag{9.39}$$

Time partial derivatives are then replaced by spatial derivatives:

$$\partial_t \rho = -\partial_x q(\rho), \quad \partial_{tt}^2 \rho = -\partial_{xt}^2 q(\rho) = -\partial_x \left(a(\rho)\partial_t \rho\right) = \partial_x \left(a^2(\rho)\partial_x \rho\right)$$

Thus,

$$\frac{\rho(x_j, t^{n+1}) - \rho(x_j, t^n)}{\Delta t^n} - \frac{\Delta t^n}{2} \partial_x \left(a^2(\rho)\partial_x \rho\right)(x_j, t^n) = \partial_t \rho(x_j, t^n) + O((\Delta t^n)^2). \tag{9.40}$$

The following spatial discretization for the second term still preserves the second order accuracy

$$\frac{\rho_j^{n+1} - \rho_j^n}{\Delta t^n} - \frac{\Delta t^n}{2} \frac{(a_{j+1/2}^n)^2(\rho_{j+1}^n - \rho_j^n) - (a_{j-1/2}^n)^2(\rho_j^n - \rho_{j-1}^n)}{h^2} \approx \partial_t \rho(x_j, t^n) \tag{9.41}$$

where $a_{j+1/2}^n$ is a second order accurate approximation of $a(x_{j+1/2}, t^n)$. For spatial derivatives, a centered formula is simply used:

$$(\partial_x q(\rho))(x_j, t^n) \approx \frac{q(\rho_{j+1}^n) - q(\rho_{j-1}^n)}{2h}. \tag{9.42}$$

The use of the formulae (9.41),(9.42) gives the nonlinear Lax-Wendroff scheme. It can be written in conservation form (exercise)

$$\rho_j^{n+1} = \rho_j^n - \lambda^n \left(\Phi_{j+1/2}^n - \Phi_{j-1/2}^n\right) \tag{9.43}$$

with the Lax-Wendroff numerical flux

$$\Phi_{j+1/2}^n = \frac{q(\rho_j^n) + q(\rho_{j+1}^n)}{2} - \frac{1}{2}\lambda^n (a_{j+1/2}^n)^2 (\rho_{j+1}^n - \rho_j^n) \tag{9.44}$$

with $\lambda^n = \Delta t^n / h$.

9.6 Nonlinear Lax-Friedrichs scheme, hybrid scheme

With similar arguments, it can be shown that the nonlinear extension of the Lax-Friedrichs scheme is a conservative scheme with numerical flux

$$\Phi_{j+1/2}^n = \frac{q(\rho_j^n) + q(\rho_{j+1}^n)}{2} - \frac{1}{2\lambda^n}(\rho_{j+1}^n - \rho_j^n). \tag{9.45}$$

The Roe scheme, the Lax-Wendroff scheme and the Lax-Friedrichs scheme have a common structure. All three can be written

$$\Phi_{j+1/2}^n = \frac{q(\rho_j^n) + q(\rho_{j+1}^n)}{2} - \frac{1}{2}\left(\lambda^n |a_{j+1/2}^n|\right)^\theta |a_{j+1/2}^n| (\rho_{j+1}^n - \rho_j^n) \tag{9.46}$$

with $\theta = -1$ for the Lax-Friedrichs scheme, $\theta = 0$ for the upwind Roe scheme and $\theta = 1$ for the Lax-Wendroff scheme. For any $\theta \in [-1, 1]$, expression (9.46) defines a hybrid interpolated numerical flux.

Of course, there are many other ways to interpolate these three schemes. For example, one could have used

$$\Phi^n_{j+1/2} = \frac{q(\rho^n_j) + q(\rho^n_{j+1})}{2} - \frac{1}{2\lambda^n}\left((\lambda^n|a^n_{j+1/2}|)^{1+\theta^+} - \theta^-(1-\lambda^n|a^n_{j+1/2}|)\right)(\rho^n_{j+1} - \rho^n_j) \quad (9.47)$$

with notations $\theta^+ = \max(\theta, 0)$ and $\theta^- = \min(\theta, 0)$. Actually, the numerical flux (9.47) is preferred to (9.46) because it is Lipschitz continuous. On the other hand the numerical flux (9.46) has an infinite derivative when $a \to 0$ for $\theta \in (-1, 0]$, what can lead to numerical instabilities for low moving characteristics. Another deeper reason to prefer expression (9.47) is that the numerical method has a discrete entropy property for $\theta \in [-1, 0)$ which is not the case for (9.46). This topic is beyond the goal of this course and is not much more detailed.

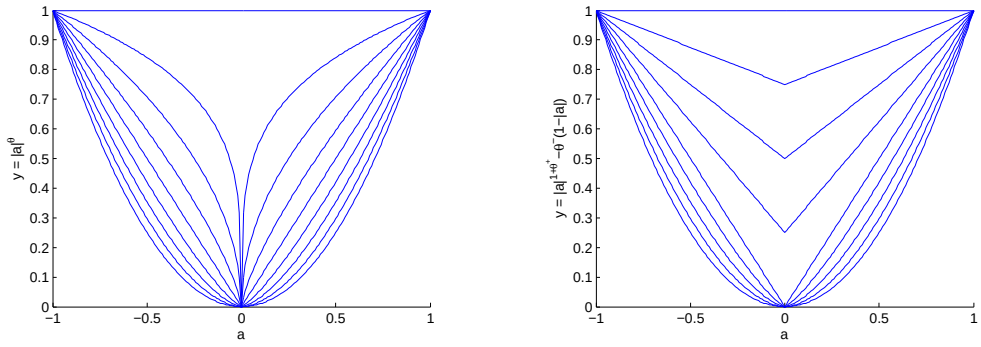


Figure 9.4: Comparison of the interpolation functions in expressions (9.46) and (9.47).

9.7 Numerical scheme for the transport equation of a vehicle fraction

Formerly, we have talk about the transport equation of the fraction of a given class of vehicles:

$$\partial_t c + u(\rho)\partial_x c = 0. \quad (9.48)$$

Actually, it is preferable to work on a conservative form of (9.48) which is equivalent to (9.48) for smooth solutions, as already seen:

$$\partial_t(\rho c) + \partial_x(\rho c u(\rho)) = 0. \quad (9.49)$$

In particular, one can write Rankine-Hugoniot jump conditions on (9.49) whereas it is impossible on the nonconservative equation (9.48).

The quantity (ρc) is the partial density of vehicles of the class. Because equation (9.49) is in conservation form (the number of vehicle of the class of interest is conserved), it is natural to look for conservative schemes for numerical discretization to keep the conservation property at the discrete level. The numerical scheme has to respect some expected properties like the fact that the fraction variable c is a quantity evolving within the interval $[0, 1]$. We have to build a numerical scheme which has a discrete local monotonicity property or a discrete maximum principle property. These properties are sufficient conditions in order to ensure the discrete sequences $(c_j^n)_{j\in\mathbb{Z}}$ to stay into $[0, 1]$ if the initial sequences $(c_j^0)_{j\in\mathbb{Z}}$ has values into $[0, 1]$.

From the previous numerical time-advance scheme for ρ:

$$\rho_j^{n+1} = \rho_j^n - \lambda^n \left(\Phi_{j+1/2}^n - \Phi_{j-1/2}^n\right) \tag{9.50}$$

with $\Phi_{j+1/2}^n$ given for example by (9.47), we look for a conservative discretization of equation (9.49) in the form

$$(\rho c)_j^{n+1} = (\rho c)_j^n - \lambda^n \left(\Psi_{j+1/2}^n - \Psi_{j-1/2}^n\right). \tag{9.51}$$

Once $(\rho c_j)^{n+1}$ is computed, one can compute c_j^{n+1} as

$$c_j^{n+1} = \frac{(\rho c)_j^{n+1}}{\rho_j^{n+1}}. \tag{9.52}$$

The difficulty is to find a convenient numerical flux which guarantees c so stay in $[0, 1]$ at the discrete level. The numerical flux $\Psi_{j+1/2}^n$ must be consistent with the physical flux $\Psi = \rho c u(\rho)$. For stability purpose, we decide to upwind the numerical flux $\Psi_{j+1/2}^n$ according to the sign of the total mass flux $\Phi_{j+1/2}^n$. This means

$$\Psi_{j+1/2}^n = \begin{cases} c_j^n \, \Phi_{j+1/2}^n & \text{if } \Phi_{j+1/2}^n \geq 0, \\ c_{j+1}^n \, \Phi_{j+1/2}^n & \text{if } \Phi_{j-1/2}^n < 0. \end{cases} \tag{9.53}$$

Remark that expression (9.53) can be written in condensed form

$$\Psi_{j+1/2}^n = c_j^n \, \max(0, \Phi_{j+1/2}^n) + c_{j+1}^n \, \min(0, \Phi_{j+1/2}^n) \tag{9.54}$$

or again

$$\Psi_{j+1/2}^n = \frac{c_j^n + c_{j+1}^n}{2} \Phi_{j+1/2}^n - \frac{1}{2} |\Phi_{j+1/2}^n| (c_{j+1}^n - c_j^n). \tag{9.55}$$

Because $\Phi_{j+1/2}^n$is consistent with the physical flux $q(\rho) = \rho \, u(\rho)$, it is clear that $\Psi_{j+1/2}^n$ is consistent with the flux $\Psi = \rho c u(\rho)$.

Let us show that the leading scheme has the expected properties under some conditions. Develop-

ing (9.51) with (9.55) gives

$$\begin{aligned} c_j^{n+1}\,\rho_j^{n+1} &= \rho_j^n c_j^n - \lambda^n \left(\Phi_{j+1/2}^n \frac{c_j^n + c_{j+1}}{2} - \frac{1}{2}|\Phi_{j+1/2}^n|(c_{j+1}^n - c_j^n) \right. \\ &\qquad \left. -\Phi_{j-1/2}^n \frac{c_{j-1}^n + c_j^n}{2} + \frac{1}{2}|\Phi_{j-1/2}^n|(c_j^n - c_{j-1}^n) \right) \\ &= c_j^n \left\{ \rho_j^n - \lambda^n(\Phi_{j+1/2}^n - \Phi_{j-1/2}^n) \right\} \\ &\qquad -\frac{\lambda^n}{2}(|\Phi_{j+1/2}^n| - \Phi_{j+1/2}^n + |\Phi_{j-1/2}^n| + \Phi_{j-1/2}^n)\, c_j^n \\ &\qquad +\frac{\lambda^n}{2}(|\Phi_{j+1/2}^n| - \Phi_{j+1/2}^n)\, c_{j+1}^n \\ &\qquad +\frac{\lambda^n}{2}|\Phi_{j-1/2}^n| + \Phi_{j-1/2}^n)\, c_{j-1}^n \\ &= \rho_j^{n+1}\, c_j^n - \frac{\lambda^n}{2}(|\Phi_{j+1/2}^n| - \Phi_{j+1/2}^n + |\Phi_{j-1/2}^n| + \Phi_{j-1/2}^n)\, c_j^n \\ &\qquad +\frac{\lambda^n}{2}(|\Phi_{j+1/2}^n| - \Phi_{j+1/2}^n)\, c_{j+1}^n \\ &\qquad +\frac{\lambda^n}{2}|\Phi_{j-1/2}^n| + \Phi_{j-1/2}^n)\, c_{j-1}^n. \end{aligned}$$

Thus we see that the numerical scheme can be written in incremental form

$$c_j^{n+1} = (1 - \alpha_j^n - \beta_j^n)c_j^n + \alpha_j^n c_{j-1}^n + \beta_j^n c_{j+1}^n \tag{9.56}$$

where α_j^n et β_j^n can be easily extracted from the previous development. In order to get the discrete maximum principle property, we need to ensure that the coefficients α_j^n and β_j^n belong to $[0,1]$. Assuming that the numerical time-advance scheme for ρ has the positivity property (meaning that $\rho_j^n > 0\ \forall j,n$), we directly have $\alpha_j^n, \beta_j^n \geq 0$. The other bound is satisfied according to the new CFL-like condition

$$\lambda^n \sup_{j\in\mathbb{Z}} \frac{|\Phi_{j+1/2}^n|}{\rho_j^{n+1}} \leq \frac{1}{2}. \tag{9.57}$$

A drawback of the CFL condition (9.57) is that the expression is not completely explicit (λ^n is needed to compute both ρ_j^{n+1} and $\Phi_{j+1/2}^n$). In practice, it is observed that a standard CFL condition less than $1/2$ gives the expected numerical properties.

Let us emphasize that the construction above is important numerically speaking. Without taking care of the discrete maximum principle property, some commonly used numerical schemes can create overshoots or undershoots with violation of the admissible interval $[0,1]$.

9.8 Numerical experiment

9.8.1 `Scilab` source code

```
// Traffic .sce (Scilab)
// Whitham first order model of traffic flow
// Periodic boundary conditions
//
function q = qrho(rho)
// Traffic parameters
  ufree = 110;
  rhoc = 400;
  q = ufree * rho .* (1-rho/rhoc);
endfunction;
//
ufree = 110;
rhoc = 400;
N = 200;
cfl = 0.5;
theta = -0.1; // Hybridation parameter
h = 1 / N;
x = h/2 : h: 1-h/2;
rho  = zeros(1,N);
phi  = zeros(1,N+1);
aroe = zeros(1,N+1);
//
// Initial data
rho = 0.4*rhoc * (1 + 0.4*sin(6*%pi*x));
clf(); plot(x, rho, '.-');
//
for it=1:100
  q = qrho(rho);
  drho = [rho,rho(1)] - [rho(N),rho];
  dq  = [q,q(1)] - [q(N),q];
  K = find(abs(drho)>=1e-5);
  L = find(abs(drho)<1e-5);
  aroe(K) = dq(K) ./ drho(K);
  aroe(L) = ufree*(1-2*rho(L)/rhoc);
  rl = cfl / max(abs(aroe));
  phi = 0.5*([q,q(1)]+[q(N),q]) ...
     - 0.5*((rl*abs(aroe)+1e-5).^theta) .* abs(aroe) .* drho;
  rho = rho - rl * ( phi(2:N+1) - phi(1:N) );
  drawlater();
  clf();
  u = ufree * (1-rho/rhoc);
  subplot(3,1,1), plot(x, rho, '.-');
  xgrid(); xtitle("Vehicle density [Nb/km]");
  subplot(3,1,2), plot(x, u , '.-');
  xgrid(); xtitle("Vehicle speed [km/h]");
  subplot(3,1,3), plot(x, rho .* u , '.-');
  xgrid(); xtitle("Vehicle flow rate [Nb/h]");
  drawnow();
end;
```

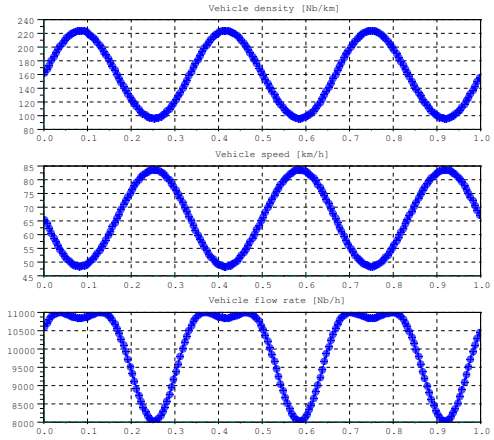


Figure 9.5: Numerical solution using a CFL number equal to 0.5 and $\theta = -0.1$, after one time iterations.

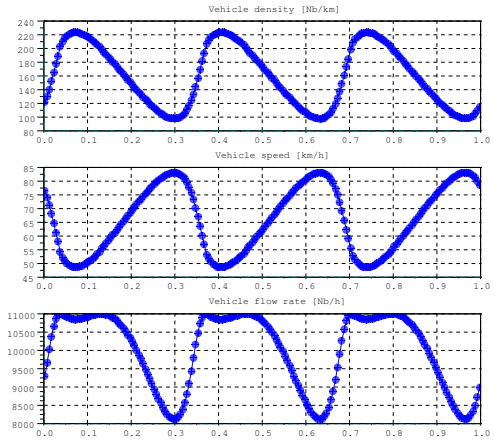


Figure 9.6: Numerical solution using a CFL number equal to 0.5 and $\theta = -0.1$, after 20 time iterations.

9.8.2 One-dimensional numerical results

9.9 Pseudo two-dimensional model for lane connection modeling

The heuristic multidimensional extension of the traffic flow equation with a diffusion term writes

$$\partial_t \rho + \nabla \cdot (\rho\, u(\rho) \boldsymbol{n}) - \nu \Delta \rho = 0. \tag{9.58}$$

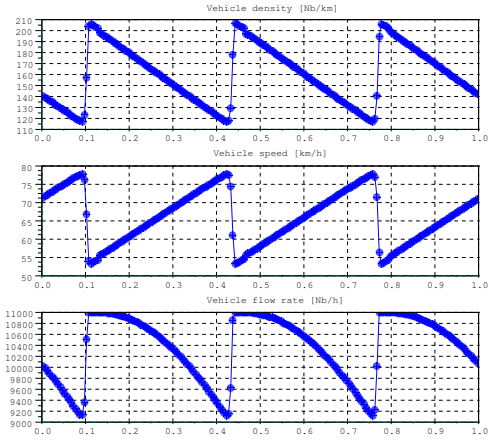


Figure 9.7: Numerical solution using a CFL number equal to 0.5 and $\theta = -0.1$, after 100 time iterations. Several shock waves appear. The numerical scheme is able to capture the discontinuities.

9.9.1 `freefem++` source code for the viscous two-dimensional model

```
// Traffic .edp (Freefem++)rho
// Peudo one-dimension vehicle trafic flow with road branching
// Model \ partial_t \rho+\nabla(rho V(rho) n)-\nu\Delta\rho=0.
// V(rho) = q(rho)/rho with q(rho) = rho * Vfree*(1-rho/rhoc)
//
real [int] A(2), B(2), C(2), D(2), E(2);
real [int] F(2), G(2), H(2), I(2), J(2);
real [int] K(2), L(2), M(2), Ng(2);
A = [0,0];  B = [9, 1.5]; C = [12, 1.5]; D = [21,0];
E = [6, 2]; F = [15, 2]; G = [0, 4]; H = [9, 2.5];
I = [12, 2.5]; J = [21, 4]; K = [0, 1]; L = [0, 3];
M = [21, 1]; Ng = [21, 3];
border c1(t=0,1){x=(1-t)*A[0]+t*B[0]; y=(1-t)*A[1]+t*B[1];}
border c2(t=0,1){x=(1-t)*B[0]+t*C[0]; y=(1-t)*B[1]+t*C[1];}
border c3(t=0,1){x=(1-t)*C[0]+t*D[0]; y=(1-t)*C[1]+t*D[1];}
border c4(t=0,1){x=(1-t)*G[0]+t*L[0]; y=(1-t)*G[1]+t*L[1];}
border c5(t=0,1){x=(1-t)*L[0]+t*E[0]; y=(1-t)*L[1]+t*E[1];}
border c6(t=0,1){x=(1-t)*E[0]+t*K[0]; y=(1-t)*E[1]+t*K[1];}
border c7(t=0,1){x=(1-t)*K[0]+t*A[0]; y=(1-t)*K[1]+t*A[1];}
border c8(t=0,1){x=(1-t)*D[0]+t*M[0]; y=(1-t)*D[1]+t*M[1];}
border c9(t=0,1){x=(1-t)*M[0]+t*F[0]; y=(1-t)*M[1]+t*F[1];}
border c10(t=0,1){x=(1-t)*F[0]+t*Ng[0]; y=(1-t)*F[1]+t*Ng[1];}
border c11(t=0,1){x=(1-t)*Ng[0]+t*J[0]; y=(1-t)*Ng[1]+t*J[1];}
border c12(t=0,1){x=(1-t)*J[0]+t*I[0]; y=(1-t)*J[1]+t*I[1];}
border c13(t=0,1){x=(1-t)*I[0]+t*H[0]; y=(1-t)*I[1]+t*H[1];}
border c14(t=0,1){x=(1-t)*H[0]+t*G[0]; y=(1-t)*H[1]+t*G[1];}
//
```

```
mesh Th = buildmesh(c1(60)+c2(30)+c3(60)
+c4(6)+c5(50)+c6(50)+c7(6)
+c8(6)+c9(50)+c10(50)+c11(6)
+c12(60)+c13(30)+c14(60) );
plot(Th, ps = "mesh.eps");
//
fespace Vh(Th, P1);
Vh rho, rhoold, sigma, sigmaold, rhoh, sigmah, p, ph;
Vh n1, n2, u1, u2, u1visu, u2visu, vrho;
fespace Wh(Th, P2);
Wh phi, phih;
//
real rho0 = 25; // [nb cars / km]
real rhoc = 300; // critical density
real vfree = 110; // [km / h]
real nu = 10; // viscosity
real dt = 0.004; // time step
real t = 0; // current time
// Initial field
rho    = rho0;
rhoold = rho0;

// Step 0. Define a velocity unit vector by solving an
// independent Laplace problem, then get the unit
// vector of the gradient of the solution
//
problem Laplace(phi, phih) =
  int2d(Th)( dx(phi)*dx(phih) + dy(phi)*dy(phih))
  +int1d(Th, c4)(phih) + int1d(Th, c7)(phih)
  +on(c8, c11, phi=0);
Laplace;
n1 = dx(phi)/sqrt(dx(phi)^2+dy(phi)^2);
n2 = dy(phi)/sqrt(dx(phi)^2+dy(phi)^2);
vrho = vfree * (1-rho / rhoc);
//
u1 = vrho*n1;
u2 = vrho*n2;
//
problem step1(rho, rhoh) =
  int2d(Th)( rho*rhoh/dt )
  -int2d(Th)( convect([u1,u2], -dt, rhoold)*rhoh/dt )
  +int2d(Th)( nu*dx(rho)*dx(rhoh) + nu*dy(rho)*dy(rhoh) )
  +int2d(Th)(rho*dx(u1)*rhoh+rho*dy(u2)*rhoh)
  +on(c4, rho=rho0)
  +on(c7, rho=rhoc);

for (int it=0 ; it<200 ; it++) {
  t += dt;
  step1;
  rhoold = rho;
  vrho = vfree * (1 - rho / rhoc);
  u1 = vrho * n1;
```

```
80   u2 = vrho * n2;
81   plot(rho, nbiso=50, fill=1, value=1, wait=0);
82  }
83  cout << "Final time = " << t << endl;
84  plot(rho, nbiso=40, fill=1, value=1, ps="rho.eps");
85  u1visu = u1/vfree;
86  u2visu = u2/vfree;
87  plot([u1visu, u2visu], value=1, ps="speed.eps");
```

9.9.2 Numerical results

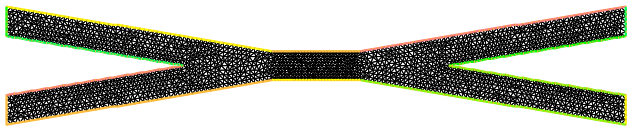


Figure 9.8: Computational mesh

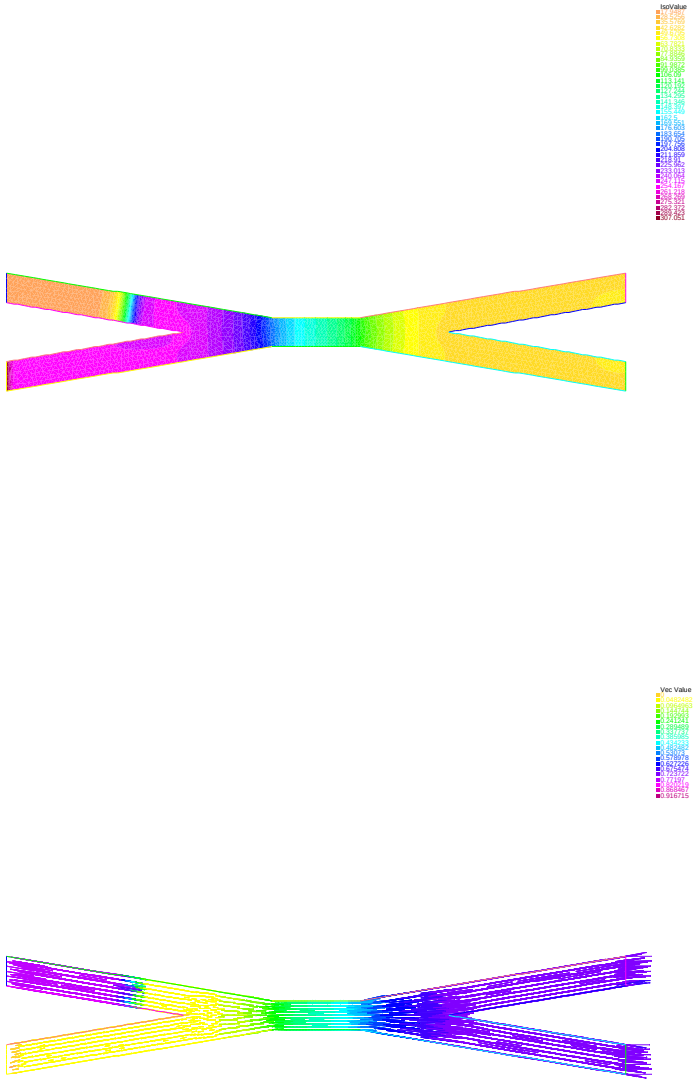


Figure 9.9: Density contour levels and velocity field at a given instant. One can observe a backward propagating traveling wave at the top left lane, revealing the downstream bottleneck.

Chapter 10

Biological cell migration and proliferation

Modeling of biological cell migration and proliferation is of importance for the understanding of diseases like cancer (tumour growth, metastases migration, tissue invasion, etc.). When the cell density is large enough, the continuous medium assumption is a good approximation and partial differential equations can be written.

10.1 Biological and mathematical requirements

Biological expectations and some expected mathematical properties lead to the following requirements. In what follows, we are going to derive the "simplest" mathematical model able to fulfill these requirements.

1. Without proliferation and apoptosis (cell death), the number of cell has to be conserved. So migration phenomenon should be modeled by a conservation law.

2. Travelling waves and sharp cell fronts are observed in biological experiments. Thus this behaviour must be reproduced by the model.

3. Sometimes cell fronts reach a steady state. That means that the cell fronts slow down and stop in finite time, revealing a cell region with a boundary (think about animal skin marks and patterns for example). This behaviour also has to be reproduced by the model.

4. Known biochemical factors like chemoattractant and chemorepellent agents are able to attract or repel biological cells.

5. Cell motility is the ability for a cell to freely move, generally with a brownian motion. From the macroscopic point of view, this is a diffusive phenomenon. If the diffusion is isotropic, then the diffusion operator is the Laplace operator.

6. There are biological regulation factors that limit the cell density up to a certain threshold.

10.2 Guidelines for PDE modeling

As already seen in a previous chapter, the Keller-Segel model already takes into account cell motility (diffusion), proliferation (source terms) and propagation due to the presence of a chemoattractant c. Moreover, the Keller-Segel system is in conservation form.

But he Keller-Segel cannot reproduce traveling waves or sharp cell moving fronts. So a modification of these equations or the adding of a new modeling term is necessary.

In traffic flow modeling we have seen that a nonlinear flux term in the equation can create unsteady or steady discontinuities (shock waves). So the idea is the replace the Keller-Segel convection term

$$\nabla \cdot (\alpha \rho \nabla c)$$

by

$$\nabla \cdot (q(\rho) \nabla c) \tag{10.1}$$

for a nonlinear concave function $q : [0, +\infty) \to \mathbb{R}$. Thus, we are looking for a mathematical model in the form

$$\partial_t \rho - \nu \Delta \rho + \nabla \cdot (q(\rho) \nabla c) = r \rho (\rho_\infty - \rho), \tag{10.2}$$

$$\partial_t c - \Delta c = s(\frac{\rho}{\rho_\infty} - c) \tag{10.3}$$

where $\nu > 0$ is the diffusion rate, $r > 0$ is a proliferation rate and $\rho_\infty > 0$ is the threshold cell density. The quantity c is the concentration of chemoattractant (or chemorepellent according to the sign of $q(\rho)$). The parameter $s > 0$ is a reaction rate for c. The convective flux for the cells is

$$\boldsymbol{J} = q(\rho) \nabla c. \tag{10.4}$$

Next step, we need a closure for $q(\rho)$. The first constraint is $q(0) = 0$ (no flux is there is no cell). In the direction $\boldsymbol{n} = \frac{\nabla c}{||\nabla c||}$, there is a flux

$$j = \boldsymbol{J} \cdot \boldsymbol{n} = q(\rho) \, ||\nabla c||.$$

The flux can be designed in order to attract cells located in low density regions toward denser regions (clustering) and to repel cells located in dense regions in order to colonize free regions (migration). One can consider for example a strictly concave function $q(\rho)$ such that $q(0) = 0$ and $q(\rho_c) = 0$ for some $\rho_c \in (0, \rho_\infty]$. One can consider for example a polynomial of degree 2 crossing the two points:

$$q(\rho) = \alpha \, \rho \left(1 - \frac{\rho}{\rho_c}\right) \tag{10.5}$$

with some constant $\alpha > 0$. Practically, the constants α, ρ_c and ρ_∞ should be chosen according to some biological considerations and by identification from measurements. Figure 10.1 shows a candidate function $q(\rho)$. The characteristic velocity for the convective term is

$$\boldsymbol{v} = q'(\rho) \, \nabla c. \tag{10.6}$$

In the case of (10.5), one gets

$$\boldsymbol{v} = \alpha \left(1 - 2 \, \frac{\rho}{\rho_c}\right) \nabla c. \tag{10.7}$$

There is two ways to get a null characteristic velocity leading in that case to a locally stationary wave. Either $\nabla c = 0$ or $q'(\rho) = 0$.

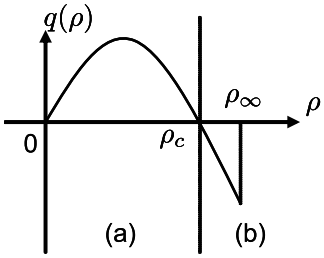


Figure 10.1: Function $\rho \mapsto q(rho)$. For $\rho \in [0, \rho_c]$, the flux is positive and the chemical species acts as a chemoattractant [region(a)]. For $\rho \in [\rho_c, \rho_\infty]$, the flux is negative and the chemical species with acts as a chemorepellent [region (b)].

10.3 Numerical results

10.3.1 `freefem++` source code

```
//
// MigrationNN.edp (Freefem++)
// Migration/ Proliferation model – florian de vuyst
//
real Lx = 3;
real Ly = 2;
real dt = 0.05;
real uf = 1;
real rhoc = 100;
mesh Th = square(60, 40, [x*Lx,y*Ly]);
fespace Uh(Th, P2, periodic=[[3,x],[1,x]]);
fespace Vh(Th, P1, periodic=[[3,x],[1,x]]);
Uh rho, lrho, rhoold, rhoh, c, ch, cold;
Vh u1, u2, u, n1, n2, v1, v2, rhop1, cp1;
rho = rhoc*exp( -40*(x-Lx/4)^2 - 40*(y-Ly/2)^2 )
  + rhoc*exp( -40*(x-3*Lx/4)^2 - 40*(y-Ly/2)^2 )
  + rhoc*exp( -40*(x-0.55*Lx)^2 - 40*(y-Ly/2)^2 );
Th = adaptmesh(Th, rho, periodic=[[3,x],[1,x]]);
plot(Th);
rho = rho;
c = rho/rhoc;
rhoold = rho;
cold = c;
// plot(rho, nbiso=50, wait=1);
//
problem migr([rho, c], [rhoh, ch]) =
  int2d(Th)(rho*rhoh/dt)
  -int2d(Th)(convect([v1,v2], -dt,rhoold)*rhoh/dt)
  +int2d(Th)(dx(v1)*rho*rhoh+dy(v2)*rho*rhoh)
  +int2d(Th)(0.01*dx(rho)*dx(rhoh)+0.01*dy(rho)*dy(rhoh))
  -int2d(Th)(0.01*rho*(rhoc-rhoold)*rhoh)
```

```
  +int2d(Th)(c*ch/dt)
  -int2d(Th)(cold*ch/dt)
  +int2d(Th)(dx(c)*dx(ch)+dy(c)*dy(ch))
  -int2d(Th)(10*(rho/rhoc-c)*ch);
//
for (int it=0; it<20; it++) {
for (int substep=0; substep<2; substep++){
  u1 = -dx(cold);
  u2 = -dy(cold);
  v1 = 0.5 * u1 * rhoold/rhoc;
  v2 = 0.5 * u2 * rhoold/rhoc;
  migr;
  Th = adaptmesh(Th, rho, periodic=[[3,x],[1,x]]);
  rho=rho;
  c=c;
  rhoold = rho;
  cold = c;
}
// Visu
rhop1 = rho; cp1 = c;
plot(Th, rhop1, nbiso=50, fill=0, value=1,
      ps="migr_it="+it+".eps");
}
cout << "Done.\n";
```

10.3.2 Numerical results

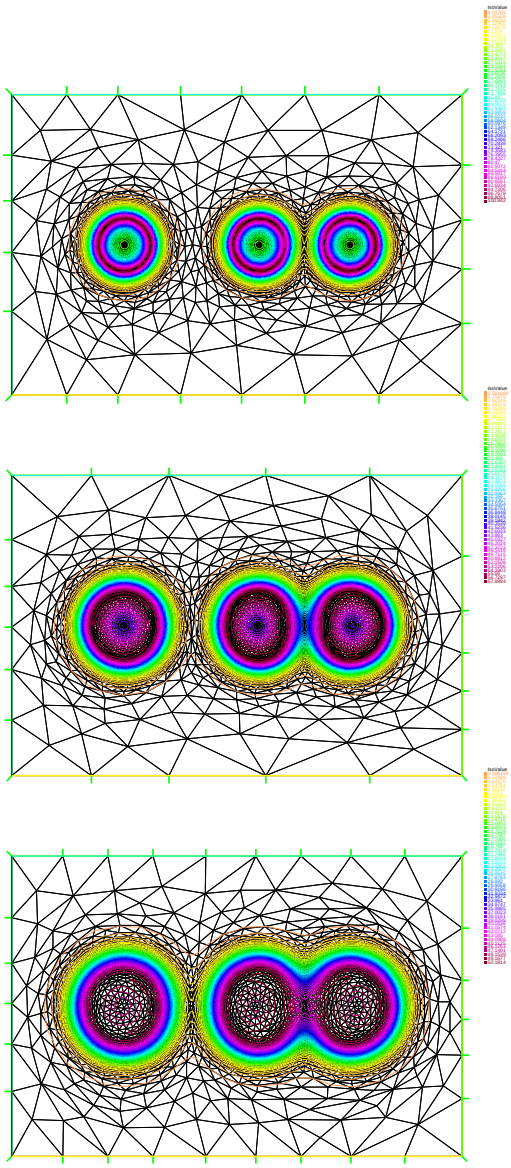


Figure 10.2: Cell density contour levels at different instants. Initially there are three cell sites. During migration and proliferation, the sites are growing and merging. One can see the sharp cell front moving in the medium.

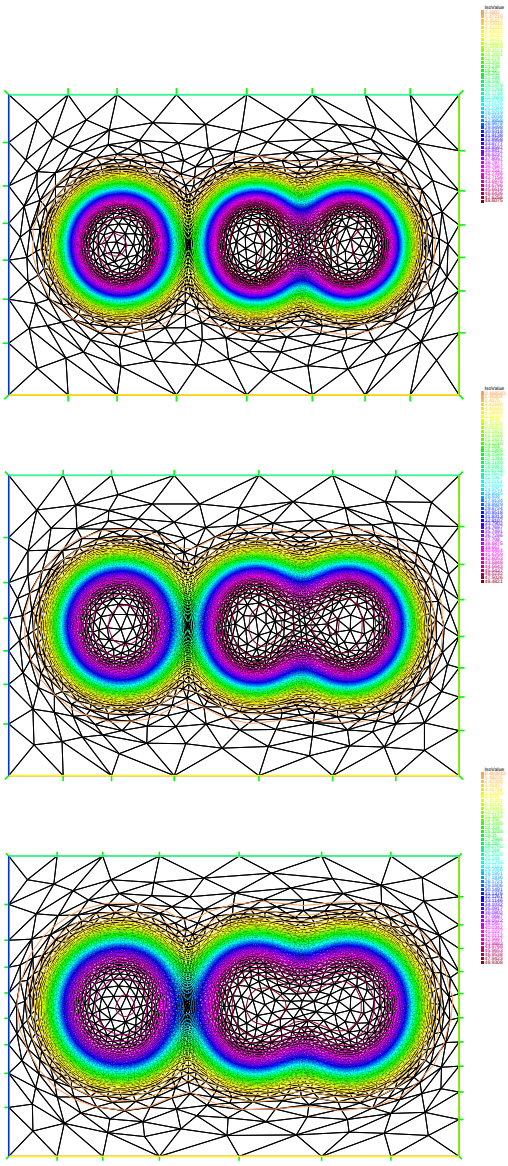


Figure 10.3: Cell density contour levels at different instants. Initially there are three cell sites. During migration and proliferation, the sites are growing and merging. One can see the sharp cell front moving in the medium.

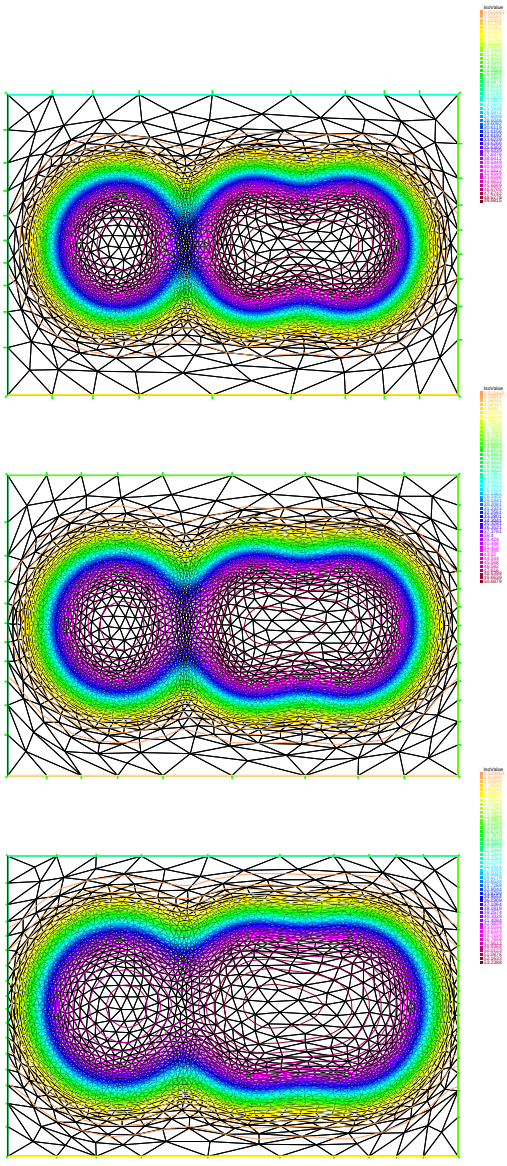


Figure 10.4: Cell density contour levels at different instants. Initially there are three cell sites. During migration and proliferation, the sites are growing and merging. One can see the sharp cell front moving in the medium.

Chapter 11

Gas Dynamics

When a fluid is considered incompressible and inviscid and is not subjected to the effect of external forces, the governing equations are the so-called compressible Euler equations. They are made of the continuity equation that expresses the conservation of the mass

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{11.1}$$

(ρ is the fluid density, $\boldsymbol{u}$ is the velocity), the equation of conservation of the momentum

$$\partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = 0 \tag{11.2}$$

(p is the pressure of the fluid) and the equation of the conservation of the energy

$$\partial_t (\rho E) + \nabla \cdot ((\rho E + p)\boldsymbol{u}) = 0 \tag{11.3}$$

where E is the specific total energy made of the kinetic energy $u^2/2$ ($u = ||\boldsymbol{u}||$) and the internal energy e:

$$E = \frac{u^2}{2} + e \tag{11.4}$$

The Euler system (11.1)-(11.3) is a rather complex system. It is known that solutions can develop discontinuity (shock waves) even if the initial data is of arbitrary regularity. This possible loss of regularity has strong implications on theoretical numerical analysis and the proper way to discretize those equations. Best numerical methods for the Euler equations are conservative upwind Finite Volume methods.

Despite we here adopt the method of Characteristics + FE strategy. For numerical experiments, we will focus on a supersonic flow around an elliptic body. We will see that the numerical solutions are not so bad and anyway give a rather good information on the features of the supersonic flow (separated shock, compression shock, etc.). So the `freefem++` environment is a good candidate a have a first sight on those king of solutions before going further in more sophisticated conservative numerical methods. In order to implement a FE-like program for the Euler equations, we need to prepare the equations in a suited form. For the continuity equation, remember that using the total derivative, for sufficiently smooth solutions one can write

$$D_t \rho + \rho \nabla \cdot \boldsymbol{u} = 0 \tag{11.5}$$

or equivalently

$$D_t a_\rho + \nabla \cdot \boldsymbol{u} = 0 \tag{11.6}$$

using the new variable $a_\rho = \log(\rho)$. Equation (11.6) is interesting because it is linear with respect to the variables a_ρ and $\boldsymbol{u}$. Now, let us consider the momentum equation. Using the standard Einstein's mute indexes, it can be rewritten

$$\partial_t(\rho u_i) + (\rho u_i u_j)_{,j} + p_{,i} = 0, \quad i = 1, \ldots, d.$$

From the continuity equation, $\partial_t \rho = -(\rho u_j)_{,j}$, it is easy to obtain

$$\partial_t u_i + u_j u_{i,j} + \frac{1}{\rho} p_{,i} = 0$$

or in vector form

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \nabla p = \boldsymbol{0} \tag{11.7}$$

or again

$$D_t \boldsymbol{u} + \frac{1}{\rho} \nabla p = \boldsymbol{0}. \tag{11.8}$$

By the same approach, one can obtain the equation on E

$$D_t E + \frac{1}{\rho} \nabla \cdot (p\boldsymbol{u}) = 0. \tag{11.9}$$

One can also successively obtain from (11.4) and (11.8)

$$\begin{aligned} 0 &= D_t e + \frac{1}{2} D_t(u^2) + \frac{1}{\rho} \nabla \cdot (p\boldsymbol{u}) \\ &= D_t e + \boldsymbol{u} \cdot D_t \boldsymbol{u} + \frac{1}{\rho} \nabla \cdot (p\boldsymbol{u}) \\ &= D_t e - \frac{1}{\rho} \boldsymbol{u} \cdot \nabla p + \frac{1}{\rho} \nabla \cdot (p\boldsymbol{u}) \\ &= D_t e + \frac{p}{\rho} \nabla \cdot \boldsymbol{u} \end{aligned} \tag{11.10}$$

which gives an equation on for the internal energy.

11.1 Perfect gas

The perfect gas closure is usually

$$e = \frac{1}{\gamma - 1} \frac{p}{\rho} \tag{11.11}$$

where γ is the ratio of the (constant here) specific heats c_p and c_v. The ratio γ is equal to

$$\gamma = 1.4 \tag{11.12}$$

for a diatomic gas. Let us write an equation for the pressure. From equations (11.5) and (11.10), one has

$$D_t(\rho e) + (\rho e + p) \nabla \cdot \boldsymbol{u} = 0, \tag{11.13}$$

then

$$D_t p + \gamma p \nabla \cdot \boldsymbol{u} = 0. \tag{11.14}$$

from (11.11). It appears again interesting to introduced the new variable $a_p = \log(p)$ to get the linear equation

$$D_t a_p + \gamma \nabla \cdot \boldsymbol{u} = 0. \tag{11.15}$$

So far we have written the original system (11.1)-(11.3) in the equivalent form (for smooth solutions)

$$D_t a_\rho + \nabla \cdot \boldsymbol{u} = 0, \tag{11.16}$$

$$D_t \boldsymbol{u} + \frac{1}{\rho} \nabla p = 0, \tag{11.17}$$

$$D_t a_p + \gamma \nabla \cdot \boldsymbol{u} = 0. \tag{11.18}$$

A nonlinear term remains in the equation, namely $\frac{1}{\rho}\nabla p$. For practical reasons appearing in the numerical approach, we will rather the following equivalent script of equation (11.17):

$$\frac{1}{T} D_t \boldsymbol{u} + \nabla a_p = 0 \tag{11.19}$$

where $T = p/\rho$ is a temperature.

11.2 Discretization in time

Let Δt be a time step. The total derivatives are discretized according to the method of characteristics:

$$\frac{a_\rho^{n+1} - a_\rho^n \circ X^n}{\Delta t} + \nabla \cdot \boldsymbol{u}^{n+1} = 0, \tag{11.20}$$

$$\frac{1}{T^n} \frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n \circ X^n}{\Delta t} + \nabla a_\rho^{n+1} = 0, \tag{11.21}$$

$$\frac{a_p^{n+1} - a_p^n \circ X^n}{\Delta t} + \gamma \nabla \cdot \boldsymbol{u}^{n+1} = 0. \tag{11.22}$$

Some comments are necessary: first remark that for stability purposes, the numerical scheme is made implicit. The divergence and gradient terms are systematically taken as implicit. In this way, the three equations are fully coupled. One can notice that the term $1/T$ is taken explicit in order to keep a linear system. The time discretization is said to be semi-implicit. In this way, the problem (11.20)-(11.22) of unknowns $(a_\rho^{n+1}, \boldsymbol{u}^{n+1}, a_p^{n+1})$ becomes linear.

11.3 Full discretization

To complete the discretization, we use the Finite Element formalism for spatial discretization. Due to the fact that we have a system of PDEs, we need a test function for each equation (four in two space dimensions).

Let us denote v the test function for the $a\rho$-equation, v_1 and v_2 the test functions for each component of the velocity and w the test function for the a_p-equation. The variational formulation reads

$$\int_\Omega \frac{a_\rho^{n+1} - a_\rho^n \circ X^n}{\Delta t} v \, d\boldsymbol{x} + \int_\Omega \nabla \cdot \boldsymbol{u}^{n+1} v \, d\boldsymbol{x}$$
$$+ \int_\Omega \frac{1}{T^n} \frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n \circ X^n}{\Delta t} \cdot (v_1, v_2) \, d\boldsymbol{x} + \int_\Omega \nabla a_\rho^{n+1} \cdot (v_1, v_2) \, d\boldsymbol{x}$$
$$+ \int_\Omega \frac{a_p^{n+1} - a_p^n \circ X^n}{\Delta t} w \, d\boldsymbol{x} + \gamma \int_\Omega \nabla \cdot \boldsymbol{u}^{n+1} \, w \, d\boldsymbol{x} = 0 \quad \forall v, v_1, v_2, w \in V^h. \quad (11.23)$$

11.4 Numerical experiments

11.4.1 `freefem++` source code of the supersonic flow problem around an ellipse

```
// Program schockellipse.edp (freefem++)
// Supersonic perfect gas flow around an ellipse
// Be careful : nonconservative formulation of the
// compressible Euler equations.
// A conservative Finite Volume method should be used
// Author : florian.de-vuyst@ecp.fr
//
real gamma = 1.4;
real pinf = 1e2;
real rhoinf = 0.3;
real cinf = sqrt(gamma*pinf/rhoinf); // speed of sound
real uinf = 1.5 * cinf; // infinite flow Mach number is 2
real radius = 10; // Radius of the infinite flow boundary
real alpha = 0.3; // Attack angle (radian)
real x0 = -4; // x-translation of the body
real ra = 2, rb = 0.3; // Features of the body ellipse
real dt = 0.01; // Time step
//
// External infinite boundary is a circle
border binf(t=0,2*pi){x=radius*cos(t); y=radius*sin(t);}
// Body is an ellipse
border wall(t=0,2*pi){x=x0+cos(alpha)*ra*cos(t)+sin(alpha)*rb*sin(t);
  y=-sin(alpha)*ra*cos(t)+cos(alpha)*rb*sin(t);}
mesh Th = buildmesh(binf(100)+wall(-80));
plot(Th, wait=0, ps="Th.eps");
//
fespace Vh(Th, P1); // FE space
Vh rho, p, arho, ap, T, u1, u2; // unknowns
Vh rhoold, pold, Told, arhoold, apold, u1old, u2old;
  // fields at former instant
Vh v, v1, v2, w; // test functions
//
// Field initialization
rho = rhoinf; p = pinf;
u1 = uinf; u2 = 0;
```

```
arho = log(rho); ap = log(p);
T    = p/rho; // temperature-like
plot(T, nbiso=50, fill=1, value=1, wait=0);
// Go for the partial differential problem
problem euler([arho, u1, u2, ap], [v, v1, v2, w]) =
  int2d(Th)(arho*v/dt)
  -int2d(Th)(convect([u1old,u2old], -dt, arhoold)*v/dt)
       +int1d(Th, binf)((uinf*N.x)*v)
       -int2d(Th)(u1*dx(v)+u2*dy(v))
  +int2d(Th)(u1*v1/Told/dt)
  -int2d(Th)(convect([u1old,u2old], -dt, u1old)*v1/Told/dt)
  +int2d(Th)(u2*v2/Told/dt)
  -int2d(Th)(convect([u1old,u2old], -dt, u2old)*v2/Told/dt)
  +int2d(Th)(dx(ap)*v1+dy(ap)*v2)
  +int2d(Th)(ap*w/dt)
  -int2d(Th)(convect([u1old,u2old], -dt, apold)*w/dt)
  +int1d(Th, binf)(gamma*(uinf*N.x)*w)
       -int2d(Th)(gamma*u1*dx(w)+gamma*u2*dy(w));

for (int it=0; it<80; it++) {
  u1old = u1; u2old = u2;
  arhoold = arho;
       apold = ap;
  Told = T;
  euler;
  if (it>70){
  Th = adaptmesh(Th,ap);
  u1=u1; u2=u2; arho=arho; ap=ap;
  }
  rho = exp(arho);
  p = exp(ap);
  T = p/rho;
  plot(rho, nbiso=50, fill=0, value=1);
  if (it<20) {
  plot(rho, nbiso=50, fill=1, value=1, ps="rho_mach1.5_it"+it+".eps");
  }
}
plot(T, nbiso=50, fill=1, value=1, ps="T.eps");
plot(T, nbiso=50, fill=0, value=1, ps="Tiso.eps");
plot(p, nbiso=50, fill=1, value=1, ps="p.eps");
plot(rho, nbiso=50, fill=1, value=1, ps="rho.eps");
plot([u1,u2], ps="velocity.eps");
plot(Th, ps="finalmesh.eps");
```

11.4.2 Numerical results at infinite Mach number equal to 1.5

11.4.3 Numerical results at infinite Mach number equal to 4

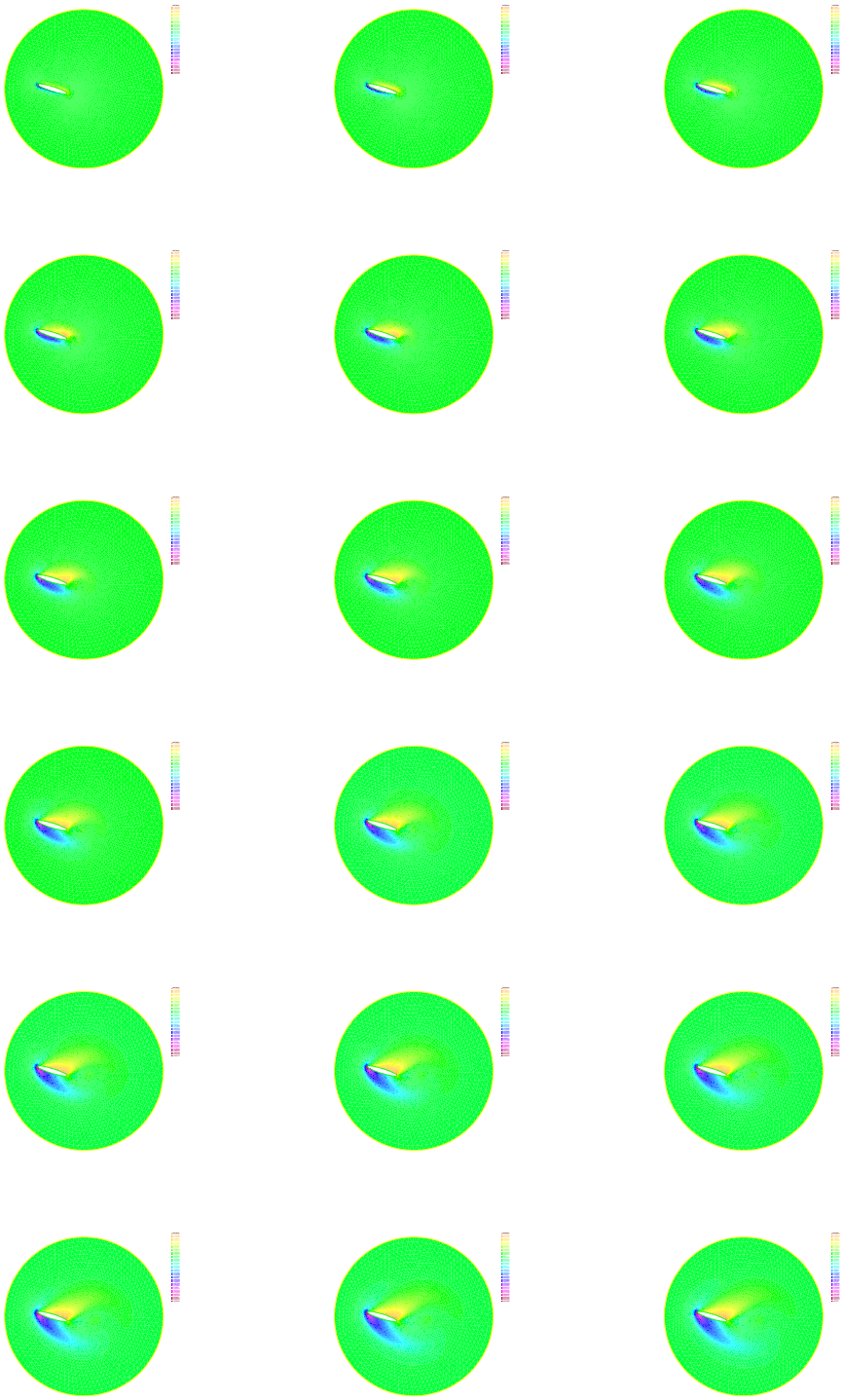


Figure 11.1: Transient phase of the flow from the initial uniform flow. Isocontours of density for the 18 first time iterations.

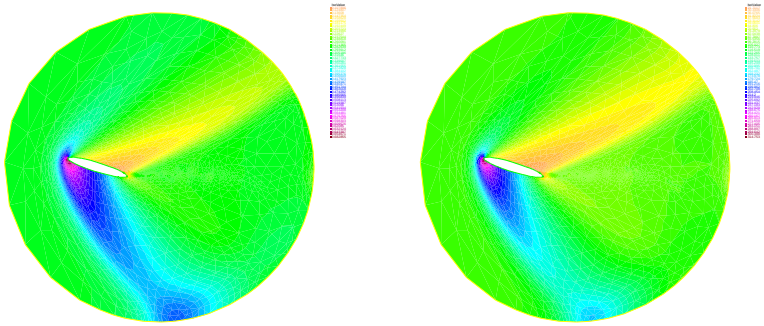


Figure 11.2: Isocontours of density and pressure for the steady state solution.

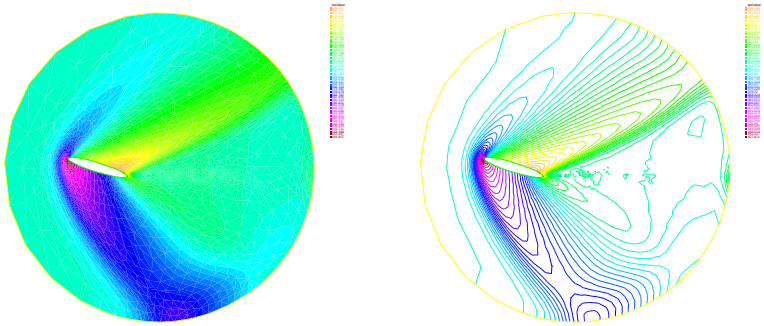


Figure 11.3: Isocontours of the temperature for the steady state solution.

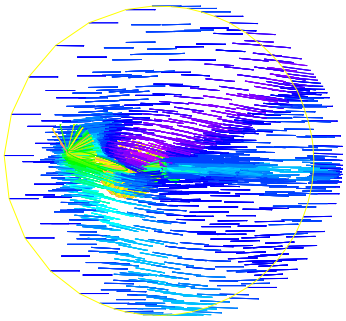


Figure 11.4: Velocity field for the steady state.

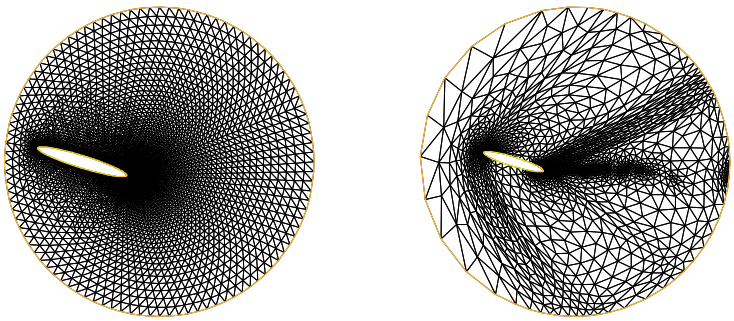


Figure 11.5: Triangular initial and final mesh at the end of computation. The `freefem++` command `adaptmesh` was used for mesh refinement.

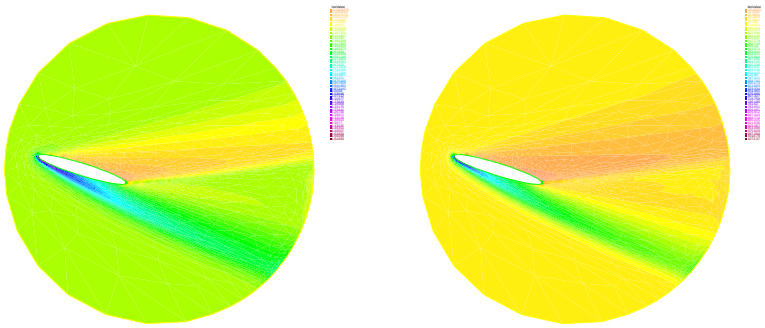


Figure 11.6: Isocontours of density and pressure for the steady state solution with infinite Mach number equal to 4.

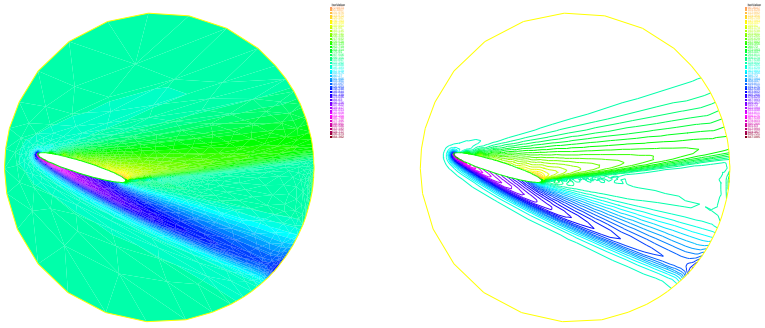


Figure 11.7: Isocontours of the temperature for the steady state solution with infinite Mach number equal to 4.

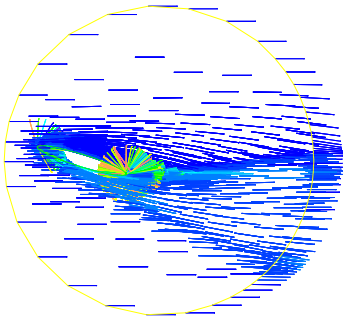


Figure 11.8: Velocity field for the steady state with infinite Mach number equal to 4.

Chapter 12

Fluid Mechanics and heat transfer

Fluid Mechanics and Heat Transfer are of fundamental interest for the engineering of energy conversion systems. Main of the energy conversion system use a working fluid that transports the heat. This is the case the nuclear power system for refrigerators, cooling devices, air conditioning, etc.

As an example, we will consider here a compressible fluid that flows in a heating pipe. We will suppose the fluid as weakly compressible with a density that depends on the temperature $\rho = \rho(\theta)$ with

$$\frac{\partial \rho}{\partial \theta} < 0 \tag{12.1}$$

meaning that the fluid is lighter when is it heated. Then we will take into account the gravity. Due to Archimedes' buoyancy principle, the lighter heated fluid will tend to go upward with appearance of Rayleigh-Taylor instabilities at the bottom thermal layer. That's we want to investigate by the numerical simulation.

12.1 Model assumptions

Because the fluid is assumed to be weakly compressible, we keep the zero-divergence assumption on the velocity field, i.e.

$$\nabla \cdot \boldsymbol{u} = 0. \tag{12.2}$$

The fluid is supposed to be Newtonian, the momentum equation then gives the standard Navier-Stokes equations with gravity as external volume force:

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \frac{1}{\rho(\theta)} \nabla p = \boldsymbol{g} \tag{12.3}$$

where ν is the kinematic viscosity of the fluid. The fluid is supposed to be thermally conductive with constant coefficient κ. The balance equation of energy can be written as an equation of evolution of the temperature. It can be reasonably approximated as a convection-diffusion equation

$$\rho\, c_p \partial_t \theta + \rho c_p \boldsymbol{u} \cdot \nabla \theta - \nabla \cdot (\kappa \nabla \theta) = 0 \tag{12.4}$$

where c_p is the specific heat at constant pressure. For low compressible fluids, the density in (12.5) is approximated by a constant ρ_0.

$$\rho_0\, c_p \partial_t \theta + \rho_0 c_p \boldsymbol{u} \cdot \nabla \theta - \nabla \cdot (\kappa \nabla \theta) = 0 \tag{12.5}$$

On the other hand, we keep the dependency of ρ on θ in (12.3) even if density variations are small in order to evidence the buoyant force due to lighter heated fluid. So we need a closure for $\theta \mapsto \rho(\theta)$. We will use the linearized law on the specific volume $\tau = 1/\rho$:

$$\tau = \tau_0 + \beta(\theta - \theta_0) \tag{12.6}$$

for some constants $\tau_0,\ \theta_0$ and $\beta > 0$. Using (12.6), for $\beta > 0$, the inequality (1.1) is satisfied.

12.1.1 Dimensionless equations

It is always interesting to derive dimensionless equations for parameter analysis and systematic study of all kind of solutions. From the engineering point of view, dimensionless numbers are of main importance for system design. We will consider the symbol $\star$ for dimensionless variables and 0 index for dimensioning constants. Considering,

$$t = \frac{t^\star}{T_0},\ \boldsymbol{x} = \frac{\boldsymbol{x}^\star}{L_0},\ \boldsymbol{u} = \frac{\boldsymbol{u}^\star}{U_0},\ \nu = \frac{\nu^\star}{\nu_0},\ \rho = \frac{\rho^\star}{\rho_0},\ \boldsymbol{g} = \frac{\boldsymbol{g}^\star}{g_0},\ \kappa = \frac{\kappa^\star}{\kappa_0},$$

from (12.2),(12.3) and (12.4) we respectively get

$$\begin{aligned}
&\frac{U_0}{L_0}\,\nabla^\star \cdot \boldsymbol{u}^\star = 0,\\
&\frac{U_0}{T_0}\partial_{t^\star}\boldsymbol{u}^\star + \frac{U_0}{L_0^2}\boldsymbol{u}^\star \cdot \nabla^\star \boldsymbol{u}^\star - \frac{\nu_0 U_0}{L_0^2}\Delta^\star \boldsymbol{u}^\star + \frac{1}{\rho_0\,\rho^\star}\frac{p_0}{L_0}\nabla^\star p^\star = g_0\,\boldsymbol{g}^\star,\\
&\rho_0 c_p \frac{\theta_0}{T_0}\partial_{t^\star}\theta^\star + \rho_0 c_p \frac{U_0\theta_0}{L_0}\boldsymbol{u}^\star \cdot \nabla^\star \theta^\star - \frac{\kappa_0\theta_0}{L_0^2}\nabla^\star \cdot (\kappa^\star \nabla^\star \theta^\star) = 0.
\end{aligned}$$

If both time and pressure scales are chosen such that

$$T_0 = \frac{L_0}{U_0},\ p_0 = \rho_0 U_0^2, \tag{12.7}$$

then we get the following equations (forgetting the $\star$ symbol for brevity's sake)

$$\begin{aligned}
&\nabla \cdot \boldsymbol{u} = 0,\\
&\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \frac{\nu_0 U_0}{L_0}\Delta \boldsymbol{u} + \frac{1}{\rho}\nabla p = \frac{L_0 g_0}{U_0^2}\,\boldsymbol{g},\\
&\partial_t \theta + \boldsymbol{u} \cdot \nabla \theta - \frac{\kappa_0}{\rho_0 U_0 c_p L_0}\,\nabla \cdot (\kappa \nabla \theta) = 0.
\end{aligned}$$

There are three dimensionless numbers. The Reynolds number

$$Re = \frac{L_0}{\nu_0 U_0} \tag{12.8}$$

involves the spatial scale, the fluid velocity and the kinematic viscosity. The Prandtl number

$$Pr = \frac{\rho_0 \nu_0 c_p}{\kappa_0} \tag{12.9}$$

shows the importance of thermal conductivity with respect to the other effects. Finally the ratio

$$\frac{L_0 g_0}{U_0} \tag{12.10}$$

gives the importance of the source term scale with respect to the spatial scale and the velocity scale. In what follows, we will consider $g_0 = U_0^2/L_0$ so that the dimensionless equations are written

$$\nabla \cdot \boldsymbol{u} = 0, \tag{12.11}$$

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \frac{1}{Re} \Delta \boldsymbol{u} + \frac{1}{\rho(\theta)} \nabla p = \boldsymbol{g}, \tag{12.12}$$

$$\partial_t \theta + \boldsymbol{u} \cdot \nabla \theta - \frac{1}{Re} \frac{1}{Pr} \nabla \cdot (\kappa \nabla \theta) = 0 \tag{12.13}$$

or in Lagrangian form

$$\nabla \cdot \boldsymbol{u} = 0, \tag{12.14}$$

$$\frac{D\boldsymbol{u}}{Dt} - \frac{1}{Re} \Delta \boldsymbol{u} + \frac{1}{\rho(\theta)} \nabla p = \boldsymbol{g}, \tag{12.15}$$

$$\frac{D\theta}{Dt} - \frac{1}{Re} \frac{1}{Pr} \nabla \cdot (\kappa \nabla \theta) = 0 \tag{12.16}$$

12.1.2 Boundary conditions

We are interested in a fluid flowing into a duct with a heating wall. We consider a two dimensional geometry as plotted in figure 12.1. The computational domain is a rectangle Ω of respective lengths L_x and L_y. Left and right borders correspond to the inlet and the outlet. The four edges Γ_{in}, Γ_T, Γ_B and Γ_{out} respectively represent the inflow border, the top wall boundary, the bottom wall boundary and the outflow boundary. At the inlet, a constant temperature θ_{in} is imposed. We also consider a steady state Poiseuille flow with parabolic velocity profile $\boldsymbol{u}_{in}$. Let us recall that the Poiseuille flow is the stationary laminar flow between two infinite plates for which one can derive an analytical solution of the Navier-Stokes equations with parabolic velocity profile.

At the outlet Γ_{out}, homogeneous Neumann boundary conditions are imposed for both velocity and temperature. At the top wall boundary Γ_T, a no slip boundary condition with zero velocity is imposed for the fluid. A zero heat flux is also consider (adiabatic wall). At the bottom boundary Γ_B, a no slip boundary condition is imposed for velocity and a constant nonzero heat flux provides the heat to the fluid:

$$\kappa \frac{\partial \theta}{\partial \boldsymbol{n}} = \Phi \quad \text{on } \Gamma_B. \tag{12.17}$$

12.2 Mathematical problem

For initial conditions, the fluid is supposed to flow according to the laminar Poiseuille steady flow. For compatibility between initial condition and inflow we consider the initial condition in velocity

$$\boldsymbol{u} = \boldsymbol{u}_{in} \quad \text{in } \Omega. \tag{12.18}$$

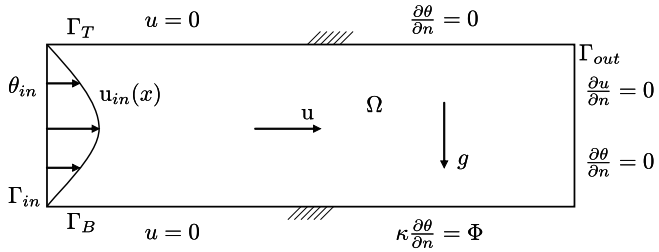


Figure 12.1: Schematic of the spatial domain Ω with boundary conditions

The initial temperature field will be assumed constant, equal to the inflow temperature:

$$\theta = \theta_{in} \quad \text{in } \Omega. \tag{12.19}$$

We summarize here the whole mathematical time-dependent initial boundary value problem:

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega \times (0,T), \tag{12.20}$$

$$D_t \boldsymbol{u} - \frac{1}{Re}\,\Delta \boldsymbol{u} + \frac{1}{\rho(\theta)}\nabla p = \boldsymbol{g} \quad \text{in } \Omega \times (0,T), \tag{12.21}$$

$$D_t \theta - \frac{1}{Re}\,\frac{1}{Pr}\,\nabla \cdot (\kappa \nabla \theta) = 0 \quad \text{in } \Omega \times (0,T), \tag{12.22}$$

$$\boldsymbol{u}(.,t=0) = \boldsymbol{u}_{in} \quad \text{in } \Omega, \tag{12.23}$$

$$\theta(.,t=0) = \theta_{in} \quad \text{in } \Omega, \tag{12.24}$$

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } (\Gamma_T \cup \Gamma_B) \times (0,T), \tag{12.25}$$

$$\boldsymbol{u} = \boldsymbol{u}_{in} \quad \text{on } \Gamma_{in} \times [0,T) \tag{12.26}$$

$$\theta = \theta_{in} \quad \text{on } \Gamma_{in} \times [0,T) \tag{12.27}$$

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = 0 \quad \text{on } \Gamma_{out} \times [0,T), \tag{12.28}$$

$$\frac{\partial \theta}{\partial \boldsymbol{n}} = 0 \quad \text{on } (\Gamma_T \cup \Gamma_{out}) \times [0,T), \tag{12.29}$$

$$\kappa\,\frac{\partial \theta}{\partial \boldsymbol{n}} = \Phi \quad \text{on } \Gamma_B \times [0,T). \tag{12.30}$$

12.3 Numerical discretization

12.3.1 Time discretization

First the partial differential equations are semi-discretized in time. As discussed in previous chapters, total derivatives are discretized thanks to the method of characteristics. The nonlinear term is discretized using a semi-implicit formula whereas the other linear terms are discretized implicitely:

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0 \quad \text{in } \Omega, \tag{12.31}$$

$$\frac{\boldsymbol{u}^{n+1}(\boldsymbol{x}) - \boldsymbol{u}^n \circ X^n(\boldsymbol{x})}{\Delta t} - \frac{1}{Re}\Delta \boldsymbol{u}^{n+1} + \frac{1}{\rho(\theta^n)}\nabla p^{n+1} = \boldsymbol{g} \quad \text{in } \Omega, \tag{12.32}$$

$$\frac{\theta^{n+1}(\boldsymbol{x}) - \theta^n \circ X^n(\boldsymbol{x})}{\Delta t} - \frac{1}{Re}\frac{1}{Pr}\nabla \cdot \left(\kappa \nabla \theta^{n+1}\right) = 0 \quad \text{in } \Omega, \tag{12.33}$$

for $n \in \mathbb{N}$ and

$$\boldsymbol{u}^0(\boldsymbol{x}) = \boldsymbol{u}_{in}(\boldsymbol{x}),\ \theta^0(\boldsymbol{x}) = \theta_{in}, \quad \boldsymbol{x} \in \Omega. \tag{12.34}$$

With such a time discretization, it appears in (12.32) and (12.33) that there is separation of the unknown variables $\boldsymbol{u}^{n+1}(\boldsymbol{x})$ and $\theta^{n+1}(\boldsymbol{x})$. Consequently, between two time steps t^n and t^{n+1}, the Navier-Stokes problem and the thermal problem can be solved separately.

12.3.2 Variational formulation

We are looking for a variational formulation of the problem with equations (12.31)-(12.34) and boundary conditions (12.25) to (12.30). For the equation on temperature, there is no particular difficulty. Considering the functional space

$$V_g = \left\{ v \in H^1(\Omega),\ v = g \text{ on } \Gamma_{in} \right\}, \tag{12.35}$$

the problem is to find a temperature $\theta^{n+1} \in V_{\theta_{in}}$ such that, for all $v \in V_0$,

$$\int_\Omega \frac{\theta^{n+1}(\boldsymbol{x}) - \theta^n \circ X^n(\boldsymbol{x})}{\Delta t} v \, d\boldsymbol{x} + \frac{1}{Re}\frac{1}{Pr}\int_\Omega \kappa \nabla \theta^{n+1} \cdot \nabla v \, d\boldsymbol{x} - \frac{1}{Re}\frac{1}{Pr}\int_{\Gamma_B} \Phi v \, d\boldsymbol{x} = 0. \tag{12.36}$$

12.4 Numerical experiments

12.4.1 `freefem++` source code of the heat transfer problem

```
// ThermalConvection.edp (Freefem++)
// Forced + Natural heat convection in a pipe
// Navier-Stokes equations + convection-diffusion on themperature
//
real lx = 0.25;
real Lx = 3;
real Ly = 1;
real gravity = 9.81;
real thetain = 20;
real pout = 1;
real Cst = 1;
```

```
real uin0 = 0.2;
func uin = uin0 * 4*(y/Ly)*(1-y/Ly);
real HeatFlux = 100;
real dt = 0.1;
real nu = 0.001;
real kappa=0.001;
real t = 0, dtsnap = 0.5, ttosnap = 2;
//
border c1(t=0, lx){x=t; y=0;}
border c2(t=lx, Lx-lx){x=t;y=0;}
border c3(t=Lx-lx,Lx){x=t;y=0;}
border c4(t=0,Ly){x=Lx;y=t;}
border c5(t=Lx,0){x=t; y=Ly;}
border c6(t=Ly,0){x=0;y=t;}
//
mesh Th = buildmesh(c1(10)+c2(200)+c3(10)+c4(30)
+c5(100)+c6(30));
plot(Th, wait=0, ps="Mesh.eps");
//
fespace Uh(Th, P1b);
Uh u, v, uold, vold, uh, vh; // Velocity fields
fespace Xh(Th, P1);
Xh p, ph, theta, thetaold, thetah; // Temperate and pressure fields
Xh tau; // = 1/rho ( specific  volume);
//
// Field  initialization
u = uin;
v = 0;
theta = thetain;
uold = u; vold = v; thetaold = theta;
// Heat problem
problem HeatStep(theta, thetah) =
  int2d(Th) (theta*thetah/dt)
  -int2d(Th)(convect([u,v], -dt, thetaold)*thetah/dt)
  +int2d(Th)(kappa*dx(theta)*dx(thetah)
     +kappa*dy(theta)*dy(thetah))
  -int1d(Th, c2)(kappa*HeatFlux*thetah)
  +on(c6, theta=thetain);
//
problem NavierStokesStep([u, v, p], [uh, vh, ph]) =
  int2d(Th)(u*uh/dt)
  - int2d(Th)(convect([uold, vold], -dt, uold)*uh/dt )
  + int2d(Th)(nu*dx(u)*dx(uh)+nu*dy(u)*dy(uh))
  + int2d(Th) (tau * dx(p)*uh)
  + int2d(Th)(v*vh/dt)
  - int2d(Th)(convect([uold, vold], -dt, vold)*vh/dt )
  + int2d(Th)(nu*dx(v)*dx(vh)+nu*dy(v)*dy(vh))
  + int2d(Th)(tau * dy(p)*vh)
  + int2d(Th)(gravity*vh)
  + int2d(Th)( dx(u)*ph + dy(v)*ph )
  + on(c6, u=uin, v=0)
  + on(c1, c2, c3, c5, u=0, v=0);
```

```
64 // Time steps
65 for (int it=0; it<150; it++) {
66   t = t + dt;
67   HeatStep;
68     thetaold = theta;
69     tau = Cst * theta;
70   NavierStokesStep;
71     uold = u; vold = v;
72   // plot([u, v]);
73   //Th = adaptmesh(Th);
74   //u = u; v = v; uold = uold; vold = vold;
75   // theta = theta; thetaold = thetaold;
76   plot([u,v], theta, nbiso = 80, value=0);
77   if (t>=ttosnap) {
78     ttosnap = ttosnap + dtsnap;
79     plot([u,v], theta, nbiso = 80, value=0,
80       ps="field_t="+t+".eps");
81   }
82 }
```

12.4.2 Numerical results

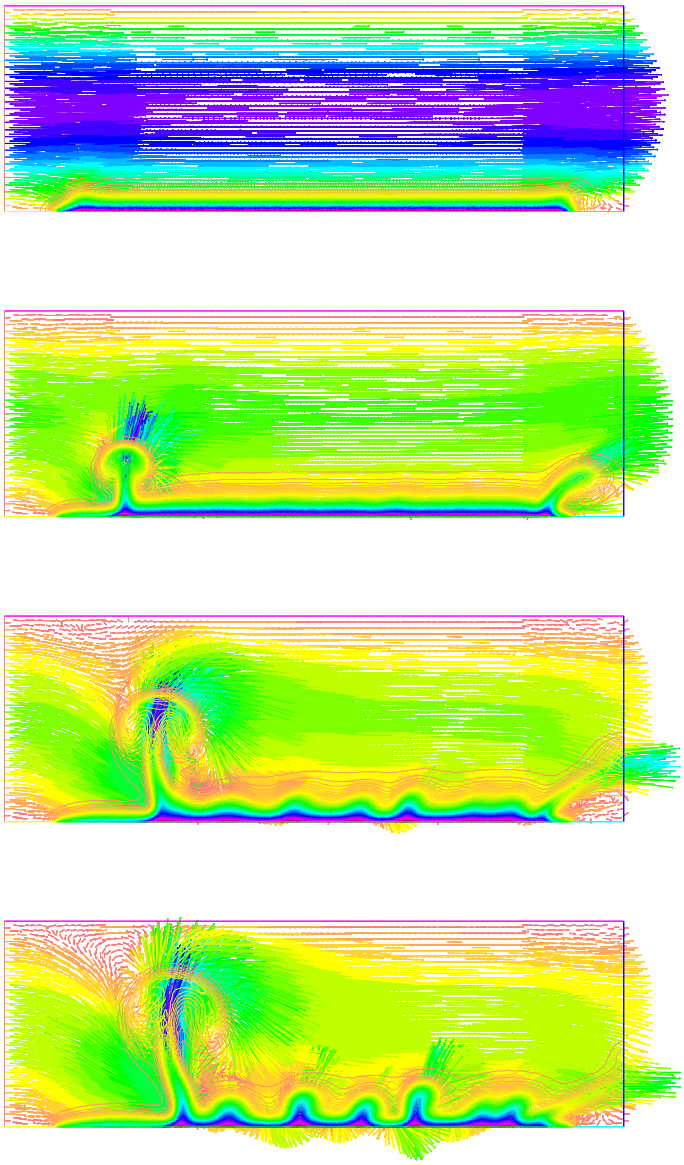


Figure 12.2: Snapshots of the velocity field (arrows) and temperature field (isocontours) at successive instants $t = 2$ s, $t = 4$s, $t = 5.1$s and $t = 5.6$s.

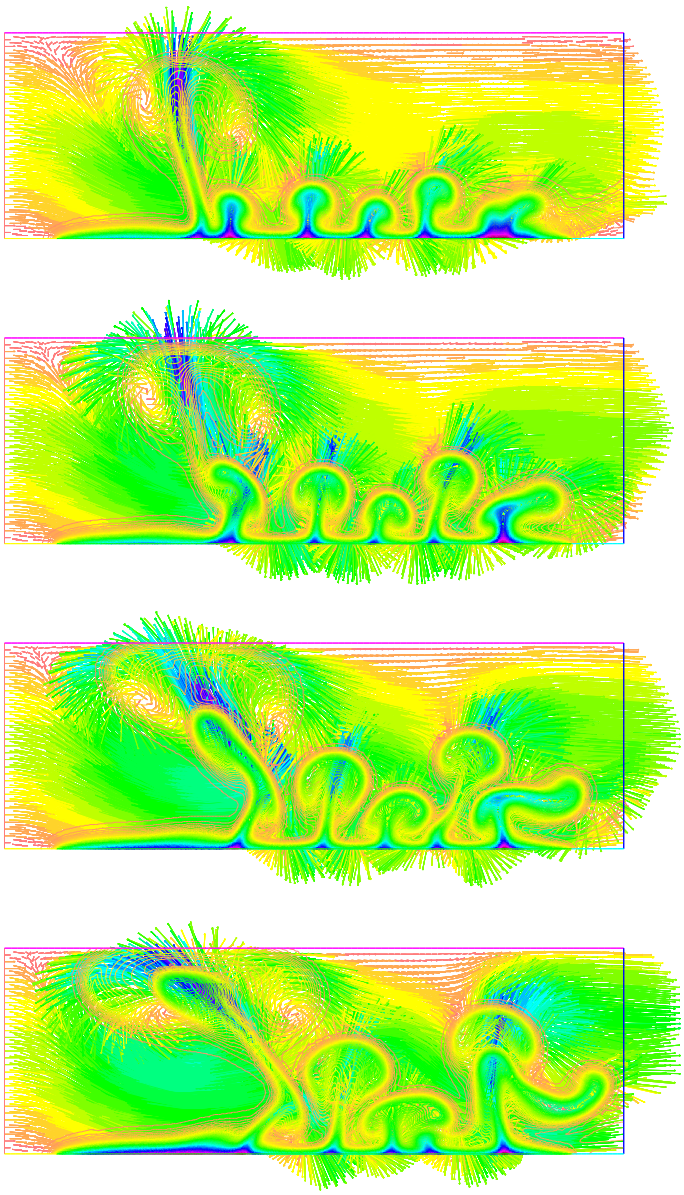


Figure 12.3: Snapshots of the velocity field (arrows) and temperature field (isocontours) at successive instants $t = 6.1$ s, $t = 6.6$s, $t = 7.1$s and $t = 7.6$s.

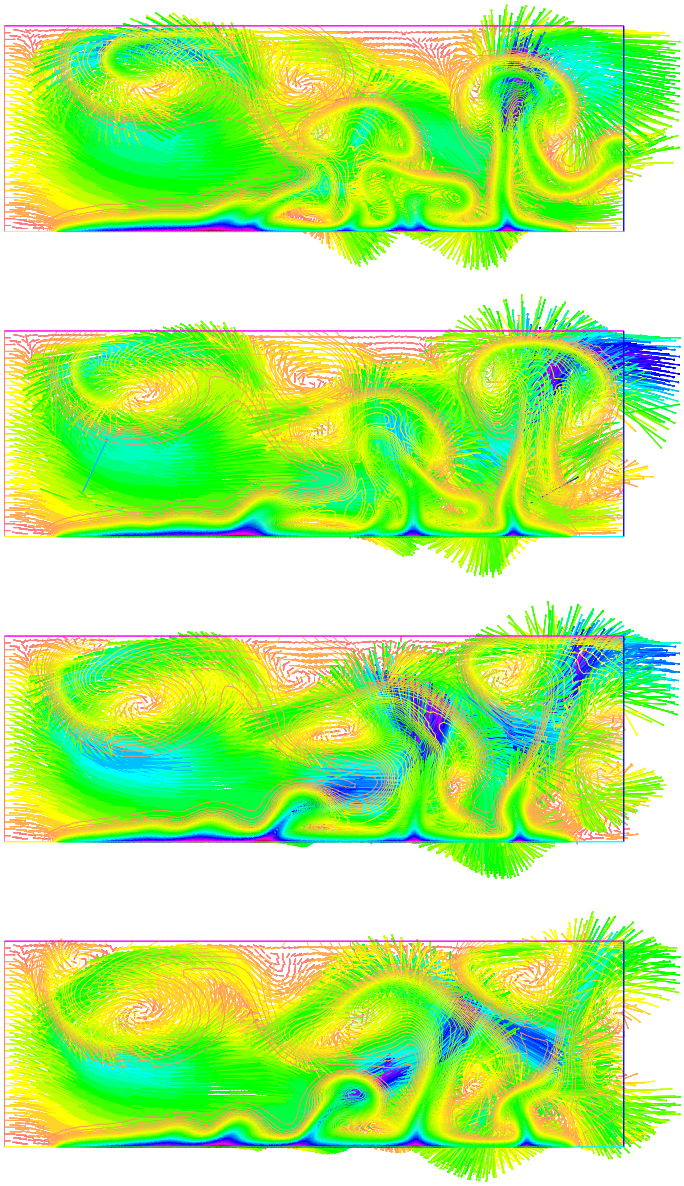


Figure 12.4: Snapshots of the velocity field (arrows) and temperature field (iso-contours) at successive instants $t = 8.1$ s, $t = 8.6$s, $t = 9.1$s and $t = 9.6$s.

Chapter 13

Stochastic diffusion processes, Fokker-Planck equations

This chapter is a short introduction to the stochastic problems. Some physical systems intrinsically include stochastic effects due to the inherent noise. Moreover, for real applications it is usual to have inexact knowledge of initial conditions or boundary conditions. Thus, an initial state can be seen as a random variable. Each integral path can evolve in a different way because of random fluctuations in the system. In the theory of stochastic differential system and Markov diffusion processes, it is shown that the probability density function of the stochastic process is solution of a partial differential equation, namely the Fokker-Planck equation which is a (possibly high-dimensional) convection-diffusion equation. The scope of this chapter is only limited to the computational aspects of the numerical solution of either stochastic differential equations or Fokker-Planck equations. The reader who is interested in the theoretical derivation of the Fokker-Planck equation can refer to the important literature on this subject like [], [] or [].

13.1 Ordinary and Stochastic differential equations

A deterministic dynamical system is written

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u}(\boldsymbol{x}), \tag{13.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^d$ and $\boldsymbol{u} : \mathbb{R}^d \to \mathbb{R}^d$ is a vector field (here supposed not depend on time itself). For well-posedness, we add to (13.1) an initial condition

$$\boldsymbol{x}(t=0) = \boldsymbol{x}^0 \in \mathbb{R}^d \tag{13.2}$$

and suppose that $\boldsymbol{u}$ is Lipschitz-continuous: there exists a Lipschitz constant $L > 0$ such that

$$||\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x}')|| \leq ||\boldsymbol{x} - \boldsymbol{x}'|| \; \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d. \tag{13.3}$$

In the case of a "non completely" known dynamical system, we have to model the uncertainty in some sense. For example the initial state $\boldsymbol{x}^0$ if replaced by a random variable $\boldsymbol{X}$. This random variable is completely defined by its density probability function (or PDF) $p(\boldsymbol{X})$. Of course this needs a closure

subject to some constraints. For example one can expect that

$$E(\boldsymbol{X}) = \int_{\mathbb{R}^d} \boldsymbol{x}\, p(\boldsymbol{x})\, d\boldsymbol{x} = \boldsymbol{x}^0 \tag{13.4}$$

where the expectation value $\boldsymbol{x}^0$ is known and the covariance matrix $\mathbf{C}$ with elements

$$C_{ij}(\boldsymbol{X}) = E\left[(x_i - E(x_i))(x_j - E(x_j))\right] \tag{13.5}$$

is also known.

Now, when the system is subject to some random fluctuations during time, the way to rigorously write the governing equation is a stochastic differential equation in the form

$$d\boldsymbol{X}(t) = \boldsymbol{u}(\boldsymbol{X}(t))\, dt + \boldsymbol{\sigma}(\boldsymbol{X}(t))\, d\boldsymbol{B}(t) \tag{13.6}$$

where $(\boldsymbol{X}(t), t \geq 0)$ now denotes a stochastic diffusion process with drift $\boldsymbol{u}$, $\boldsymbol{B}$ in (13.5) maps $\mathbb{R}^d$ to $\mathscr{M}_{d\times p}$ and $\boldsymbol{B}(t)$ denotes p-dimensional brownian motion. The matrix-valued function $\boldsymbol{\sigma}$ is also supposed to be Lipschitz continuous. Below we will denote $p(\boldsymbol{x}, t)$ the probability density function at state $\boldsymbol{x}$ and time t.

13.2 Fokker-Planck equations

The theory (see for example []) states that p is solution of a partial differential equation:

Theorem 5 (Fokker-Planck equations)**.** *Let us denote by $p_0(\boldsymbol{x})$ the probability density function of the random variable $\boldsymbol{X}$. If the probability density function $p(\boldsymbol{x}, t)$ of the stochastic process $(\boldsymbol{X}(t), t \geq 0)$ has sufficient regularity, then it is solution of the Fokker-Planck problem*

$$\frac{\partial p}{\partial t} + \sum_{j=1}^{d} \frac{\partial}{\partial x_j}\left(u_j(\boldsymbol{x}) p\right) - \frac{1}{2}\sum_{i,j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}\left(a_{ij}(\boldsymbol{x})\, p\right) = 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^n,\ t \geq 0, \tag{13.7}$$

$$p(\boldsymbol{x}, t = 0) = p_0(\boldsymbol{x}) \quad \textit{almost surely in } \boldsymbol{x}, \tag{13.8}$$

where

$$a_{ij}(\boldsymbol{x}) = \sum_{k=1}^{p} \sigma_{ik}(\boldsymbol{x})\sigma_{jk}(\boldsymbol{x}). \tag{13.9}$$

What is remarkable here is that the "space" variable $\boldsymbol{x}$ is in with a dimension d that can be of course big. Fokker-Planck equations naturally are high-dimensional PDEs. This feature makes their numerical solution especially hard once d is greater than four.

Let us show that the Fokker-Planck equation is actually a standard convection-diffusion equation. First denote $A(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{x})\boldsymbol{\sigma}(\boldsymbol{x})^T \in \mathscr{M}_{d\times d}$. We have (using Einstein's index summation)

$$\begin{aligned} \partial_i(\partial_j(a_{ij}(\boldsymbol{x})\, p)) &= \partial_i\left[a_{ij,j}(\boldsymbol{x})\, p + a_{ij}(\boldsymbol{x})\, p_{,j}\right] \\ &= \nabla_{\boldsymbol{x}} \cdot (A(\boldsymbol{x})\, \nabla p) + \partial_i\left(a_{ij,j}(\boldsymbol{x})\, p\right). \end{aligned}$$

Therefore, denoting by $\boldsymbol{v}(\boldsymbol{x})$ the vector field defined by

$$v_i(\boldsymbol{x}) = u_i(\boldsymbol{x}) - \frac{1}{2}\sum_{j=1}^{d} \frac{\partial a_{ij}}{\partial x_j}(\boldsymbol{x}), \tag{13.10}$$

the Fokker-Planck equation is written

$$\frac{\partial p}{\partial t} + \nabla_x \cdot (\boldsymbol{v}(\boldsymbol{x})\, p) - \frac{1}{2}\nabla_x \cdot (A(\boldsymbol{x})\nabla p) = 0, \tag{13.11}$$

this is a linear high-dimensional convection-diffusion equation with symmetric positive variable coefficient diffusion tensor $A(\boldsymbol{x})$.

In the next sections, we are going to see how to solve these problem numerically.

13.3 Computational approaches based on Stochastic Differential Equations

For high-dimensional problems, it more convenient to integrate the SDE. The stochastic extension of the Euler integration scheme is named the Euler-Maruyama approximation ([]): from a time instant t^n, a random variable $\boldsymbol{X}^n$ at time t^n, Δt a time step, one computes the stochastic discrete process $(\boldsymbol{X}^n, n \in \mathbb{N})$

$$\boldsymbol{X}^{n+1} = \boldsymbol{X}^n + \Delta t\, \boldsymbol{u}(\boldsymbol{X}^n) + \sqrt{\Delta t}\, \boldsymbol{\sigma}(\boldsymbol{X}^n)\, (\boldsymbol{B}^{n+1} - \boldsymbol{B}^n), \quad n \in \mathbb{N}, \tag{13.12}$$

with initial random variable

$$\boldsymbol{X}^0 = \boldsymbol{X}. \tag{13.13}$$

In (13.12), the random variables $\boldsymbol{B}^n$ are p-dimensional independent variables with components $B_1^n,\ B_2^n,\ \ldots, B_p^n$ which are $\mathcal{N}(0,1)$ independent normally distributed random variables.

13.3.1 Monte-Carlo methods

The expressions (13.12),(13.13) are still theoretical. In order to have practical computations, some realizations of these random variables are needed. Particle Monte-Carlo methods [] consist of the direct numerical simulation of a certain number M of sample paths (also referred to as particles) using M realizations of the initial random variable $\boldsymbol{X}$ and M discrete realization of the stochastic process $(\boldsymbol{X}^n, n \in \mathbb{N})$, also involving M realizations of the p-dimensional Gaussian variables $\boldsymbol{B}^n$, $n \in \mathbb{N}$.

Monte-Carlo methods clearly are independent of the dimension of the problem what makes particularly attractive for high-dimensional problems. Unfortunately, it is known that these methods suffer from a lack of accuracy and anyway require a large number of realizations to get statistically accurate results.

13.4 Numerical solution of the Fokker-Planck equations

For $d = 1,\ 2$ or 3, the Fokker-Planck equation (13.11) can be discretized using standard Finite element or Finite Volume methods. This kind of discretization leads to much more accurate methods than Monte-Carlo. Introducing the total derivative

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{v}(\boldsymbol{x}) \cdot \nabla_x, \tag{13.14}$$

equation (13.11) can be written

$$\frac{Dp}{Dt} + p\,(\nabla \cdot \boldsymbol{v}) - \frac{1}{2}\nabla \cdot (A(\boldsymbol{x})\nabla p) = 0. \tag{13.15}$$

A semi-discretization of this equation gives, as already discussed in previous chapters

$$\frac{p^{n+1}(\boldsymbol{x}) - p^n \circ X^n(\boldsymbol{x})}{\Delta t^n} + p^{n+1}(\boldsymbol{x})(\nabla \cdot \boldsymbol{v})(\boldsymbol{x}) - \frac{1}{2}\nabla \cdot (A(\boldsymbol{x})\nabla p^{n+1}(\boldsymbol{x})) = 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{13.16}$$

For spatial discretization, we need to restrict the state space $\mathbb{R}^d$ to a bounded domain. What is generally chosen is a truncation rectangular domain with boundaries far enough such that the probability density is close to zero on the artificial boundary, for all time $t \geq 0$.

13.4.1 Boundary conditions

Equation (13.11) is clearly a conservation equation because of the form

$$\frac{\partial p}{\partial t} + \nabla_x \cdot \boldsymbol{J} = 0 \tag{13.17}$$

with a probability flux $\boldsymbol{J}$ composed of a convective flux

$$\boldsymbol{J}^c(\boldsymbol{x}) = \boldsymbol{v}(\boldsymbol{x})\, p \tag{13.18}$$

and a diffusive flux

$$\boldsymbol{J}^d(\boldsymbol{x}) = -\frac{1}{2}A(\boldsymbol{x})\nabla p. \tag{13.19}$$

This conservation principle is necessary to guarantee the "mass" is conserved, i.e.

$$\frac{d}{dt}\int_{\mathbb{R}^d} p(\boldsymbol{x}, t)\, d\boldsymbol{x} = 0. \tag{13.20}$$

Now, if the computational domain is restricted to bounded domain Ω, we need to satisfy the same equation (13.19). By integrating equation (13.17) over Ω and applying Green's formula, we get

$$\frac{d}{dt}\int_{\mathbb{R}^d} p(\boldsymbol{x}, t)\, d\boldsymbol{x} + \int_{\partial\Omega} \boldsymbol{J} \cdot \boldsymbol{n}\, d\gamma = 0, \tag{13.21}$$

where $\boldsymbol{n}$ is the standard exterior unit vector to Ω. So a natural boundary condition is zero flux condition $\boldsymbol{J} \cdot \boldsymbol{n} = 0$ to ensure mass conservation, i.e.

$$(\boldsymbol{v} \cdot \boldsymbol{n})\, p - \frac{1}{2}\,(A(.)\nabla p) \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega \tag{13.22}$$

which is a generalized Robin boundary condition. To summarize, now we have to solve by Finite Elements the spatial problem

$$\frac{p^{n+1} - p^n \circ X^n}{\Delta t^n} + p^{n+1}(\nabla \cdot \boldsymbol{v}) - \frac{1}{2}\nabla \cdot (A\nabla p^{n+1}) = 0 \quad \text{in } \Omega, \tag{13.23}$$

$$(\boldsymbol{v} \cdot \boldsymbol{n})\, p^{n+1} - \frac{1}{2}\,\left(A\nabla p^{n+1}\right) \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega. \tag{13.24}$$

Therefore, a variational problem in the $H^1(\Omega)$ Sobolev space is formulated as follows: *find* $p^{n+1} \in H^1(\Omega)$ *such that, for all* $q \in H^1(\Omega)$,

$$\int_\Omega \frac{p^{n+1} - p^n \circ X^n}{\Delta t^n} q \, d\boldsymbol{x} + \int_\Omega (\nabla \cdot \boldsymbol{v}) \, p^{n+1} \, q \, d\boldsymbol{x} - \int_{\partial\Omega} (\boldsymbol{v} \cdot \boldsymbol{n}) \, p^{n+1} \, q \, d\gamma + \frac{1}{2} \int_\Omega A \nabla p^{n+1} \cdot \nabla q \, d\boldsymbol{x} = 0. \tag{13.25}$$

Using conformal finite elements with a finite element discrete space $V^h \subset H^1(\Omega)$, the discrete problem to solve is to find $p^{h,n+1} \in V^h$ such that

$$\int_{\Omega^h} \frac{p^{h,n+1} - p^{h,n} \circ X^n}{\Delta t^n} q^h \, d\boldsymbol{x} + \int_{\Omega^h} (\nabla \cdot \boldsymbol{v}) \, p^{h,n+1} \, q^h \, d\boldsymbol{x} - \int_{\partial\Omega^h} (\boldsymbol{v} \cdot \boldsymbol{n}) \, p^{h,n+1} \, q^h \, d\gamma + \frac{1}{2} \int_{\Omega^h} A \nabla p^{h,n+1} \cdot \nabla q^h \, d\boldsymbol{x} = 0 \quad \forall q^h \in V^h. \tag{13.26}$$

13.5 Numerical example : metabolite reactions

Consider the following five reactions for the molecular species X and Y modeling the creation of two metabolites controled by two enzymes, a reaction and their destruction:

$$\emptyset \xrightarrow{k_X} X \qquad \emptyset \xrightarrow{k_Y} Y$$
$$X + Y \xrightarrow{k_2 xy} \emptyset$$
$$X \xrightarrow{\mu x} \emptyset \qquad \xrightarrow{\mu y} \emptyset.$$

Without stochastic effects, according to the reaction rates, the deterministic system if differential equations that governs the molecular system is

$$\frac{dx}{dt} = k_X - \mu x - k_2 xy, \tag{13.27}$$

$$\frac{dy}{dt} = k_Y - \mu y - k_2 xy \tag{13.28}$$

with $x, y > 0$. It is easy to check that this system has a unique stable equilibrium state in $(\mathbb{R}^+)^2$. For a biomolecular system, it is often not reasonable to assume that the biomolecular system evolve continuous as a continuous medium because molecules like enzymes or metabolites are in too low number and molecular collision arise with some noise. Here, because there are five reactions, there are five independent noises that represent the uncertainty of molecular collision for each reaction. Statistics theory states that the standard deviation on reaction rates is proportional to the square root of the mean reaction rates. Thus, the stochastic differential system is written

$$dX = (k_X - \mu X - k_2 XY) \, dt + \sqrt{k_X} \, dB_1(t) + \sqrt{\mu X} \, dB_3(t) + \sqrt{k_2 XY} \, dB_5(t), \tag{13.29}$$

$$dY = (k_Y - \mu Y - k_2 XY) \, dt + \sqrt{k_Y} \, dB_2(t) + \sqrt{\mu Y} \, dB_4(t) + \sqrt{k_2 XY} \, dB_5(t). \tag{13.30}$$

Notice that the system (13.29),(13.30) is in the form (13.6) with

$$\boldsymbol{\sigma}(t) = N\, W^{1/2}, \tag{13.31}$$

where $W = diag\,(k_X, k_Y, \mu x, \mu y, k_2 xy)$ and N is the stoichiometric coefiicient matrix

$$N = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \tag{13.32}$$

That gives the following diffusion tensor A

$$A(\boldsymbol{x}) = \begin{pmatrix} k_X + \mu x + k_2 xy & k_2 xy \\ k_2 xy & k_Y + \mu y + k_2 xy \end{pmatrix}. \tag{13.33}$$

For numerical experiments, we will use the following coefficients: $k_X = k_Y = 0.6$, $\mu = k_2 = 0.001$. The truncated state domain $\Omega = \{0 \le x, y \le 200\}$ will be used.

13.5.1 `Scilab` source code of the Monte Carlo approach

```
// Metobilte.sce (Scilab)
// Simple stochastic metabolite reaction model
//
// Numerical solution using the Maruyama−Euler scheme.
// Monte−Carlo approach.
clear;
kx = 0.6; ky=kx;
mu = 0.001;
k2 = 0.001;
N = 5000;
Deltat = 0.05;
sqdt = sqrt(Deltat);
X = zeros(2,N);
//
// Initial cloud (N realisation of the
// initial random state)
X(1,:) = 140 + 5*rand(1,N, "normal");
X(2,:) = 160 + 5*rand(1,N, "normal");
clf(); plot(X(1,:), X(2,:), 'o');
plot([0, 200], [0,200], '.w'); xgrid();
xtitle("Initial Probability Density Function");
xlabel("X");ylabel("Y"); drawnow();
// stop;
//
// Time loop
//
Bnold = sqdt*rand(5,N, "normal");
//
for it=1:2000
   Bn = sqdt*rand(5,N, "normal");
   // Euler
   dX = (kx-mu*X(1,:)-k2*X(1,:).*X(2,:))*Deltat + ...
```

```
          sqrt(kx).*(Bn(1,:)-Bnold(1,:)) ...
           +sqrt(mu*X(1,:)) .*(Bn(3,:)-Bnold(3,:)) ...
           +sqrt(k2*X(1,:).*X(2,:)) .*(Bn(5,:)-Bnold(5,:)) ;
//
   dY = (ky-mu*X(2,:)-k2*X(1,:).*X(2,:))*Deltat + ...
             sqrt(ky).*(Bn(2,:)-Bnold(2,:)) ...
           +sqrt(mu*X(2,:)) .*(Bn(4,:)-Bnold(4,:)) ...
           +sqrt(k2*X(1,:).*X(2,:)) .*(Bn(5,:)-Bnold(5,:)) ;
   Bnold = Bn;
   X = X + [dX; dY];
   if ~modulo(it,100)
     drawlater();
     clf(); plot(X(1,:), X(2,:), 'o');xgrid();
     plot([0, 200], [0,200], '.w');
     xtitle("Probability Density Function");
     xlabel("X");ylabel("Y"); drawnow();
     drawnow();
   end;
end; //for it
```

13.5.2 Numerical results of the Monte-Carlo method

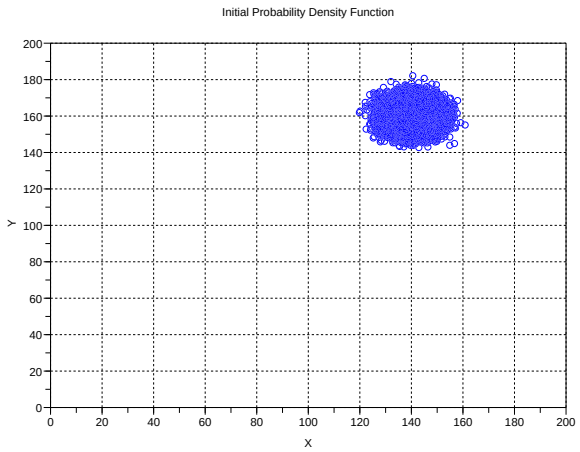


Figure 13.1: Monte-Carlo approach. $N = 20000$ realizations of the initial random variable are computed.

13.5.3 `freefem++` source of the Fokker-Planck solver

```
// Fokker.edp (Freefem++)
// Fokker-Planck equations,
// Simple stochastic metabolite reaction model
//
```

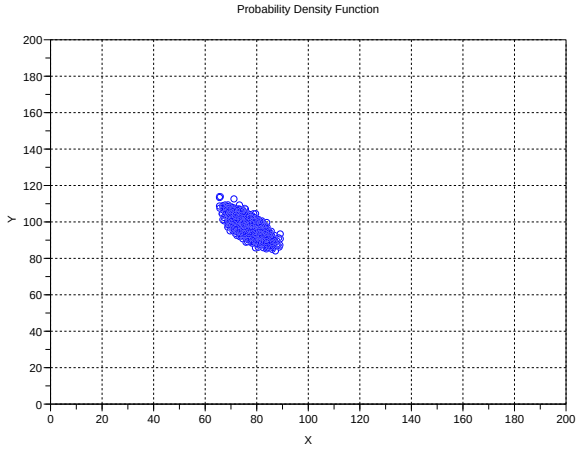


Figure 13.2: Monte-Carlo approach. Monte-Carlo particles in the state space during the transient transport phase.

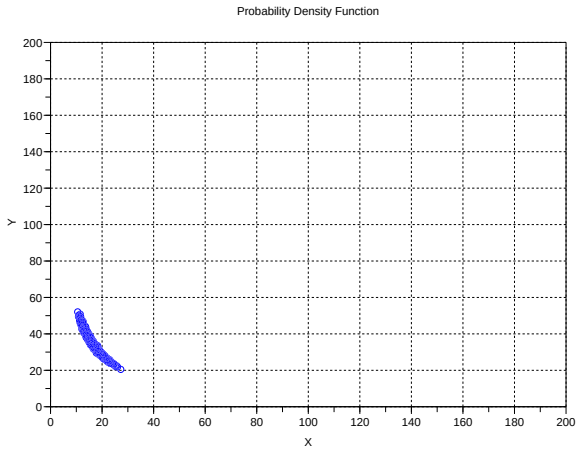


Figure 13.3: Monte-Carlo approach. Monte-Carlo particles in the state space at the statistically steady state.

```
5 real kx = 0.6;
6 real ky = kx;
7 real mu = 0.001;
8 real k2 = 0.001;
9 real dt = 0.1;
```

```
//
real Lx = 200, Ly = 200;
mesh Th = square(60, 60, [x*Lx, y*Ly]);
fespace Vh(Th, P2); fespace Wh(Th,P1);
Vh p, qh, pold;
Wh v1h, v2h;
func a11 = kx + mu*x+k2 * x*y;
func a12 = k2*x*y;
func a21 = k2*x*y;
func a22 = ky+mu*y+k2 * x*y;
func v1 = (kx-mu*x-k2*x*y) - 0.5 * (mu+k2*y +k2*x);
func v2 = (ky-mu*y-k2*x*y) - 0.5 * (k2*y +mu+k2*x);
func dxv1 = - mu - k2*y - 0.5 * k2;
func dyv2 = - mu - k2*x -0.5 * k2;
//
v1h=v1; v2h=v2; plot([v1h, v2h], ps="Velocity.eps");
// Initializing the initial PDF
real sigma = 5, sigma2=sigma^2;
p = exp(-0.5*(x-140)^2/sigma2)*exp(-0.5*(y-160)^2/sigma2);
Th = adaptmesh(Th, p); p =p;
real massp = int2d(Th)(p); p = p / massp;
pold = p;
plot(p,   nbiso=60, ps="initial.eps");
plot(p, Th, nbiso=60, ps="initialmesh.eps");
//
problem fokker(p, qh) =
  int2d(Th)(p*qh/dt)
  - int2d(Th)(convect([v1, v2], -dt, pold)*qh/dt)
  +int2d(Th)(0.5*a11*dx(p)*dx(qh)+0.5*a22*dy(p)*dy(qh))
  +int2d(Th)(0.5*a12*dy(p)*dx(qh)+0.5*a21*dx(p)*dy(qh))
  - int1d(Th, 1,2,3,4)((v1*N.x+v2*N.y)*p*qh)
  +int2d(Th)( (dxv1+dyv2)*p*qh );
//
int it;
for (it=0; it<200; it++) {
  fokker; massp = int2d(Th)(p); p = p /massp; pold = p;
  fokker; massp = int2d(Th)(p); p = p /massp; pold = p;
  fokker; massp = int2d(Th)(p); p = p /massp; pold = p;
  fokker; massp = int2d(Th)(p); p = p /massp; pold = p;
  fokker; massp = int2d(Th)(p); p = p /massp; pold = p;
  Th = adaptmesh(Th, p);
  massp = int2d(Th)(p); p = p /massp; pold = p;
  plot(p, Th, nbiso=60);
  cout << "*** Mass = " << int2d(Th)(p) << endl;
}
plot(p, nbiso=40, ps="transient"+it+".eps");
plot(p, Th, nbiso=40, ps="transientmesh"+it+".eps");
```

13.5.4 Numerical results with the Fokker-Planck model

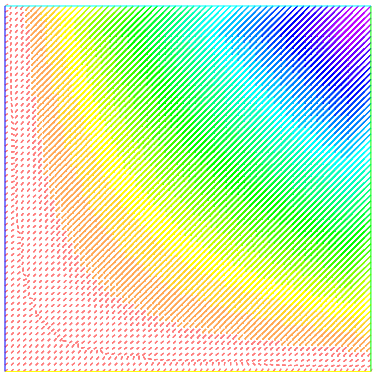


Figure 13.4: Velocity field in the state space.

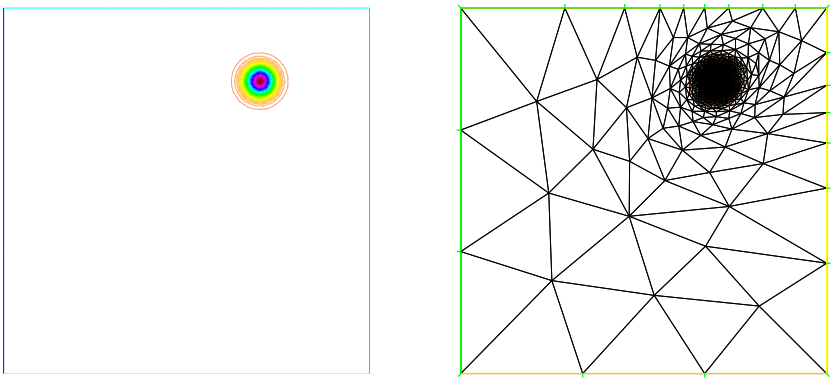


Figure 13.5: Initial probability density function. The mesh was adapted according to the initial data.

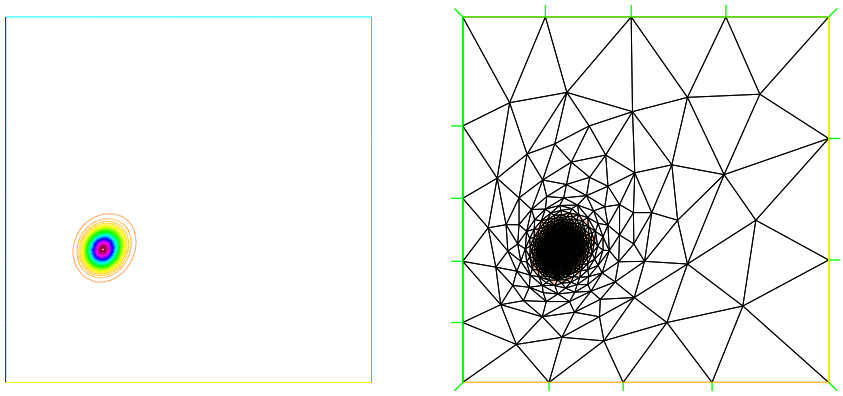


Figure 13.6: Probability density function during the transient regime. Mesh adaptation is performed during simulation.

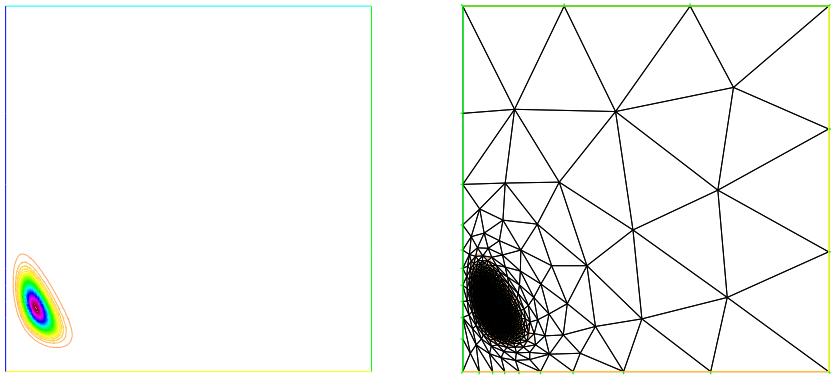


Figure 13.7: Discrete probability density function at steady state.

Chapter 14

Multiphase flows

Multiphase flows are probably the flows of most interest in Fluid Engineering. Multiphase arise in almost all fluid-based engineering process: petroleum pipe flows, plastics molding industry, pollutant treatment, gas transportation in giant gas burner, power generators, engines, propulsion, etc.

As a matter of fact, theses kinds of flows are particularly hard to numerically model because of some features: complex Physics but also free boundaries, phase change, big density ratios, etc.

Free boundary is an important feature. There are many ways to track free boundaries at the numerical point of view : level set methods, volume of fluid methods, meshless particle methods, etc. As an introduction, we will consider in this chapter a simple method which ...

As examples, we will consider two applications. The first one is a liquid sloshing in a closed box.

Of course, the simplicity of the front tracking method presented here is compensated by numerous drawbacks and numerical artefacts: loss of material mass, diffuse boundary, inaccuracy on free boundary position. To improve the front tracking, we should use advanced computational method like high-order accuracy level sets method or conservative volume of fluid methods. But this is largely beyond the scope and goal of this course of this course.

14.1 Setting of the equations

Let us consider two Newtonian viscous incompressible immiscible fluids $k = 1, 2$ with respective (constant) density ρ_k, $k = 1, 2$ and constant dynamic viscosity μ_k, $k = 1, 2$. Let us consider a spatial bounded domain $\Omega \subset \mathbb{R}^d$ filled up by the two fluids. Although each fluid is incompressible with constant density. The whole density function defined on the whole domain Ω can be seen as a function of space and time

$$\rho(\boldsymbol{x}, t) = \rho_1 \, 1_{(\boldsymbol{x} \in \Omega_1^t)}(\boldsymbol{x}) + \rho_2 \, 1_{(\boldsymbol{x} \in \Omega_2^t)}(\boldsymbol{x}) \tag{14.1}$$

where Ω_k^t, $k = 1, 2$ is the volume occupied by the fluid k at time t in Ω. In the same way, we have for the dynamic viscosity

$$\mu(\boldsymbol{x}, t) = \mu_1 \, 1_{(\boldsymbol{x} \in \Omega_1^t)}(\boldsymbol{x}) + \mu_2 \, 1_{(\boldsymbol{x} \in \Omega_2^t)}(\boldsymbol{x}). \tag{14.2}$$

By also defining a velocity $\boldsymbol{u}$ and a pressure p on the whole domain Ω, under gravity $\boldsymbol{g}$ the flow is governed by the Navier-Stokes equations

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{14.3}$$

$$\partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) - \nabla \cdot (\mu \nabla \boldsymbol{u})) + \nabla p = \rho \boldsymbol{g}. \tag{14.4}$$

The Navier-Stokes equations are in conservation form, i.e. in the form

$$\partial_t \boldsymbol{U} + \nabla \cdot \mathbf{F} = \boldsymbol{S}, \tag{14.5}$$

with $\boldsymbol{U} = (\rho, \boldsymbol{u})$, $\boldsymbol{F}$, $\boldsymbol{F} = (\rho \boldsymbol{u}, \rho \boldsymbol{u} \otimes \boldsymbol{u} - \mu \nabla \boldsymbol{u} + p\, I)$ and $\boldsymbol{S} = (0, \rho \boldsymbol{g})$.

14.1.1 Transmission conditions, jump conditions

We have to understand if the equations intrinsically include the necessary transmission condition between the two fluids. The surface tension forces at fluid interface is neglected in this study.

Suppose that $\boldsymbol{U}$ is a piecewise $\mathscr{C}^1$ function solution of (14.5). Let $p = 1 + d$ and $\varphi \in \mathscr{D}(\Omega)$. By applying the Green formula, we have for any spatio-temporal domain $D \in \Omega \times (0, T)$

$$\int_D \left\{ \boldsymbol{U} \cdot \frac{\partial \varphi}{\partial t} + \sum_{j=1}^{d} \boldsymbol{F}_j \cdot \frac{\partial \varphi}{\partial x_j} - \boldsymbol{S} \cdot \varphi \right\} d\boldsymbol{x}\, dt = 0. \tag{14.6}$$

Now let Σ be the fluid interface between the two immiscible fluids, M be a point of Σ and D be a small ball centered at M in the $(\boldsymbol{x}, t)$-plane. We denote by D^- and D^+ the two open components of D on each side of Σ (see figure 14.1. From (14.6), we also have

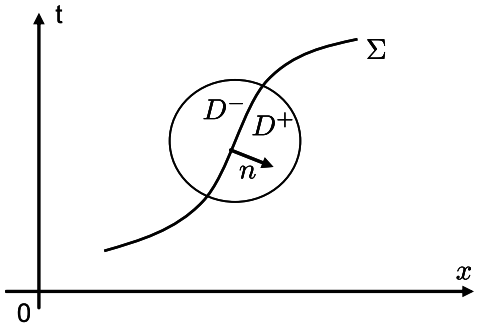


Figure 14.1: Jump transmission condition at fluid interface ($(\boldsymbol{x}, t)$-plane).

$$\int_{D^-} \{\boldsymbol{U} \cdot \frac{\partial \varphi}{\partial t} + \sum_{j=1}^{d} \boldsymbol{F}_j \cdot \frac{\partial \varphi}{\partial x_j} - \boldsymbol{S} \cdot \varphi\}\, d\boldsymbol{x}\, dt + \int_{D^+} \{\boldsymbol{U} \cdot \frac{\partial \varphi}{\partial t} + \sum_{j=1}^{d} \boldsymbol{F}_j \cdot \frac{\partial \varphi}{\partial x_j} - \boldsymbol{S} \cdot \varphi\}\, d\boldsymbol{x}\, dt = 0. \tag{14.7}$$

Suppose for instance that the normal vector $\boldsymbol{n}$ to the surface Σ points in D^+. Then, applying Green's formula in D^+ and D^- gives

$$\begin{aligned} 0 \;=\; & -\int_{D^+} \{\frac{\partial \boldsymbol{U}}{\partial t} + \sum_{j=1}^{d} \frac{\partial \boldsymbol{F}^j}{\partial x_j} - \boldsymbol{S}\} \cdot \varphi\, d\boldsymbol{x}\, dt - \int_{\Sigma \cap D} \{n_t \boldsymbol{U}^+ + \sum_{j=1}^{d} n_{x_j} \boldsymbol{F}_j^+\} \cdot \varphi\, ds \\ & - \int_{D^-} \{\frac{\partial \boldsymbol{U}}{\partial t} + \sum_{j=1}^{d} \frac{\partial \boldsymbol{F}^j}{\partial x_j} - \boldsymbol{S}\} \cdot \varphi\, d\boldsymbol{x}\, dt + \int_{\Sigma \cap D} \{n_t \boldsymbol{U}^- + \sum_{j=1}^{d} n_{x_j} \boldsymbol{F}_j^-\} \cdot \varphi\, ds. \end{aligned}$$

Since $\boldsymbol{U}$ is a classical solution of (14.5), we obtain

$$\int_{\Sigma\cap D}\{-n_t(\boldsymbol{U}^+-\boldsymbol{U}^-)-\sum_{j=1}^{d}\left(\boldsymbol{F}_j^+-\boldsymbol{F}_j^-\right)\cdot\varphi\,ds\}=0.$$

Since φ is arbitrary, we obtain the jump relation at the point M

$$-n_t(\boldsymbol{U}^+-\boldsymbol{U}^-)-\sum_{j=1}^{d}\left(\boldsymbol{F}_j^+-\boldsymbol{F}_j^-\right)=0. \tag{14.8}$$

The jump relation (14.8) is known the Rankine-Hugoniot condition, generalized here to any space dimension. If we denote by

$$|[\boldsymbol{U}]|=\boldsymbol{U}^+-\boldsymbol{U}^-$$

the jump of $\boldsymbol{U}$ and similarly by

$$|[\boldsymbol{F}_j|]=\boldsymbol{F}_j^+-\boldsymbol{F}_j^-$$

the jump of $\boldsymbol{F}_j$, $j=1,\ldots,d$, the jump condition may be written

$$n_t\,|[\boldsymbol{U}|]+\sum_{j=1}^{d}n_{x_j}\,|[\boldsymbol{F}_j|]=0. \tag{14.9}$$

If $(n_{x_1},\ldots,n_{x_d})\neq 0$, let us set

$$\boldsymbol{n}=(-\sigma,\boldsymbol{\nu})$$

where $\sigma\in\mathbb{R}$ and $\boldsymbol{\nu}=(\nu_1,\ldots,\nu_d)$ is a unit vector of $\mathbb{R}^d$. Thené(14.9) can be equivalently written in the form

$$\sigma|[\boldsymbol{U}|]=\sum_{j=1}^{d}\nu_j\,|[\boldsymbol{F}_j]|. \tag{14.10}$$

If Σ is oriented and $\frac{\boldsymbol{n}}{|\boldsymbol{n}|}$ is the outward unit normal vector to σ, then $\boldsymbol{\nu}$ and σ may be interpreted respectively as the direction and the speed of propagation of the discontinuity Σ. Using the definition of $\boldsymbol{U}$ and $\boldsymbol{F}$, we get component by component

$$\sigma|[\rho]|=|[\rho\,\boldsymbol{u}\cdot\nu]| \tag{14.11}$$

and

$$\sigma|[\rho\boldsymbol{u}]|=|[\rho\boldsymbol{u}\,\boldsymbol{u}\cdot\boldsymbol{\nu}+p\boldsymbol{\nu}-\mu\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{\nu}}]|. \tag{14.12}$$

Let us define $\Phi=\rho(\boldsymbol{u}\cdot\boldsymbol{\nu}-\sigma)$ the mass flux through the interface Σ. Because the two fluids are supposed to be immiscible, there is no mass transfer between the two fluids. Thus $\Phi=0$ and we have

$$\boldsymbol{u}^+\cdot\boldsymbol{\nu}=\boldsymbol{u}^-\cdot\boldsymbol{\nu}=\sigma.$$

and in particular

$$|[\boldsymbol{u}\cdot\nu]|=0. \tag{14.13}$$

The transmission condition (14.13) is referred to as the kinematic jump condition. The second compatibility condition (14.12) can be rewritten

$$|[\Phi\,\boldsymbol{u}+p\boldsymbol{\nu}-\mu\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{\nu}}]|=0. \tag{14.14}$$

But because $\Phi = 0$ at the fluid interface, we have the second jump condition

$$|[-\mu \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\nu}} + p\boldsymbol{\nu}]| = 0. \tag{14.15}$$

Equation (14.15) is referred to as the dynamic jump condition.

14.1.2 Final model

Because each of the two fluids are incompressible, one may look for a simpler system of equations. Because $\nabla \cdot \boldsymbol{u} = 0$, the continuity equation is actually a transport equation which can be written

$$D_t \rho = 0 \tag{14.16}$$

using the Lagrangian derivative. This is also true for the dynamic viscosity

$$D_t \mu = 0. \tag{14.17}$$

Still because $\nabla \cdot \boldsymbol{u} = 0$, the momentum equation can write

$$D_t(\rho \boldsymbol{u}) - \nabla \cdot (\mu \nabla \boldsymbol{u}) + \nabla p = \rho \boldsymbol{g}. \tag{14.18}$$

Finally, because $D_t \rho = 0$, we have also

$$\rho \, D_t \boldsymbol{u} - \nabla \cdot (\mu \nabla \boldsymbol{u}) + \nabla p = \rho \boldsymbol{g}. \tag{14.19}$$

The system is closed with the zero-divergence velocity condition:

$$\nabla \cdot \boldsymbol{u} = 0. \tag{14.20}$$

One gets the standard Navier-Stokes equations (14.19),(14.20) coupled with the two transport equations (14.17),(14.18). It is a simple matter to show that equations (14.19) and (14.20) respect both kinematic and dynamic jump conditions. For example, multiplying once again equation (14.20) by a function $\varphi \in \mathscr{D}(\Omega)$, integrating over the open D and applying Green's formula, one gets

$$\int_{D^+} \boldsymbol{u} \cdot \nabla \varphi \, d\boldsymbol{x} \, dt + \int_{D^-} \boldsymbol{u} \cdot \nabla \varphi \, d\boldsymbol{x} \, dt = 0.$$

Suppose for instance that the normal vector $\boldsymbol{n}$ the the surface Σ points in D^+. Then applying once again Green's formula in D^+ and D^- gives

$$0 = -\int_{D^+} \nabla \cdot \boldsymbol{u} \, \varphi \, d\boldsymbol{x} \, dt - \int_{\Sigma \cap D} \{\boldsymbol{u}^+ \cdot \boldsymbol{\nu}\} \varphi \, ds - \int_{D^-} \nabla \cdot \boldsymbol{u} \, \varphi \, d\boldsymbol{x} \, dt + \int_{\Sigma \cap D} \{-\boldsymbol{u}^- \cdot \boldsymbol{\nu}\} \varphi \, ds.$$

Since $\nabla \cdot \boldsymbol{u} = 0$ in D^- and $\nabla \cdot \boldsymbol{u} = 0$ in D^+, we obtain

$$\int_{\Sigma \cap D} |[\boldsymbol{u} \cdot \boldsymbol{\nu}]| \, \varphi \, ds = 0$$

for any arbitray $\varphi \in \mathscr{D}(\Omega)$ so that the kinematic condition is satisfied..

14.2 Semi-discretization in time

As already seen, the total derivatives are discretized according to be characteristic method. For numerical stability purposes, the other terms are made implicit. From a time step Δ, the following discrete-in-time equations are written

$$\frac{\rho^{n+1} - \rho^n \circ X^n}{\Delta t} = 0, \tag{14.21}$$

$$\frac{\mu^{n+1} - \mu^n \circ X^n}{\Delta t} = 0, \tag{14.22}$$

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0, \tag{14.23}$$

$$\rho^{n+1} \frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n \circ X^n}{\Delta t} - \nabla \cdot (\mu^{n+1} \nabla \boldsymbol{u}^{n+1}) + \nabla p^{n+1} = \rho^{n+1} \boldsymbol{g}, \tag{14.24}$$

One can observe that, due to the features of the characteristic method, to the first equations can be solved separately before the Navier-Stokes equations numerical solution. Once ρ^{n+1} and μ^{n+1} are computed, one has to solve a standard Navier-Stokes problem, but with spatially varying density and dynamic viscosity.

For a full discretization, one can use a standard Finite Element method suited for the Navier-Stokes equations.

14.3 Front tracking by a level function

Actually, one of the drawbacks of the actual numerical scheme (14.21)-(14.24) is that the transport equations $D_t \rho = 0$ and $D_t \mu = 0$ are solved by an approximation scheme. Thus, inherent numerical diffusion tends to create a discrete profile for density and viscosity near the fluid interface. Although it is expected that the density of the continuous solution is either ρ_1 or ρ_2 according to the fluid being present, the discrete density solved by the characteristic method can have unphysical values of density, especially in the interval $[\min(\rho_1, \rho_2), \max(\rho_1, \rho_2)]$. This is also true for the viscosity. A way to proceed is to consider a levelset variable ψ such that $\psi < 0$ in Ω_1^t, $\psi > 0$ in Ω_2^t and the level $\psi = 0$ exactly tracks the fluid interface σ. Because of the kinematic jump condition, a natural partial differential equation for ψ is

$$D_t \psi = 0. \tag{14.25}$$

Suppose the function Ψ known. Then both density and viscosity are computed as

$$\rho(\boldsymbol{x}, t) = \rho_1 \, 1_{(\psi(\boldsymbol{x},t)<0)}(\boldsymbol{x}) + \rho_2 \, 1_{(\psi(\boldsymbol{x},t)>0)}(\boldsymbol{x}). \tag{14.26}$$

$$\mu(\boldsymbol{x}, t) = \mu_1 \, 1_{(\psi(\boldsymbol{x},t)<0)}(\boldsymbol{x}) + \mu_2 \, 1_{(\psi(\boldsymbol{x},t)>0)}(\boldsymbol{x}). \tag{14.27}$$

By construction this computational approach leads to sharp density and viscosity profiles. This accuracy is sometimes paid by stability problems, especially for large density ratios between the two fluids.

14.4 Application. Liquid sloshing in a box.

14.4.1 `freefem++` source code of the sloshing problem

```
// Twofluid.edp (Freefem++)
// Two-fluid flow ( incompressible fluids )
// Equations :
// div u = 0 in Omega
// D_t rho = 0 in Omega
// D_t(rho u) + div(mu grad u) + grad p = rho g.
//
real Lx = 2;
real Ly = 1;
real gravity = 9.81;
real rhom = 1.0, rhop = 10.0;
real num = 0.05, nup = 0.01;
real dt = 0.2;
real myratio=1.1;
mesh Th = square(120, 60, [x*Lx, y*Ly]);
mesh Thcoarse = square(40, 30, [x*Lx, y*Ly]);
fespace Uh(Th, P1b);
fespace Uhcoarse(Thcoarse, P1);
fespace Vh(Th, P1);
fespace Wh(Th, P2);
Uh u1, u2, u1old, u2old, u1h, u2h, pvisu;
Uhcoarse u1coarse, u2coarse;
Vh p, ph;
Wh phi, phih, phiold;
Wh rho, rhoold, mu;
//
// Initialisation
phi = 0.75*Ly/Lx*x -y; phiold=phi;
u1 = 0.0; u1old = u1;
u2 = 0.0; u2old = u2;
//
problem ConvectLevel(phi, phih) =
    int2d(Th)(phi*phih/dt)
  - int2d(Th)(convect([u1old,u2old], -dt, phiold)*phih/dt)
  +int2d(Th)(0.001*dx(phi)*dx(phih)+0.001*dy(phi)*dy(phih));
//
problem NavierStokes([u1, u2, p], [u1h, u2h, ph]) =
    int2d(Th)(rho*u1*u1h / dt)
  - int2d(Th)(rho*convect([u1old,u2old], -dt, u1old)*u1h/dt)
  +int2d(Th)(mu*dx(u1)*dx(u1h)+mu*dy(u1)*dy(u1h))
  -int2d(Th)(p*dx(u1h))
  + int2d(Th)(rho*u2*u2h/dt)
  - int2d(Th)(rho*convect([u1old,u2old], -dt, u2old)*u2h/dt)
  +int2d(Th)(mu*dx(u2)*dx(u2h)+mu*dy(u2)*dy(u2h))
  -int2d(Th)(p*dy(u2h))
  +int2d(Th)(rho*gravity*u2h)
  +int2d(Th)((dx(u1)+dy(u2))*ph+0.000001*p*ph)
  +on(1,2,3,4, u1=0, u2=0);
```

```
//
for (int it=0; it<30; it++) {
  // First, convect density
  for (int innerloop=0; innerloop<1;innerloop++){
  rhoold = rho;
  phiold = phi;
  ConvectLevel;
  rho = rhom*(phi<=0)+rhop*(phi>0);
  mu = rhom*num*(phi<=0)+rhop*nup*(phi>0);
  NavierStokes;
  u1old = u1; u2old = u2;
  }
  u1coarse = u1;
  u2coarse = u2;
  //Th = adaptmesh(Th, rho, u1, u2); rho = rho; mu = mu; u1=u1; u2=u2;
  // plot(rho, nbiso=60, fill =1, ps="rho_it="+it+".eps");
  pvisu = p;
  plot(rho, nbiso=40, fill=1, value=1,
    ps="TwoFluidLevel_it"+it+".eps");
  cout << "pmin, pmax = " << p[].min << " " << p[].max << endl;
}
cout << "\n\nTwofluid.edp: Done ! \n\n\n";
```

14.4.2 Numerical results

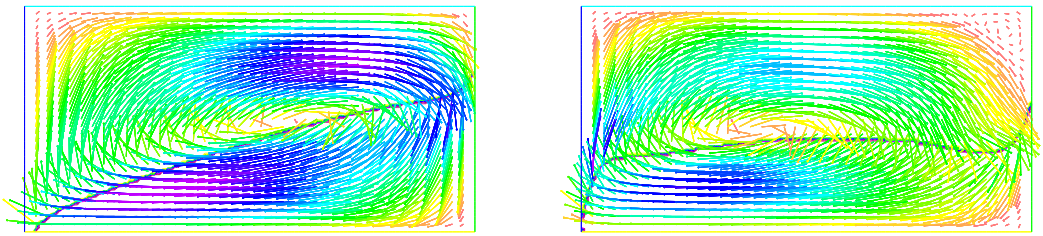


Figure 14.2: Simulation of liquid sloshing in a closed tank. Discrete free boundary and velocity field.

14.5 Application. Injection moulding problem

14.5.1 `freefem++` source code of the injection moulding problem

```
// Moulding.edp (Freefem++)
//
real[int] PA(2); PA = [0, 0];
real[int] PB(2); PB = [9, 0];
real[int] PC(2); PC = [9, 9];
real[int] PD(2); PD = [8, 9];
real[int] PE(2); PE = [0, 9];
```

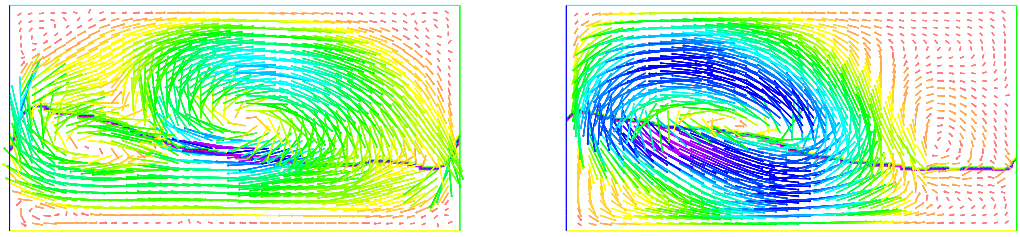


Figure 14.3: Simulation of liquid sloshing in a closed tank. Discrete free boundary and velocity field.

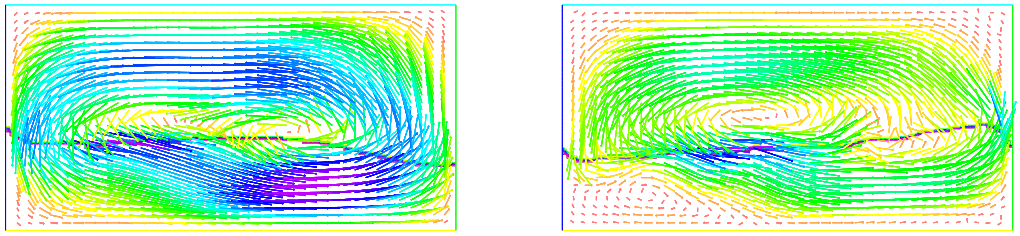


Figure 14.4: Simulation of liquid sloshing in a closed tank. Discrete free boundary and velocity field.

```
real[int] PF(2); PF = [0, 6];
real[int] PG(2); PG = [0,4];
real[int] PH(2); PH = [1, 1];
real[int] PK(2); PK = [1,8];
real[int] PL(2); PL = [8,8];
real[int] PM(2); PM = [8,5];
real[int] PO(2); PO = [5,5];
real[int] PP(2); PP = [5,4];
real[int] PQ(2); PQ = [8,4];
real[int] PR(2); PR = [8,1];
border c1(t=0,1){x=(1-t)*PA[0]+t*PB[0]; y=(1-t)*PA[1]+t*PB[1];}
border c2(t=0,1){x=(1-t)*PB[0]+t*PC[0]; y=(1-t)*PB[1]+t*PC[1];}
border c3(t=0,1){x=(1-t)*PC[0]+t*PD[0]; y=(1-t)*PC[1]+t*PD[1];}
border c4(t=0,1){x=(1-t)*PD[0]+t*PE[0]; y=(1-t)*PD[1]+t*PE[1];}
border c5(t=0,1){x=(1-t)*PE[0]+t*PF[0]; y=(1-t)*PE[1]+t*PF[1];}
border c6(t=0,1){x=(1-t)*PF[0]+t*PG[0]; y=(1-t)*PF[1]+t*PG[1];}
```

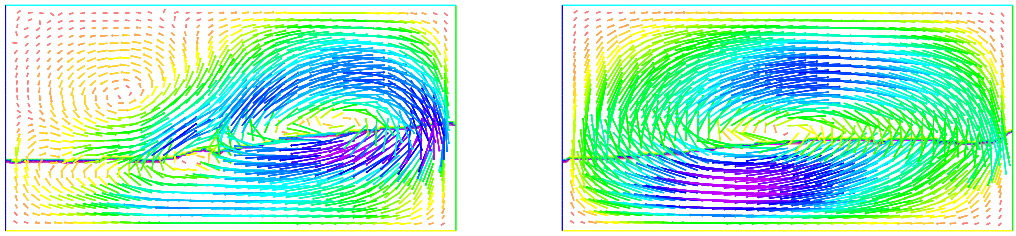


Figure 14.5: Simulation of liquid sloshing in a closed tank. Discrete free boundary and velocity field.

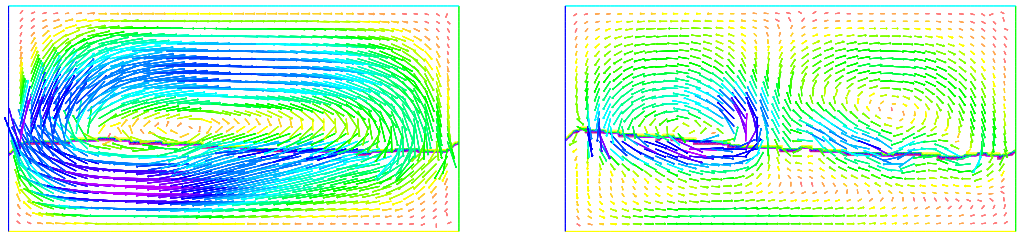


Figure 14.6: Simulation of liquid sloshing in a closed tank. Discrete free boundary and velocity field.

```
24 border c7(t=0,1){x=(1-t)*PG[0]+t*PA[0]; y=(1-t)*PG[1]+t*PA[1];}
25 //
26 border c8(t=0,1){x=(1-t)*PH[0]+t*PK[0]; y=(1-t)*PH[1]+t*PK[1];}
27 border c9(t=0,1){x=(1-t)*PK[0]+t*PL[0]; y=(1-t)*PK[1]+t*PL[1];}
28 border c10(t=0,1){x=(1-t)*PL[0]+t*PM[0]; y=(1-t)*PL[1]+t*PM[1];}
29 border c11(t=0,1){x=(1-t)*PM[0]+t*PO[0]; y=(1-t)*PM[1]+t*PO[1];}
30 border c12(t=0,1){x=(1-t)*PO[0]+t*PP[0]; y=(1-t)*PO[1]+t*PP[1];}
31 border c13(t=0,1){x=(1-t)*PP[0]+t*PQ[0]; y=(1-t)*PP[1]+t*PQ[1];}
32 border c14(t=0,1){x=(1-t)*PQ[0]+t*PR[0]; y=(1-t)*PQ[1]+t*PR[1];}
```

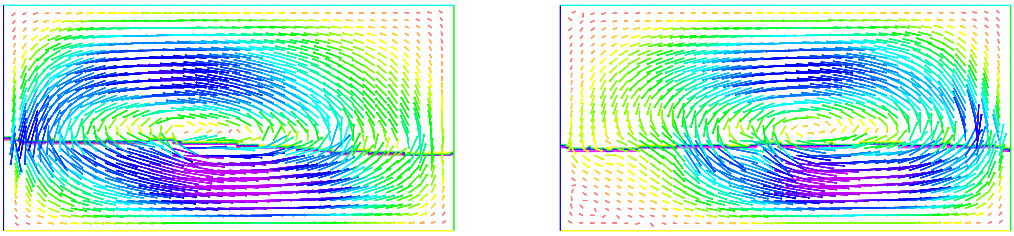


Figure 14.7: Simulation of liquid sloshing in a closed tank. Discrete free boundary and velocity field.

```
33 border c15(t=0,1){x=(1-t)*PR[0]+t*PH[0]; y=(1-t)*PR[1]+t*PH[1];}
34 //
35 int m=2;
36 mesh Th=buildmesh( c1(80*m)+c2(50*m)+c3(10*m)+c4(40*m)
37       +c5(30*m)+c6(10*m)+c7(30*m)
38       +c8(80*m)+c9(40*m)+c10(25*m)+c11(20*m)
39       +c12(5*m)+c13(20*m)+c14(25*m)+c15(30*m) );
40 plot(Th);
41 //
42 real gravity = 9.81;
43 real rhom = 1.0, rhop = 20.0;
44 real num = 0.1, nup = 10;
45 real dt = 1;
46 fespace Uh(Th, P1b);
47 fespace Vh(Th, P1);
48 fespace Wh(Th, P1b);
49 Uh u1, u2, u1old, u2old, u1h, u2h;
50 Vh p, ph;
51 Wh u1view, u2view;
52 Wh rho, rhoold, rho2, mu, muold;
53 //
54 //  Initialisation
55 rho =rhom;
56 mu = rhom*num;
57 rho = rho + (rhop-rhom)*((x<0.2)*(y>PG[1])*(y<PF[1]));
58 mu = mu + (rhop*nup-rhom*num)*((x<0.2)*(y>PG[1])*(y<PF[1]));
59 u1 = 0.0; u1old = u1;
60 u2 = 0.0; u2old = u2;
61 //
62 plot(rho, nbiso = 60, fill=1);
```

```
int it = 0;
problem NavierStokes([u1, u2, p], [u1h, u2h, ph]) =
    int2d(Th)(rho*u1*u1h/dt)
  - int2d(Th)(rhoold*convect([u1old,u2old], -dt, u1old)*u1h/dt)
  +int2d(Th)(mu*dx(u1)*dx(u1h)+mu*dy(u1)*dy(u1h))
  +int2d(Th)(dx(p)*u1h)
  + int2d(Th)(rho*u2*u2h/dt)
  - int2d(Th)(rhoold*convect([u1old,u2old], -dt, u2old)*u2h/dt)
  +int2d(Th)(mu*dx(u2)*dx(u2h)+mu*dy(u2)*dy(u2h))
  +int2d(Th)(dy(p)*u2h)
  +int2d(Th)(rho*gravity*u2h)
  +int2d(Th)((dx(u1)+dy(u2))*ph+0.000001*p*ph)
  +on(c6, u1=0.3, u2=0)
  +on(c1,c2,c4,c5,c7,c8,c9,c10,c11,c12,c13,c14,c15, u1=0, u2=0);
//
real[int] viso(1); viso = [5, 10, 15];
for (it=0; it<40; it++) {
  // First, convect density
  for (int sit=0; sit<5; sit++) {
  NavierStokes;
  u1old = u1; u2old = u2;
  rhoold = rho; muold = mu;
  rho = convect([u1old,u2old], -dt, rho);
  mu = convect([u1old,u2old], -dt, mu);
  }
  //Th = adaptmesh(Th, rho, u1, u2); rho = rho; mu = mu; u1=u1; u2=u2;
  rho2 = rhop - rho;
  plot(rho, fill=1, nbiso=3, viso=viso, value=1, ps="Moulding_rho_it"+it+"
      .eps");
  // plot(rho2, nbiso=60, fill=1, grey=1, bw=1, ps="Moulding_rho_it"+it+".eps");
}
//
cout << "\n\n\nDone.\n\n\n";
```

14.5.2 Numerical results

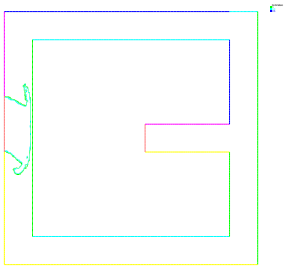


Figure 14.8: Simulation of injection molding. Material front just after the beginning of injection.

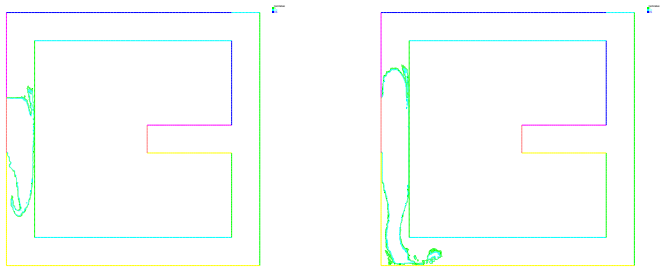


Figure 14.9: Simulation of injection molding. Material front profile during injection.

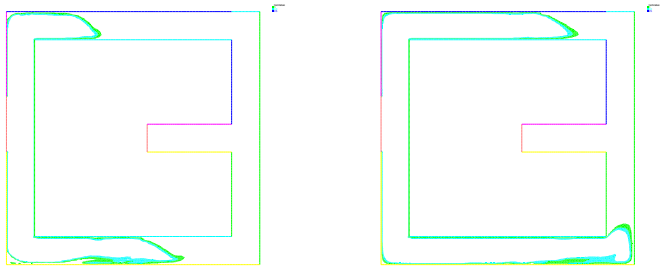


Figure 14.10: Simulation of injection molding. Material front profile during injection.

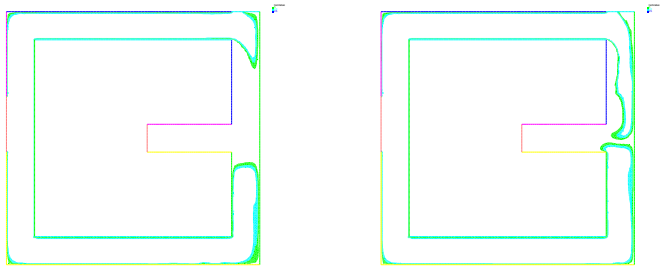


Figure 14.11: Simulation of injection molding. Material front profile during injection.

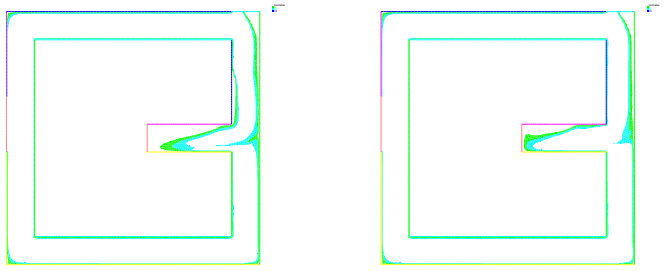


Figure 14.12: Simulation of injection molding. Material front profile during injection.

Index

accuracy, 18, 94
adaptmesh (freefem++), 120
amplification factor, 20
artificial viscosity, 23
autosimilar solution, 89

bifurcation, 81
bilinear form, 30
biomolecular system, 137
boundary condition, 79
boundary conditions, 9
buoyancy, 123

Cauchy-Schwarz inequality, 30
CFL condition, 15, 93
chemotaxis, 77
coercive form, 30
compressible flow, 113
conservation form, 10
conservative equation, 9
conservative form, 87
conservative scheme, 93
consistency, 18
continuity equation, 113
convect (freefem++), 42
convection problem, 44
convection-diffusion, 123
convection-reaction-diffusion, 78
convective flux, 136
cost function, 36
Courant number, 16

diffusion process, 134
diffusive flux, 78, 136
dimensionless equation, 78, 124
discontinuity, 88
dispersion relation, 80
dynamic jump condition, 148

eigenvalue problem, 80
elliptic problem, 29
energy equation, 113
equilibrium, 79
equivalent equation, 23
Euler scheme, 14
Euler-Maruyama scheme, 135

finite difference, 13, 14
finite element, 31
finite volume, 13
Fokker-Planck equation, 134
Fourier transform, 17
fractional step method, 82
freefem++, 29
freeway, 102
front tracking, 149
functional, 36
fundamental diagram, 88

gas dynamics, 11
Green's formula, 10, 29, 136

heat problem, 32
heat transfer, 123
Heun scheme, 61
Hilbert space, 30
hybrid scheme, 22

incompressible, 9, 11
inf-sup condition, 50

jump condition, 147
jump conditions, 91

kinematic condition, 147
kinematic equation, 7
kinematic viscosity, 123

Lagrangian, 9

Lagrangian derivative, 7
lagrangian derivative, 41
Laplace problem, 29
Lax-Friedrichs scheme, 20, 94
Lax-Milgram Theorem, 30, 33
Lax-Wendroff scheme, 18, 94
LBB condition, 50
linear stability, 79
Lokta-Volterra equations, 65

Mach number, 117
metabolite reactions, 137
method of characteristics, 41
momentum equation, 113
Monte-Carlo method, 135

Navier-Stokes equations, 123
nonconservative form, 115
numerical flux, 19, 21, 93

optimization, 36
order of accuracy, 18
ordinary differential equation, 133

particle derivative, 7, 41
planar wave, 80
Poiseuille flow, 125
Prandtl number, 124
probability density function, 134

Rankine-Hugoniot condition, 147
Rankine-Hugoniot relations, 91
rarefaction wave, 90
Rayleigh-Taylor instabilities, 123
Reynolds number, 124
Reynolds theorem, 10
RK2, 61
Robin boundary condition, 136
Roe scheme, 93

scilab, 23
semi-discrete form, 32
semi-discretization, 82
semi-implicit, 82
shape optimization, 36
shock wave, 90
Sobolev space, 30, 33, 137
stability, 15, 79
stable equilibrium, 79
steady state, 79
stochastic, 133, 137
stochastic differential equation, 135

thermal conductivity, 37, 125
thermal engineering, 34
total derivative, 41, 115
trace function, 33
transport equation, 7, 11

uncertainty, 137
unstable equilibrium, 79
upwind scheme, 19

variational formulation, 33, 116
vehicle traffic flow, 87
von Neumann stability, 17

weak solution, 91
working fluid, 123

zero flux boundary condition, 79, 136

Bibliography

[1] P.A. Raviart, J.M. Thomas, Introduction à l'Analyse numérique des équations aux dérivées partielles, Dunod (2004).

[2] I. Danaila, F. Hecht, O. Pironneau, Simulation numérique en C++ - Cours et exercices corrigés, Dunod (2003).

[3] E. F. Toro, Godunov Methods: theory and applications, Kluwer Academic (2001).

[4] E. Godlewski, P.A. Raviart, Numerical approximation of hyperbolic systems of conservation laws, Springer-Verlag (1996).

[5] H. Yamaguchi, Engineering Fluid Mechanics, Kluwer Academic Publishers (2008)

[6] R. Temam and A. Miranville, Modélisation Mathématique et Mécanique des milieux continus, Springer (2002).

[7] R. Haberman, Mathematical models: mechanical vibrations, population dynamics and traffic flow, Society for Industrial and Applied Mathematics, Classics in Applied Mathematics (1998).

[8] B. Lucquin, O. Pironneau, Introduction to Scientific Computing, John Wiley and Sons (1998).

[9] N. G. van Kampen, Stochastic processes in Physics and Chemistry, North-Holland (1992).

[10] J. Istas, Mathematical modeling for the life sciences, Springer-Verlag Berlin (2005).

[11] B. S. Kerner, The Physics of Traffic: empirical freeway pattern features, engineering applications and Theory, Springer-Verlag Berlin and Heidelberg (2004).

[12] L. Edelstein-Keshet, Mathamtical Models in Biology, Society for Industrial and Applied Mathematics, Classics in Applied Mathematics (2008).

[13] J. N. Reddy, The Finite Element Method in heat transfer and Fluid Dynamics, CRC Press Inc. (2000).

[14] B. Oksendal, Stochastic Differential Equations: an introduction with applications, Springer-Verlag (2003).

[15] P. Kotelenez, Stochastic ordinary and stochastic partial differential equations: transition from microscopic to macroscopic equations, Springer-Verlag New-York Inc. (2008).

[16] Stochastic Integration and differential equations, Springer-Verlag Berlin (2003).

[17] Freefem++ Open Source Software, http://www.freefem.org/.

[18] Scilab Open Source Software, http://www.scilab.org/.

[19] J.P. Chancelier, D. Delebecque, C. Gomez, M. Goursat, Introduction à Scilab, Springer Editions (2007).