ENSTA B2-1 Lecture notes

Introduction to the finite element method applied to incompressible fluid mechanics

Stéphane GOUNAND[1]

English translation by
Stéphane GOUNAND[1] and Sergey KUDRIAKOV[2]

July 16, 2021

[1]Commissariat à l'Énergie Atomique et aux Énergies Alternatives
Université Paris-Saclay, DES/ISAS/DM2S/SEMT/LTA, F-91191 Gif-sur-Yvette, France
mailto:stephane.gounand@cea.fr
[2]Commissariat à l'Énergie Atomique et aux Énergies Alternatives
Université Paris-Saclay, DES/ISAS/DM2S/STMF/LATF, F-91191 Gif-sur-Yvette, France
mailto:sergey.kudriakov@cea.fr

Contents

Introduction

Goal

These lecture notes are intended for undergraduate students. It is expected that the students have already been taught Fluid Mechanics and the Finite Element method. However, a brief review of these two subjects is given in chapters 1 and 3.

The main goal of these lecture notes is to present the peculiarities of the finite element method when used for the computation of Newtonian incompressible flows described by the Navier-Stokes equations.

The approach taken here is to give basic knowledge about the generic difficulties one will have to deal with while using a Computational Fluid Dynamics (CFD) code. The presentation will be quite brief on theoretical aspects. Instead, we shall give a lot of practical examples. For a more complete and rigorous approach, the reader is referred for example to Ern and Guermond's book [EG04] or to the book by Elman et al. [ESW14].

Each difficulty will be examined by considering a model problem, which usually comes from a simplification of the incompressible Navier-Stokes equations. This will help us in bringing up the cause and possible cure of the aforementioned difficulties. At the beginning of each chapter, we frame the terms in the Navier-Stokes equations on which we shall focus our attention.

The computer program we use in the examples is the CastɜM finite element code. However any other program using the same method (FreeFEM, Comsol…) could be used because the difficulties we look at are generic. In order for the student to reproduce and build on the examples, the significant part of the CastɜM data file is given. The complete data files can be found on the CastɜM Web site: `http://www-cast3m.cea.fr/`.

Plan

In chapter 1 we derive the incompressible Navier-Stokes from first principles: conservation of mass, momentum and energy. We draw particular attention to the simplifications and constitutive laws that we use.

In chapter 2 we consider two equilibrium problems for which we have a minimization (variational) principle: the Dirichlet (thermal) problem and a Linear Elasticity (mechanical) problem. These two problems can be discretized by the finite-element method in a quite natural way.

Incompressible fluid mechanics deviates from such a variational setting in two major ways:

- some terms in the Navier-Stokes equations, such as the convective terms, cannot be simply derived from a variational statement;

- the underlying variational statement can be more complex: constrained minimization instead of classical minimization, as is the case for the Stokes problem.

Due to the above difficulties application of the finite-element method to incompressible fluid mechanics is less natural than in the simple thermal and mechanical problems.

In chapter 3 we generalize the finite element method to a class of equations more general than those which are derived from a variational principle. This generalization is called the weighted residual method. The important concepts of convergence, consistency and stability are brought forward.

In chapter 4 we focus on the discretization of convective terms. We show that sometimes one has to use the so-called upwinding methods in order to compute non-oscillating discrete solutions. We will explain the various possible causes that can give rise to these oscillations: lack of stability of the discrete problem (unbounded oscillations), difficulty in approximating stiff solutions (shocks) with continuous functions due to the Gibbs phenomenon (bounded oscillations).

Chapter 5 deals with time discretization of unsteady problems. We focus on implicit finite-difference time discretization. The important choice of the initial condition will also be discussed.

Chapter 6 tackles the solution of non-linear partial differential equations (PDE), such as Navier-Stokes equations. Two solution methods will be described: Newton and fixed-point iterations.

In chapter 7 we show how the non-linearities in the equations can have interesting effects on the regularity of the solutions, such as the occurence of discontinuities (shocks). We present a mid-way summary at the end of this chapter.

Chapter 8 tackles the second aforementioned difficulty: the constrained minimization statement underlying Stokes problem. Due to this difficulty, the choice of finite-element discretization spaces for the velocity variable and for the constraining pressure variable is not arbitrary. It must be treated with care in order to have a well-posed discrete problem.

Chapter 9 is somewhat technical but important: we describe in details the boundary conditions and conservation properties attached to the finite-element method. We feel that these points are of great importance and are frequently overlooked. Due to its underlying variational nature, the finite-element method is able to tackle the notions of essential and natural boundary conditions in a very elegant way compared to other methods. The conservation properties of the finite-element method arise from the boundary conditions in an intimately linked way.

Finally chapter 10 brings together the methods of the previous chapter. As an application we describe a simple relaxed fixed point algorithm that is used in order to find an approximate solution to non-linear and unsteady incompressible fluid dynamics equations discretized by the finite-element method. Its practical implementation in the CastƎM code (`EXEC` procedure) is discussed. The students will use this procedure in their final projects. Learning how to prescribe the main parameters of the solution algorithm (mesh, time step, relaxation factor…) is dealt with in this chapter. This is of paramount importance in order to obtain reliable numerical solutions.

Acknowledgements

I thank Alberto Beccantini, Serge Pascal and Matteo Bucci for careful reading of the manuscript. Their useful suggestions helped in improving it. Frédéric Dabbene and Henri Paillère, who also authored some lectures notes given at ENSTA engineering school [DP00], were a great source of inspiration. I thank them for that. Some freely available software programs were used in the preparation of this document: Linux, LaTeX, Xfig, Emacs, Cast3M. I wish to thank their respective authors.

Chapter 1

Fluid Mechanics review

We recall in this chapter different forms of the conservation laws that will be considered in these notes. This chapter mimics the one from the book by Bird et al. [BAH87]. However, we use the more common notations and sign conventions of continuum mechanics which can be found, for example, in the book of Gurtin [Gur81].

1.1 Conservation laws

The motion of a fluid parcel is described by the conservation laws for mass, momentum and energy.

1.1.1 Mass conservation

Let us consider the mass conservation equation in *integral form.* We choose a *fixed* arbitrary volume Ω with boundary $\delta\Omega$, and let $\boldsymbol{n}$ be the outgoing normal to the boundary. We have:

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\,\mathrm{d}\Omega}_{\text{Mass increase rate in }\Omega} \quad = \quad \underbrace{-\int_{\delta\Omega}\rho\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}\delta\Omega}_{\text{Mass flux flowing in through }\delta\Omega} \tag{1.1}$$

where ρ is the fluid density and $\boldsymbol{u}$ is its velocity. Exchanging the integral sign and the time derivative[1] in the left term and applying the divergence theorem on the right term, one gets:

$$\int_{\Omega}\frac{\partial\rho}{\partial t}+(\operatorname{div}\rho\boldsymbol{u})\;\mathrm{d}\Omega=0 \tag{1.2}$$

Finally using the localization theorem [Gur81], and since Ω is an arbitrary volume, we are led to:

$$\boxed{\frac{\partial\rho}{\partial t}=-\operatorname{div}\rho\boldsymbol{u}} \tag{1.3}$$

This is the *mass conservation* equation or *continuity* equation.

[1] This is possible because the domain Ω is fixed.

1.1.2 Momentum conservation

In integral form:

$$\begin{aligned}\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\boldsymbol{u}\,\mathrm{d}\Omega}_{\text{Momentum increase rate in }\Omega} &= \underbrace{-\int_{\delta\Omega}\rho(\boldsymbol{u}\otimes\boldsymbol{u})\boldsymbol{n}\,\mathrm{d}\delta\Omega}_{\text{Incoming momentum flux due to fluid transport through }\delta\Omega} \\ &\quad \underbrace{+\int_{\delta\Omega}\boldsymbol{\sigma}\boldsymbol{n}\,\mathrm{d}\delta\Omega}_{\text{Incoming momentum flux due to molecular interactions at boundary }\delta\Omega} \quad \underbrace{+\int_{\Omega}\rho\boldsymbol{g}\,\mathrm{d}\Omega}_{\text{Body force due to gravity acting on the fluid}}\end{aligned} \tag{1.4}$$

where $\boldsymbol{\sigma}$ is the stress tensor and $\rho\boldsymbol{g}$ is the volumetric force due to gravity. Using the divergence and localization theorem as before, one gets:

$$\boxed{\frac{\partial\rho\boldsymbol{u}}{\partial t} = -\operatorname{div}\rho(\boldsymbol{u}\otimes\boldsymbol{u}) + \operatorname{div}\boldsymbol{\sigma} + \rho\boldsymbol{g}} \tag{1.5}$$

This is the *momentum conservation equation* which reflects the force balance per unit volume.

If we take the dot product of the force balance with the speed vector, we get the kinetic energy balance:

$$\frac{\partial\frac{1}{2}\rho u^2}{\partial t} = -\operatorname{div}\frac{1}{2}\rho u^2\boldsymbol{u} + \operatorname{div}\boldsymbol{\sigma}\cdot\boldsymbol{u} + \rho\boldsymbol{g}\cdot\boldsymbol{u} \tag{1.6}$$

This equation does not bring additional information compared to (1.5) but it is useful as a *power* balance (work of the different forces per unit time) per unit volume. It allows to derive alternative forms of the energy balance.

1.1.3 Total energy conservation

In integral form:

$$\begin{aligned}\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\frac{1}{2}\rho u^2+\rho e\,\mathrm{d}\Omega}_{\text{Total energy increase rate in }\Omega} &= \underbrace{-\int_{\delta\Omega}(\frac{1}{2}\rho u^2+\rho e)\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}\delta\Omega}_{\text{Incoming energy flux due to fluid transport through }\delta\Omega} \quad \underbrace{-\int_{\delta\Omega}\boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}\delta\Omega}_{\text{Incoming energy flux due to molecular interactions at boundary }\delta\Omega} \\ &\quad \underbrace{-\int_{\delta\Omega}\boldsymbol{\sigma}\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}\delta\Omega}_{\text{Incoming energy flux due to stress tensor work at boundary }\delta\Omega} \quad \underbrace{+\int_{\Omega}\rho\boldsymbol{g}\cdot\boldsymbol{u}\,\mathrm{d}\Omega}_{\text{Power due to the work exerted by gravity on the fluid}}\end{aligned} \tag{1.7}$$

where e is the internal energy per fluid unit mass and $\boldsymbol{q}$ is the flux[2] due to thermal conduction. Using the divergence and localization theorem as before, one gets:

$$\frac{\partial(\frac{1}{2}\rho u^2+\rho e)}{\partial t} = -\operatorname{div}(\frac{1}{2}\rho u^2+\rho e)\boldsymbol{u} - \operatorname{div}\boldsymbol{q} + \operatorname{div}\boldsymbol{\sigma}\boldsymbol{u} + \rho\boldsymbol{g}\cdot\boldsymbol{u} \tag{1.8}$$

[2] $\boldsymbol{q}$ should rather be called a thermal surface density of flux, but we shall keep the shorter name.

which is the local expression (per unit volume and per unit time) of the total energy conservation.

By subtracting the kinetic energy balance, assuming that the stress tensor is symmetric, we can get the balance of internal energy (which is not in a conservative form):

$$\frac{\partial \rho e}{\partial t} = -\operatorname{div} \rho e \boldsymbol{u} - \operatorname{div} \boldsymbol{q} + \boldsymbol{\sigma} : \boldsymbol{\nabla} \boldsymbol{u} \tag{1.9}$$

1.1.4 Non-conservative form of the balance equations

We just wrote the balance equations in two forms:

- in a rather general integral form (1.1, 1.4 and 1.7);
- in a so-called conservative[3] local form (1.3, 1.5 and 1.8) which is more convenient to handle but less general because we have used the localization theorem which requires *continuity* of the argument under the integral sign.

A third form which is also local is called non-conservative. This form is convenient to use because of its conciseness. We first define the *material derivative* operator:

$$\frac{D}{Dt} s = \frac{\partial s}{\partial t} + \boldsymbol{u} \cdot \nabla s \tag{1.10}$$

$$\frac{D}{Dt} \boldsymbol{v} = \frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{\nabla} \boldsymbol{v}) \cdot \boldsymbol{u} \tag{1.11}$$

Physically speaking, this is the time derivative of a quantity as seen by an observer moving with the fluid at speed $\boldsymbol{u}$. Starting from the equations in conservative form, expanding the derivatives containing products and using the mass conservation equation, one can get:

$$\frac{D}{Dt} \rho = -\rho \operatorname{div} \boldsymbol{u} \tag{1.12a}$$

$$\rho \frac{D}{Dt} \boldsymbol{u} = \operatorname{div} \boldsymbol{\sigma} + \rho \boldsymbol{g} \tag{1.12b}$$

$$\rho \frac{D}{Dt} e = -\operatorname{div} \boldsymbol{q} + \boldsymbol{\sigma} : \boldsymbol{\nabla} \boldsymbol{u} \tag{1.12c}$$

These three forms of the balance equations are equivalent (i.e. we did not make any approximations to obtain them) provided the assumptions of continuity already mentioned hold. Many other forms of the balance equations are possible, see for example Candel's book [Can01].

1.2 Constitutive laws

The conservation equations that we have written are valid for all fluids. We will now restrict ourselves to the class of *incompressible Newtonian* fluids.

[3] A conservative equation can be written as : $\frac{\partial (\text{conserved quantity})}{\partial t} + \text{div}(\text{flux of the quantity}) = 0$

1.2.1 Constitutive law for the density

In general, the density ρ of a fluid depends on the thermodynamic state variables, such as pressure p and temperature T. For liquids, ρ may often be regarded as constant:

$$\boxed{\rho = \rho_0 = \mathrm{C}^{\mathrm{ste}}} \tag{1.13}$$

The fluid is then called *incompressible*. This allows us to simplify the local equation of mass conservation:

$$\boxed{\operatorname{div} \boldsymbol{u} = 0} \tag{1.14}$$

Relation (1.14) is also used to simplify some terms in the other balance equations.

1.2.2 Constitutive law for the stress tensor

For an isotropic Newtonian fluid, we write the stress tensor as:

$$\begin{aligned} \boldsymbol{\sigma} &= -p\mathsf{I} + \boldsymbol{\tau} &(1.15)\\ &= -p\mathsf{I} + \mu\left(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}^t\boldsymbol{u}\right) + \left(\kappa - \frac{2}{3}\mu\right)\operatorname{div}\boldsymbol{u}\mathsf{I} &(1.16)\end{aligned}$$

where p is the thermodynamic pressure, I is the unit tensor, $\boldsymbol{\tau}$ is the *viscous* part of the stress tensor, μ is the dynamic viscosity and κ is the dilatational viscosity.

Generally p is determined using a thermodynamic equation of state: $p = p(\rho, T)$. For incompressible fluids, this equation of state reduces to $\rho = \mathrm{C}^{\mathrm{ste}}$ which does not depend on the pressure p anymore. Thus, the pressure p becomes an unknown of the problem[4].

Dilatational viscosity κ is zero in the case of the perfect, monoatomic gas. For incompressible fluids, the term $\left(\kappa - \frac{2}{3}\mu\right)\operatorname{div}\boldsymbol{u}\mathsf{I}$ containing κ is zero. As a consequence, we can simplify the expression for the stress tensor:

$$\begin{aligned} \boldsymbol{\sigma} &= -p\mathsf{I} + \mu\left(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}^t\boldsymbol{u}\right) &(1.17)\\ &= -p\mathsf{I} + \mu\dot{\boldsymbol{\gamma}} &(1.18)\end{aligned}$$

where $\dot{\boldsymbol{\gamma}} = \boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}^t\boldsymbol{u}$ is called the strain tensor.

1.2.3 Constitutive law for the heat flux

For pure fluids or non diffusive mixtures, we consider that the heat flow $\boldsymbol{q}$ obeys the Fourier law:

$$\boxed{\boldsymbol{q} = -k\nabla T} \tag{1.19}$$

where k is the thermal conductivity.

[4] We will see in chapter 8 that, for incompressible fluids, p can be seen as the Lagrange multiplier in charge of ensuring mass conservation (1.14).

1.2.4 Constitutive law for the specific internal energy

For incompressible fluids, specific enthalpy h is preferred to specific internal energy e:

$$h = e + \frac{p}{\rho} \tag{1.20}$$

Indeed, in this case (p, T) is a better choice of state variables compared to (ρ, T), given that the density is assumed to be constant. We therefore write h in terms of these state variables[5]:

$$\begin{aligned} \mathrm{d}h &= \left(\frac{\partial h}{\partial T}\right)_p \mathrm{d}T + \left(\frac{\partial h}{\partial p}\right)_T \mathrm{d}p && (1.21)\\ &= c_p \mathrm{d}T + \left[\frac{1}{\rho} - T\left(\frac{\partial \frac{1}{\rho}}{\partial T}\right)_p\right] \mathrm{d}p && (1.22)\end{aligned}$$

where c_p is the heat capacity at constant pressure. In the case of incompressible fluids, this general expression simplifies because: $\left(\frac{\partial \frac{1}{\rho}}{\partial T}\right)_p = 0$. We can write a balance equation for the enthalpy, using the balance equation for the internal energy (1.12c) and the mass conservation equation (1.3):

$$\rho \frac{D}{Dt} h = -\operatorname{div} \boldsymbol{q} + \boldsymbol{\sigma} : \boldsymbol{\nabla u} + \frac{D}{Dt} p \tag{1.23}$$

1.3 Simplified form of the balance equations

Using balance equations for mass (1.14), momentum (1.12b) and enthalpy (1.23), constitutive laws (1.13), (1.17), (1.19) and (1.21), assuming that the coefficients in these laws (μ, k, c_p) are constant and using tensorial and vectorial identities given in [BAH87], we can write the balance equations as:

$$\begin{aligned} \operatorname{div} \boldsymbol{u} &= 0 && (1.24a)\\ \rho \frac{D}{Dt} \boldsymbol{u} &= -\nabla p + \mu \boldsymbol{\Delta u} + \rho \boldsymbol{g} && (1.24b)\\ \rho c_p \frac{D}{Dt} T &= k \Delta T + \frac{1}{2} \mu \left(\dot{\boldsymbol{\gamma}} : \dot{\boldsymbol{\gamma}}\right) && (1.24c)\end{aligned}$$

This form is computationally attractive because only three variables $\boldsymbol{u}, p, T$ and three equations remain. The final form that we retain is obtained by dividing (1.24b) (resp. (1.24c)) by the density ρ (resp. ρc_p) and by neglecting the heat term corresponding to viscous dissipation $\frac{1}{2}\mu\left(\dot{\boldsymbol{\gamma}} : \dot{\boldsymbol{\gamma}}\right)$:

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + \left(\boldsymbol{\nabla u}\right) \cdot \boldsymbol{u} &= -\nabla p^* + \nu \boldsymbol{\Delta u} + \boldsymbol{s_u} && (1.25a)\\ \nabla \cdot \boldsymbol{u} &= 0 && (1.25b)\\ \frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T &= \alpha \Delta T + s_T && (1.25c)\end{aligned}$$

[5] We assume that h depends only on these two variables and not on other variables: the constraint state of the system, for instance.

where we have defined: $p^* = p/\rho$, $\nu = \mu/\rho$ the kinematic viscosity, $\alpha = k/(\rho c_p)$ the thermal diffusivity, $\boldsymbol{s_u}$ a source term for the velocity equation and s_T a source term for the temperature equation.

We will try to solve this simplified form of the balance equations. Despite the simplifications the interaction between the different terms of these equations is the source of interesting physical behaviors and of numerical modeling challenges.

In the following chapters we shall treat the system (1.25) by keeping only two or three terms (chapters 2 to 8). Chapter 9 will focus on boundary conditions appropriate for system (1.25) together with the conservation properties of the finite-element discretization method that we will use. An algorithm for the solution of the total system (1.25) and its implementation will be given in chapter 10.

At each chapter head, we recall system (1.25) and we frame the terms which are treated in this chapter.

Chapter 2

Natural derivation of a finite element method

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u}) \cdot \boldsymbol{u} &= -\nabla p^* + \boxed{\nu \boldsymbol{\Delta u}} + \boldsymbol{s_u} \\ \nabla \cdot \boldsymbol{u} &= 0 \\ \frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T &= \boxed{\alpha \Delta T} + s_T \end{aligned}$$

This chapter will focus on the diffusive terms of the above equations. We bring forward the fundamental character of the Laplacian operator by considering the intimately related Dirichlet problem (section 2.1). This problem is expressed in the form of an optimization problem for an unknown scalar-valued function (variational form). We show that this variational form leads quite naturally to a numerical method: after a particular discretization step, we get the finite-element method for the Dirichlet problem.

We give some examples of solutions to the Dirichlet problem in section 2.2. There, we also discuss the regularity properties of these solutions.

The last section 2.3 will deal with the linear elasticity problem. The operator linked to this problem is another kind of Laplacian, acting on vector-valued functions, rather than on scalar-valued functions.

2.1 The Dirichlet problem

2.1.1 The continuous Dirichlet problem

An extrapolation problem

Let us look at the problem illustrated in figure 2.1. Let Ω be a closed domain with boundary $\delta\Omega$ and a function T_0 defined on $\delta\Omega$. We are looking for a function T, defined on the entire domain Ω, which extrapolates T_0, such that:

$$T|_{\delta\Omega} = T_0 \tag{2.1}$$

Obviously there are an infinite number of such functions, so we shall impose that the function be such that its gradient ∇T, which characterizes its spatial variations, be as

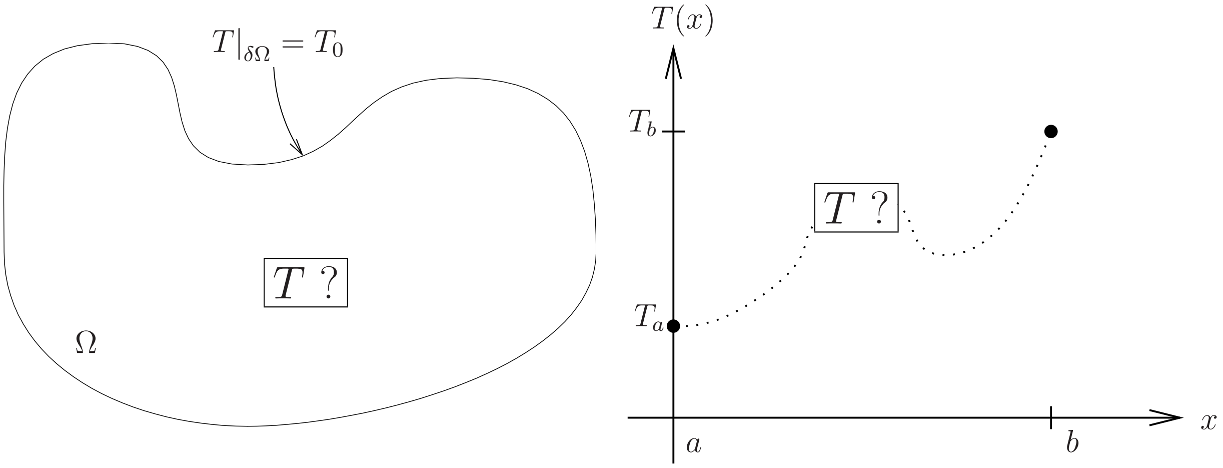


Figure 2.1: Extrapolation problem. Left: in two-dimensional space. Right: in one-dimensional space.

small as possible in a certain norm. In order to take into account the values of ∇T on the whole domain Ω, one should consider an average of ∇T on Ω (in integral sense).

The total variation

Let us see the way to formalize mathematically the above problem in one-dimensional space. A first idea can be to consider the following optimization problem: find a function T such that the real $J(T)$, called the total variation, is minimal:

$$\min_{T\in\mathcal{TV}_{\mathrm{D}}(\Omega)} J(T) = \min_{T\in\mathcal{TV}_{\mathrm{D}}(\Omega)} \int_a^b \left|\frac{\partial T}{\partial x}\right| \mathrm{d}x \tag{2.2}$$

$T(x) \in \mathcal{TV}_{\mathrm{D}}(\Omega)$ means that T is a function defined on Ω, that T belongs to $\mathcal{TV}(\Omega)$ and that the value of T is prescribed on the boundary: $T|_{x=a,b} = T_{a,b}$. A function T belongs to $\mathcal{TV}(\Omega)$ if the absolute value of its derivative is integrable, so that $J(T)$ is computable. J is called a real-valued *functional*, its input is a function and its output is a real number.

In fact, it is easy to see what the solutions of problem (2.2) will be: we evaluate the integral, first on the part of the domain where $\frac{\partial T}{\partial x} > 0$ and second, on the part of the domain where $\frac{\partial T}{\partial x} \leq 0$. Therefore $J(T)$ will be the sum of the upgoing height differences and of the downgoing height differences (figure 2.2 on the left). Thus $J(T)$ has minimal value $|T_b - T_a|$ and this value is obtained for any *monotone* function on $\Omega = [a, b]$. These functions are not necessarily continuous but they are necessarily included in the rectangle defined by the boundary conditions (figure 2.2 on the right).

Going back to the initial extrapolation problem, we can see that the minimization criterion (2.2) selects a set of solutions satisfying a monotonicity property, but we would rather like to have a unique solution.

The Dirichlet functional

Therefore we will choose another norm for measuring the gradient in order to obtain a unique solution. Consider the following optimization problem: find a function T such that

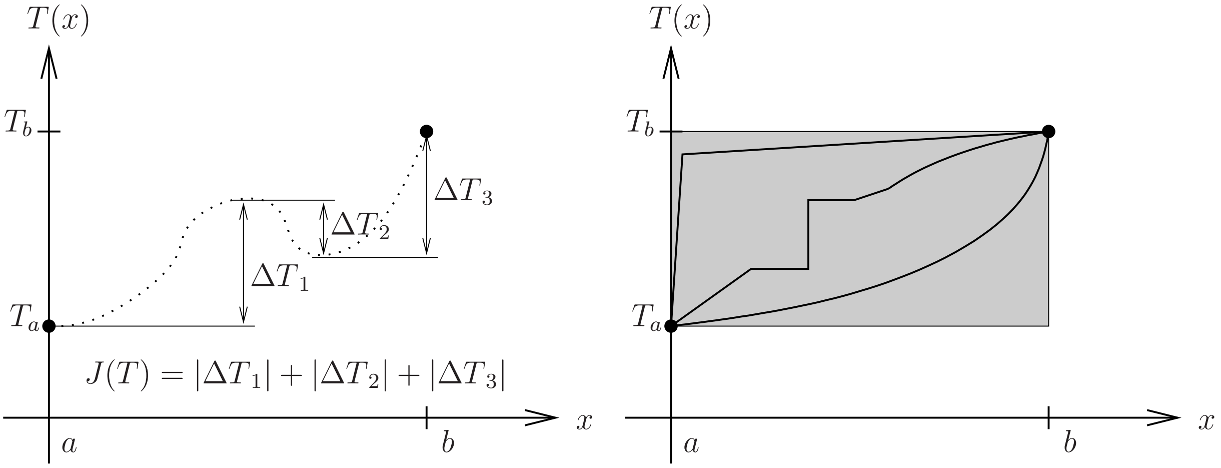


Figure 2.2: Left: the total variation $J(T)$ is the sum of absolute height differences. Right: examples of function T such that $T|_{x=a,b} = T_{a,b}$ and such that the total variation $J(T)$ is minimized.

the real $I(T)$ is minimum:

$$\min_{T \in \mathcal{H}^1_{\mathrm{D}}(\Omega)} I(T) = \min_{T \in \mathcal{H}^1_{\mathrm{D}}(\Omega)} \int_\Omega \frac{\alpha}{2} \|\nabla T\|^2 \,\mathrm{d}\Omega \tag{2.3}$$

$T(x) \in \mathcal{H}^1_{\mathrm{D}}(\Omega)$ means that T is a function defined on Ω, that T belongs to $\mathcal{H}^1(\Omega)$ and that the value of T is prescribed on the boundary:

$$T|_{\delta\Omega} = T_0 \tag{2.4}$$

A function T belongs to $\mathcal{H}^1(\Omega)$ if its gradient is square integrable, so that $I(T)$ is computable. α is a positive constant scalar coefficient with suitable physical units. The fact that this optimization problem, called the *Dirichlet problem*, is *well-posed* (see section 3.3 for this notion) is of fundamental importance. In particular, well-posedness implies that this problem admits a unique solution. The demonstration of this fact is outside the scope of these lecture notes: we will limit ourselves to a characterization of the solution, presupposing its existence.

The functional derivative

Under suitable assumptions on the regularity of I, a necessary condition for I to be minimum is to write that its derivative vanishes.

The derivative we will use here is the *functional derivative*. Recall that for a function T depending on a vector $\boldsymbol{x}$, we use the *directional derivative* along a given direction $\boldsymbol{y}$:

$$\mathrm{D}_{\boldsymbol{y}} T(\boldsymbol{x}) = \lim_{\epsilon \to 0} \frac{T(\boldsymbol{x} + \epsilon \boldsymbol{y}) - T(\boldsymbol{x})}{\epsilon} \tag{2.5}$$

For a functional I depending on a function T, we use the functional derivative along a "direction" U, which is, in fact, a function:

$$\delta_U I(T) = \lim_{\epsilon \to 0} \frac{\mathrm{d}}{\mathrm{d}\epsilon} I(T + \epsilon U) \tag{2.6}$$

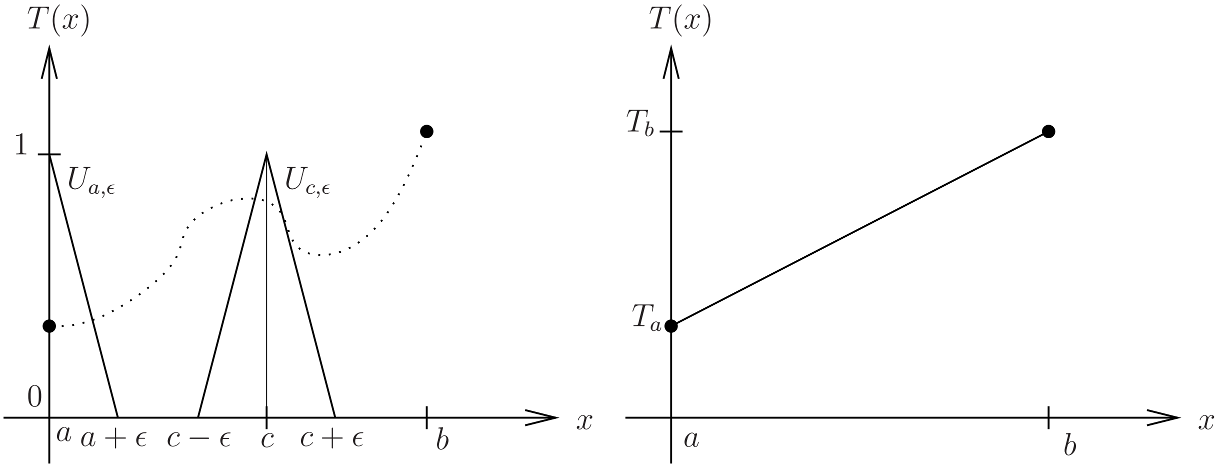


Figure 2.3: Left: examples of test functions U, called hat functions. Right: solution T of the Dirichlet problem (2.3)–(2.4) in 1D.

This formula reads: the derivative of the functional I at “point” T, along the “direction” U equals... In the formula, derivation with respect to ϵ is carried out while T and U are kept fixed.

The functional derivative is also a functional which, given two functions T and U, gives as a result a real value $\delta_U I(T)$.

Minimization condition

The minimization condition for problem (2.3) reads:

$$\delta_U I(T) = 0 \quad \forall U \in \mathcal{H}_0^1(\Omega) \tag{2.7}$$

Applying definition (2.6) for the functional derivative:

$$\delta_U I(T) = \lim_{\epsilon \to 0} \frac{\mathrm{d}}{\mathrm{d}\epsilon} \left(\int_\Omega \frac{\alpha}{2} \|\nabla T\|^2 \,\mathrm{d}\Omega + \epsilon \int_\Omega \alpha \nabla T \cdot \nabla U \,\mathrm{d}\Omega + \epsilon^2 \int_\Omega \frac{\alpha}{2} \|\nabla U\|^2 \,\mathrm{d}\Omega \right) \tag{2.8}$$

Finally:

$$\delta_U I(T) = \int_\Omega \alpha \nabla T \cdot \nabla U \,\mathrm{d}\Omega = 0 \quad \forall U \in \mathcal{H}_0^1(\Omega) \tag{2.9}$$

$U(\boldsymbol{x}) \in \mathcal{H}_0^1(\Omega)$ means that U, like T, is a function with square-integrable gradient. However, unlike T, the function U vanishes at the domain boundary: $U|_{\delta\Omega} = 0$. This is because we want that $T + \epsilon U$, like T, be in $\mathcal{H}_\mathrm{D}^1(\Omega)$ for all ϵ.

Interpretation of the minimization condition in 1D

In order to explain the meaning of the minimization condition (2.9) for the Dirichlet problem in 1D, let us choose a particular test function U. We take a piecewise-linear characteristic function, centered at $c \in]a, b[$, of width ϵ, denoted $U_{c,\epsilon}$ and presented on the left of figure 2.3. It is called a hat function.

We have:

$$\delta_{U_{c,\epsilon}} I(T) = \int_{c-\epsilon}^{c} \alpha \frac{dT}{dx} \frac{1}{\epsilon} \,\mathrm{d}x + \int_{c}^{c+\epsilon} \alpha \frac{dT}{dx} \frac{-1}{\epsilon} \,\mathrm{d}x = 0 \tag{2.10}$$

This means that the average values of $\alpha\frac{dT}{dx}$ on $[c-\epsilon, c]$ and $[c, c+\epsilon]$ are equal. Considering all the points $c \in [a+\epsilon, b-\epsilon]$ and passing to the limit $\epsilon \to 0^+$, we conclude:

$$\alpha\frac{dT}{dx} = \mathrm{C}^{\mathrm{ste}} \quad \text{sur} \quad]a, b[\tag{2.11}$$

where $\mathrm{C}^{\mathrm{ste}}$ is an arbitrary constant. The condition (2.11) leads to the fact that the solution of the 1D Dirichlet problem is the straight line connecting the points corresponding to the Dirichlet boundary conditions $T|_{x=a,b} = T_{a,b}$ (figure 2.3, right).

In addition, let us consider the half-hat function $U_{a,\epsilon}$. We can not choose this function in the framework of the Dirichlet problem because it does not vanish on the boundary and thus it is not in $\mathcal{H}_0^1(\Omega)$. Nonetheless, if we write the minimization condition, we obtain:

$$\delta_{U_{a,\epsilon}} I(T) = \int_a^{a+\epsilon} \alpha\frac{dT}{dx}\frac{-1}{\epsilon}\,\mathrm{d}x = 0 \tag{2.12}$$

This means that the average of $\frac{dT}{dx}$ on $[a, a+\epsilon]$ is zero. Passing to the limit $\epsilon \to 0^+$, we deduce:

$$-\alpha\frac{dT}{dx}\bigg|_a = 0 \tag{2.13}$$

This is another type of boundary condition, which appears when T is no longer prescribed on the boundary. It is called a *Neumann* or *natural* boundary condition.

Equivalent partial differential equation

In space dimension larger than one, the generalization of the interpretation made in the previous subsection involves the important *integration by parts* formulae. With these formulae, the minimization statement (2.7) can be rewritten into a more or less equivalent *partial differential equation* (PDE). The goal is to rewrite the minimization condition in the form:

$$\int_\Omega \mathrm{PDE} \times U\,\mathrm{d}\Omega = 0 \quad \forall U \tag{2.14}$$

From this equation we will obtain: PDE = 0 by virtue of the localization theorem. This theorem is also known as the fundamental lemma of the calculus of variations, see [Gur81].

Let us apply the method. Integrating (2.9) by parts gives:

$$\int_\Omega -\alpha\Delta T \times U\,\mathrm{d}\Omega + \int_{\delta\Omega} \alpha\nabla T \cdot \boldsymbol{n} \times U\,\mathrm{d}\delta\Omega = 0 \tag{2.15}$$

The boundary integral in equation (2.15) vanishes because U is zero at the boundary. This is due to T being already known on the boundary (*Dirichlet* or *essential* boundary condition). Nonetheless, the first term in the boundary integral is: $\alpha\nabla T \cdot \boldsymbol{n}$. This is a *flux* of the variable T through the boundary. The vanishing of this flux is a second possible boundary condition for the problem at hand: it is called a *Neumann* or *natural* boundary condition.

We will discuss in detail the expression of boundary conditions for this problem and several others in the dedicated chapter 9.

The first term in (2.15) allows us to express the PDE that we are actually solving:

$$-\alpha\Delta T = 0 \quad \text{sur} \quad \Omega \setminus \delta\Omega \tag{2.16}$$

Strong points of the variational method

One can legitimately ask if it is really necessary to introduce the optimization (or variational) setting for the Dirichlet problem and the associated fundamental tools (variational derivative, integration by parts) in order to obtain one of the simplest PDE (2.16). We answer this question positively for the following reasons:

1. The brevity of principle (2.3) from which we can infer not only PDE (2.16), but also the adequate Dirichlet and Neumann boundary conditions.

2. The natural occurence of the functional space $\mathcal{H}^1(\Omega)$ in which we seek the problem solution. This space is larger than the one that appears in PDE (2.16) which seems to require twice-differentiable functions. The principle (2.3) is thus more general.

3. The principle (2.3) seeks a solution T in $\mathcal{H}^1_{\mathrm{D}}(\Omega)$, a continuous functional space. However, we can apply exactly the same principle in order to find an approximate solution T_h in a smaller discrete subspace. We shall use this in the next section.

One can also make the following interpretation. Solving a Laplacian with Dirichlet boundary conditions answers the question: *What is the most regular*[1] *function satisfying the prescribed boundary conditions ?*

This interpretation is important because it highlights the regularizing role of the Laplacian operator. This role will appear several times in the following chapters:

- In chapter 4, we will find that a stable and an unstable scheme for the solution of a convection-diffusion equation differ only by a discretized Laplacian-like term.

- In chapter 5, we will find that an unsteady heat equation[2] with a singular initial condition at $t = 0$ admits a regular solution, for $t > 0$ in the continuous case, and after a sufficient number of time steps in the discrete case.

- In chapter 7, we will find that we need a small diffusive Laplacian term to exhibit the physical solution of an otherwise ill-posed problem described by the Burgers equation…

2.1.2 The discrete Dirichlet problem

Discrete function spaces

The functional space $\mathcal{H}^1_{\mathrm{D}}(\Omega)$ is the space in which the solution T of the Dirichlet problem (2.3) lives. It is an infinite-dimensional space and is thus too large to be handled by a computer! Therefore, we will seek an approximate solution T_h in a finite-dimensional subspace of $\mathcal{H}^1_{\mathrm{D}}(\Omega)$: $\mathtt{H}^1_{\mathrm{D}}(\Omega)$ of dimension N. Let N_i be a basis of $\mathtt{H}^1_{\mathrm{D}}(\Omega)$ so that we can write:

$$T_h(\boldsymbol{x}) = \sum_{i=1}^{N} T_i N_i(\boldsymbol{x}) \tag{2.17}$$

[1] By regularity here, we mean: having the smallest gradient, measured in the $\mathcal{L}^2(\Omega)$ norm. That is we seek a function as constant as possible.

[2] This equation involves a time derivative and a Laplacian term

The *finite-element method* consists in constructing the basis N_i in a particular way and then applying the minimization principle (2.3), thus looking for a solution in the restricted space $\mathtt{H}^1_{\mathrm{D}}(\Omega)$.

To build the basis N_i, one does a partition of the domain Ω (mesh) consisting of geometrically simple subdomains Ω_k (triangles, squares, tetrahedra). The N_i are then expressed as simple functions of the space coordinates on each subdomain Ω_k.

The problem unknowns are now the T_i. Often in the finite-element method the T_i correspond to the values of T_h at the mesh nodes. This property holds if and only if N_i is a *nodal basis* of $\mathtt{H}^1_{\mathrm{D}}(\Omega)$. A nodal basis satisfies the following properties:

- At node P_i with coordinates $\boldsymbol{x}_{P_i} : N_i(\boldsymbol{x}_{P_i}) = 1$;
- At any other node $P_j \neq P_i$ with coordinates $\boldsymbol{x}_{P_j} : N_i(\boldsymbol{x}_{P_j}) = 0$.

We give examples of such a nodal basis in chapter 3.

Discrete minimization condition

The minimization principle (2.3) restricted to a discrete functional space $\mathtt{H}^1_{\mathrm{D}}(\Omega)$ consists in finding a function T_h of that space minimizing the real number $I(T_h)$:

$$\min_{T_h \in \mathtt{H}^1_{\mathrm{D}}(\Omega)} I(T_h) = \min_{T \in \mathtt{H}^1_{\mathrm{D}}(\Omega)} \int_\Omega \frac{\alpha}{2} \|\nabla T_h\|^2 \,\mathrm{d}\Omega \tag{2.18}$$

We will necessarily have: $\min_{T_h \in \mathtt{H}^1_{\mathrm{D}}(\Omega)} I(T_h) \geq \min_{T \in \mathcal{H}^1_{\mathrm{D}}(\Omega)} I(T)$ since $\mathtt{H}^1_{\mathrm{D}}(\Omega) \subset \mathcal{H}^1_{\mathrm{D}}(\Omega)$. However, it can be shown for the Dirichlet problem (Lax-Milgram theorem, cf. [EG04]) that T_h converges to T when we increase the dimension N of $\mathtt{H}^1_{\mathrm{D}}(\Omega)$, that is when we refine the mesh.

The minimization condition (2.9) applied to the discrete problem reads:

$$\delta_{N_i} I(T_h) = \int_\Omega \alpha \nabla T_h \cdot \nabla N_i \,\mathrm{d}\Omega = 0 \quad \forall N_i \in \mathtt{H}^1_0(\Omega) \quad \text{i.e.} \quad \forall i \in \Omega \setminus \delta\Omega \tag{2.19}$$

Expanding T_h on the N_i basis, we obtain the *linear system* in the unknowns T_j:

$$\begin{aligned} \int_\Omega \alpha \nabla T_h \cdot \nabla N_i \,\mathrm{d}\Omega &= \sum_j T_j \Big(\int_\Omega \nabla N_j \cdot \nabla N_i \,\mathrm{d}\Omega \Big) \\ &= \sum_j T_j \mathsf{R}_{ji} = \mathsf{R}\underline{T} = 0 \quad \forall i \in \Omega \setminus \delta\Omega \end{aligned} \tag{2.20}$$

$\underline{T}$ is the vector grouping the unknowns T_j. The matrix R_{ij} is often called the *rigidity* matrix due to the fact that the finite element method has historically been applied first on continuum mechanics problems (see next section 2.3).

To conclude on the Dirichlet problem, we see that applying a minimization principle and discretizing a functional space led us in a very simple way to a linear system that lends itself well to a computer solution.

The discretization process just performed is summarized in table 2.1.

We underline an important point: once the discretization is performed, we generally cannot use integration by parts anymore in the discrete minimization condition (2.19), because the discrete basis functions N_i are not regular enough (not twice-differentiable).

Continous		Discrete		Linear system
$T \in \mathcal{H}^1_{\mathrm{D}}(\Omega)$		$T_h = \sum_{i=1}^{N} T_i N_i$		
$\min_T \int_\Omega \frac{\alpha}{2} \lVert \nabla T \rVert^2 \, \mathrm{d}\Omega$	$\Rightarrow$	$\min_{T_i} \int_\Omega \frac{\alpha}{2} \lVert \nabla T_h \rVert^2 \, \mathrm{d}\Omega$		
$\downarrow$		$\Downarrow$		
$\int_\Omega \alpha \nabla T \cdot \nabla U \, \mathrm{d}\Omega = 0 \quad \forall U$		$\int_\Omega \alpha \nabla T_h \cdot \nabla N_j \, \mathrm{d}\Omega = 0 \quad \forall N_j$	$\Rightarrow$	$\mathsf{R}\underline{T} = 0$
$\downarrow$				
$-\alpha \Delta T = 0$				

Table 2.1: Discretization of the Dirichlet problem: follow the double arrows ⇒. From the Dirichlet problem to the Laplace equation: follow the simple arrows →.

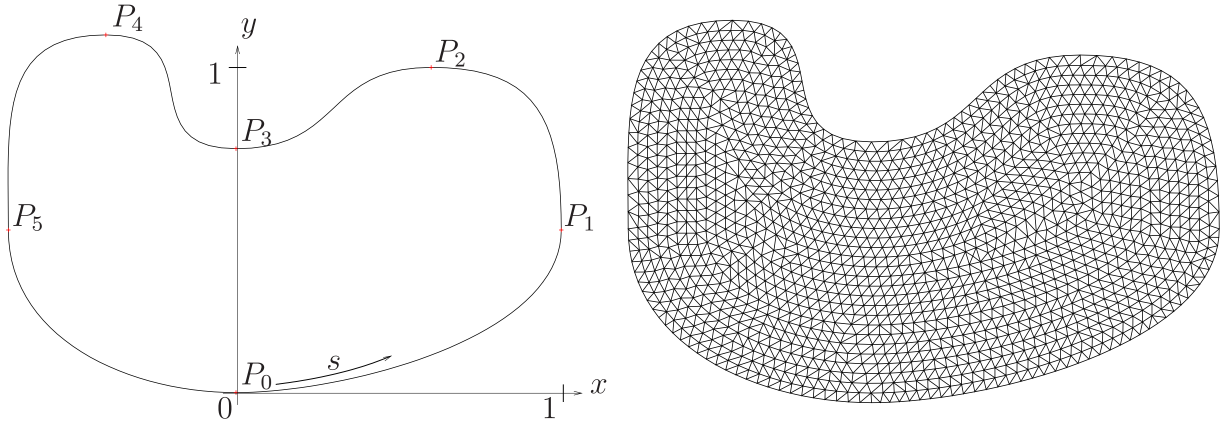


Figure 2.4: Left: Boundary $\delta\Omega$ with some control points. Right: triangular mesh of the domain Ω.

As a consequence, T_h is often called a *weak solution*[3] of the discrete Dirichlet problem. We will return to this important point in the chapter dedicated to the boundary conditions (chapter 9). In fact, because we generally cannot use integration by parts, it will be seen that the Neumann boundary conditions are also weakly prescribed.

2.2 Examples of discrete solutions

2.2.1 Regular solutions

We illustrate the behavior of solutions to problem (2.3)–(2.4) in 2D, with plots of discrete solutions obtained by solving the linear system (2.20). Figure 2.4 shows the boundary and the mesh of the chosen domain Ω.

We first choose, for the boundary condition, a Gaussian function of the curvilinear abscissa s centered at point P_1 :

$$T_h|_{\delta\Omega}(s) = e^{-\left(\frac{s-s_{P_1}}{0.15}\right)^2} \tag{2.21}$$

[3] The name weak solution is not pejorative. It means that the *weak formulation* of the Dirichlet problem (2.9) is more general and thus admits more solutions than its *strong formulation* (2.16).

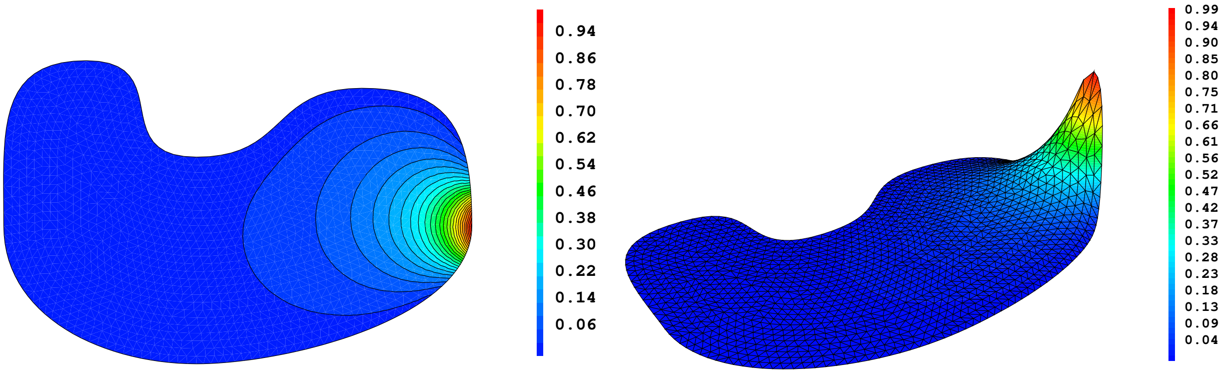


Figure 2.5: Numerical solution T_h of the problem (2.18)–(2.21). Left: isovalues of T_h. Right: 3D representation of T_h; $z = T_h(x, y)$.

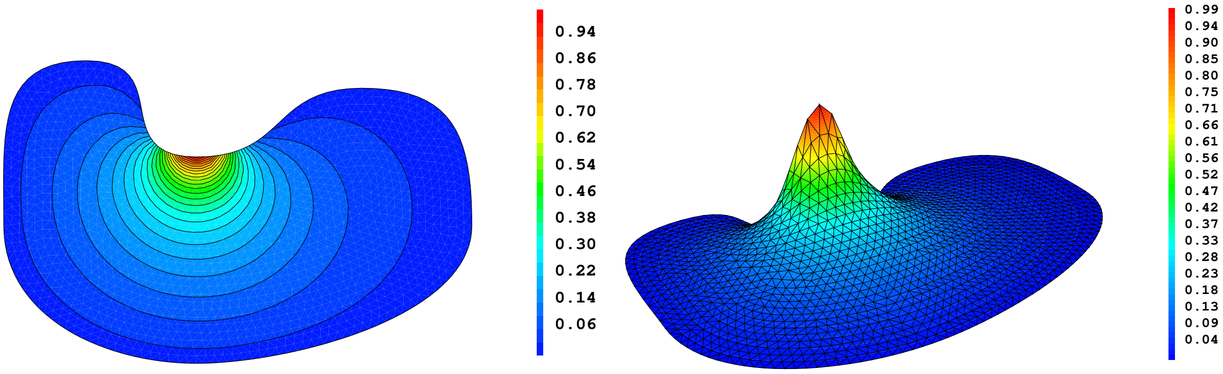


Figure 2.6: Numerical solution T_h of the problem (2.18)–(2.22). Left: isovalues of T_h. Right: 3D representation of T_h ; $z = T_h(x, y)$.

We obtain the numerical solution shown on figure 2.5. We notice that the solution T_h is regular and diminishes quickly away from point P_1 such that $\|\nabla T\|^2$ is very low almost everywhere in Ω.

Now if we choose, for the boundary condition, a Gaussian curve centered at point P_3 :

$$T_h|_{\delta\Omega}(s) = e^{-\left(\frac{s-s_{P_3}}{0.15}\right)^2} \tag{2.22}$$

We obtain the numerical solution shown on figure 2.6. This time we notice that the solution T_h becomes smaller a little slower with distance from the point P_3: this is because P_3 is located in a concave part of $\delta\Omega$.

2.2.2 Singular solutions

We have already said that the solutions of the Dirichlet problem were generally very regular due to the fact that they are of minimal gradient. However, there are at least three cases of interest where these solutions can exhibit singularities:

1. The Dirichlet boundary condition $T|_{\delta\Omega}$ is itself a singular function;
2. The boundary $\delta\Omega$ has reentrant corners;

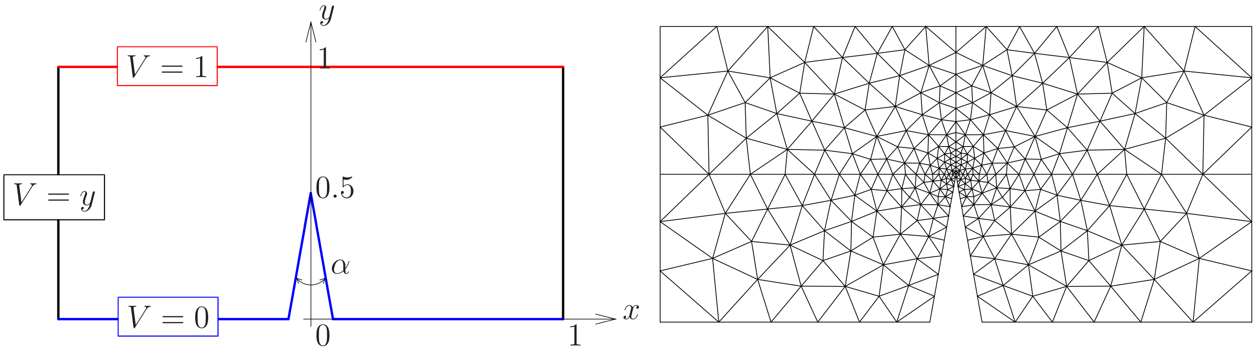


Figure 2.7: Left: Geometry of the domain Ω and prescribed boundary conditions for the lightning rod case. Right: triangular mesh of the domain Ω.

3. The boundary conditions change in nature (Dirichlet–Neumann transition) on the boundary $\delta\Omega$.

In all these cases, the singularity comes from the boundary. This is expected because the solutions of the Dirichlet problem are completely determined by the shape of the boundary and the prescribed boundary conditions.

Peak singularity: the lightning rod

Let us consider the second case in which the computational domain Ω is displayed in figure 2.7. The boundary $\delta\Omega$ of the domain has a very concave part when $\alpha \to 0$. The bottom part of the boundary can be thought as a lightning rod connected to the ground so that its electric potential $V = 0$. The upper part of the boundary can be thought as a cloud at non-zero potential $V = 1$. We assume that the electric potential V satisfies the following electrostatic equation: $\Delta V = 0$, that is V is a solution of the Dirichlet problem (2.3)–(2.4).

The discrete solution V_h of this problem is shown on figure 2.8. One can notice that the isovalues of V_h are very close to each other near the peak of the rod. This means that the gradient of the electric potential, i.e. the electric field is very large in this zone: this is called the *peak effect*. It provides a first explanation of how the lightning rod works: air will be ionized near the peak, due to the large electric field. In turn, this will generate a channel with high electric conductivity where the lightning will preferentially strike.

Dirichlet-Neumann singularity

Let us now consider the case of the lightning rod with $\alpha = 0$. The geometry and boundary conditions are symmetrical with respect to the $x = 0$ axis. Thus, we will consider only one half of the domain (see figure 2.9).

Now, there is a change in boundary conditions at point E, on the left boundary AD:

- Homogeneous Dirichlet boundary condition $V = 0$ on the lightning rod AE;
- Homogeneous Neumann boundary condition $\frac{\partial V}{\partial n} = 0$ on ED. This condition implies that the flux of the unknown V through ED is zero, by symmetry.

As in the lightning rod case, the solution of this problem exhibits a singularity at point E. One can show [SF88] that, in the vicinity of point E, the solution behaves like

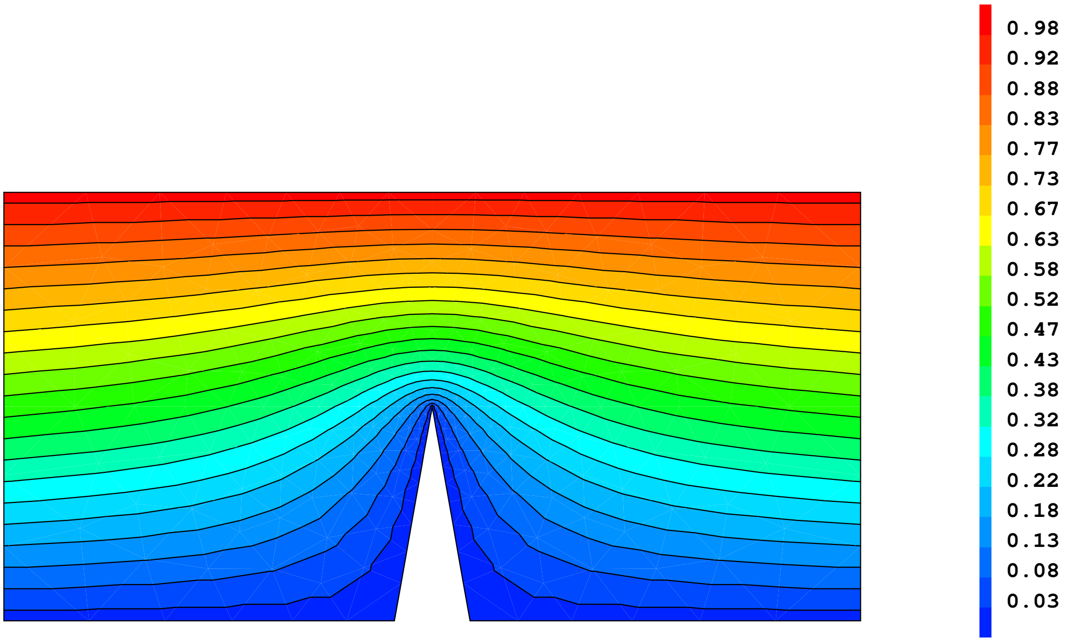


Figure 2.8: Isovalues of the discrete solution V_h for the lightning rod case.

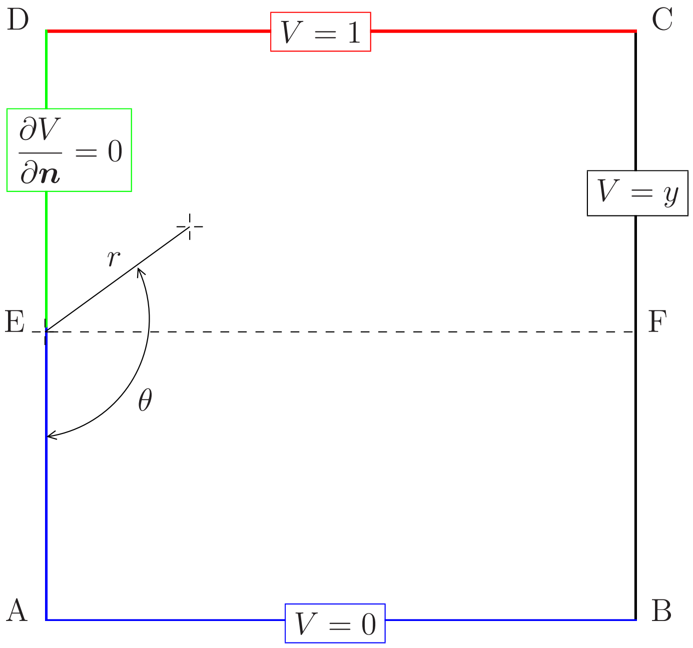


Figure 2.9: The half-domain Ω and prescribed boundary conditions for the lightning rod case, figure 2.7, with $\alpha = 0$.

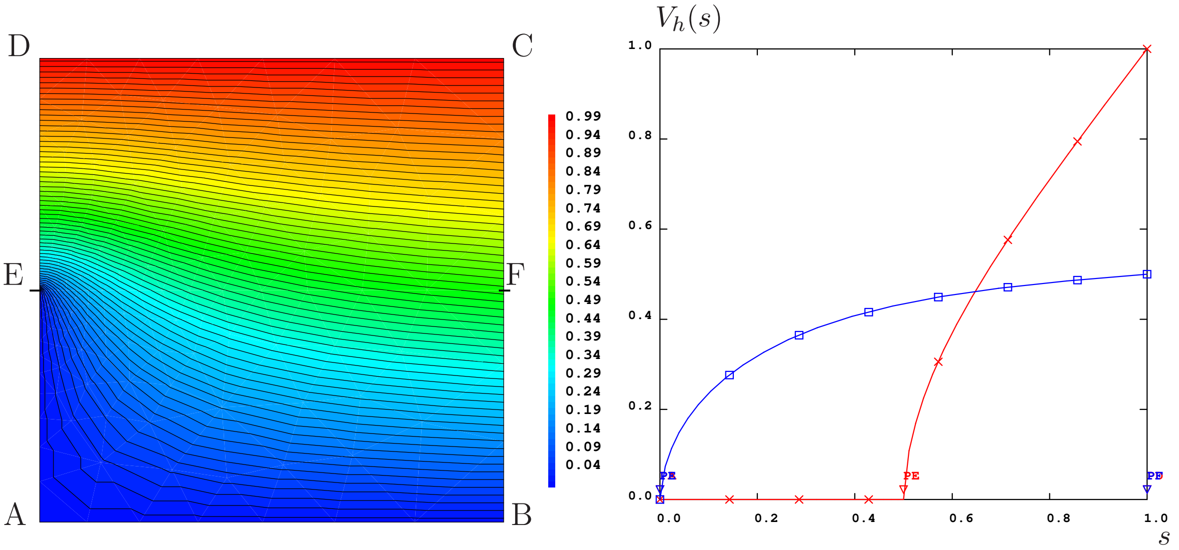


Figure 2.10: Left: Isovalues of the discrete solution V_h for the lightning rod case with $\alpha = 0$. Right: profiles of the discrete solution V_h versus the curvilinear abscissa s. ×: along the line AD. □: along the line EF.

$\sqrt{r}\sin\frac{\theta}{2}$, r being the distance to point E. The right of figure 2.10, showing the profiles of the electric potential V along the AD and EF lines, illustrates this behavior.

The problem solution V is continuous but not derivable at point E. However, its gradient ∇V *is* still square-integrable on Ω! Going to cylindrical coordinates (r, θ) centered at E, the $\mathcal{H}^1(\Omega)$ norm of the solution is:

$$\int_\Omega \|\nabla V(r,\theta)\|^2 \,\mathrm{d}\Omega = \int_\Omega \left\{ \left(\frac{\partial V}{\partial r}\right)^2 + \left(\frac{1}{r}\frac{\partial V}{\partial \theta}\right)^2 \right\} r\mathrm{d}r\mathrm{d}\theta \tag{2.23}$$

$$= \int_\Omega \left\{ \left(C(\theta)\frac{1}{\sqrt{r}}\right)^2 + \left(D(\theta)\frac{\sqrt{r}}{r}\right)^2 \right\} r\mathrm{d}r\mathrm{d}\theta \tag{2.24}$$

which is well-defined.

We can get a qualitative feeling as of why there is a singularity of the gradient of V at the change in boundary conditions by examining closely the isovalues of the unknowns in the vicinity of point E. As seen on the left-hand side of figure 2.10:

- Above the point E, the isovalues are perpendicular to the boundary AD, because of the symmetry condition $\frac{\partial V}{\partial n} = 0$;
- Under the point E, the isovalues are parallel to the boundary AD, the boundary part AE being itself the isovalue $V = 0$.

At the point E, the isovalues should be both parallel and perpendicular to the boundary, hence the singularity.

Singularity influence

As you will see in your final projects, in many practical cases of interest, one or more singularities will be displayed. It is important to identify and locate them since they can have a negative influence on the precision and convergence order of most of the discretization methods. Once identified, a basic remedy is to refine the mesh in the vicinity of the singularity. For example, this is what we have done for the lightning rod case (see right hand side of figure 2.7).

For a more detailed discussion about singularities in relationship with the finite-element method, one can refer to chapter 8 of Strang and Fix's book [SF88].

2.3 The linear elasticity problem

In this section, we consider a slightly more complex problem that can be derived from a minimisation principle. The following principle is used in the context of solid mechanics in order to model the small deformations of a linear elastic material:

$$\min_{\boldsymbol{u}\in\left(\mathcal{H}^1_{\mathrm{D}}(\Omega)\right)^3} I(\boldsymbol{u}) = \min_{\boldsymbol{u}\in\left(\mathcal{H}^1_{\mathrm{D}}(\Omega)\right)^3} \frac{1}{2}\int_\Omega \left\{\mu\left\|\frac{\boldsymbol{\nabla u}+\boldsymbol{\nabla}^t\boldsymbol{u}}{2}\right\|^2 + \lambda(\nabla\cdot\boldsymbol{u})^2 - \boldsymbol{f}\boldsymbol{u}\right\}\mathrm{d}\Omega \tag{2.25}$$

where λ and μ are the Lamé coefficients.$\boldsymbol{f}$ is a body force applied on the solid (gravity for example) and $\boldsymbol{u}$ are the solid's displacements with respect to a chosen reference configuration. The quantity $I(\boldsymbol{u})$ to be minimized is called the *elastic energy*. We have chosen prescribed displacements boundary conditions on $\delta\Omega$. The following formulae relate the Lamé coefficients to the Young modulus E and the Poisson coefficient ν:

$$E = \mu\frac{3\lambda+2\mu}{\lambda+\mu} \quad ; \quad \nu = \frac{1}{2}\frac{\lambda}{\lambda+\mu} \tag{2.26}$$

The minimization condition reads:

$$\delta_v I(\boldsymbol{u}) = \int_\Omega \{\mu\boldsymbol{\nabla u}:\boldsymbol{\nabla v} + (\lambda+\mu)\,(\nabla\cdot\boldsymbol{u})\,(\nabla\cdot\boldsymbol{v}) - \boldsymbol{f}\boldsymbol{v}\}\,\mathrm{d}\Omega = 0 \quad \forall\boldsymbol{u}\in\left(\mathcal{H}^1_0(\Omega)\right)^3 \tag{2.27}$$

This condition is known as the *virtual work principle*.

From this condition, we can derive the Navier equations of linear elasticity:

$$\mu\boldsymbol{\Delta u} + (\lambda+\mu)\nabla\,(\nabla\cdot\boldsymbol{u}) + \boldsymbol{f} = 0 \tag{2.28}$$

These equations can be rewritten in the following more familiar mixed form involving the stress tensor $\boldsymbol{\sigma}$:

$$\begin{aligned} \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}+\boldsymbol{f} &= 0 & (2.29)\\ \boldsymbol{\sigma} &= \lambda(\nabla\cdot\boldsymbol{u})\mathsf{I} + \mu\left(\boldsymbol{\nabla u}+\boldsymbol{\nabla}^t\boldsymbol{u}\right) & (2.30)\end{aligned}$$

Equation (2.29) expresses a local force equilibrium (i.e. momentum conservation) and equation (2.30) is a constitutive law for the stress tensor $\boldsymbol{\sigma}$.

This linear elasticity problem is not unrelated to fluid mechanics. In fact, we will show in chapter 8 that when the first Lamé coefficient $\lambda\to\infty$, the linear elasticity system gives the incompressible Stokes equations which is a limiting case of the incompressible Navier-Stokes equations for small Reynolds number.

2.4 Summary

In this chapter we have focused on how to derive a finite element method for the Dirichlet problem from a first variational principle because we feel it proceeds in a most natural way. We will go back to the very important subject of boundary conditions in chapter 9. It is well-known that variational principles play a very important role in physics and we found it useful to link these rather theoretical principles with concrete applications, such as the finite element method and the calculation of approximate solutions of partial differential equations by computers.

We can give other examples of physical phenomena which can be derived from a variational principle in fluid mechanics: for example, surface tension produces capillary forces that tend to minimize the surface they act upon,etc…

In the field of incompressible fluid mechanics at least two difficulties differ from the variational setting we have just examined:

- some terms in the equations cannot be derived naturally from a variational principle: this is the case of the first, convective, term in the scalar convection-diffusion equation $\boldsymbol{u} \cdot \nabla T - \alpha \Delta T = 0$;
- some optimization principles more complex than simple minimization must sometimes be invoked. This is the case for the *incompressible Stokes* problem which can be viewed as a constrained minimization (also called saddle-point) problem.

Because of these difficulties, the finite element method applies somewhat less naturally to incompressible fluid mechanics: we will have to use with care some supplementary mathematical tools.

In chapter 3 we will generalize the finite element method to a class of partial differential equations larger than those that derive from a variational principle. In chapter 4 we will deal with the difficulties linked to the convective term on a scalar stationary convection-diffusion problem. The difficulties related to the divergence constraint for the incompressible Stokes problem will be the subject of chapter 8.

Chapter 3

The finite element method

$$\begin{aligned}\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u}) \cdot \boldsymbol{u} &= -\nabla p^* + \nu \boldsymbol{\Delta u} + \boldsymbol{s_u}\\ \nabla \cdot \boldsymbol{u} &= 0\\ \frac{\partial T}{\partial t} + \boxed{\boldsymbol{u} \cdot \nabla T} &= \boxed{\alpha \Delta T} + s_T\end{aligned}$$

3.1 Introduction

In this chapter we describe a discretization method that can be applied to a more general class of partial differential equations than those that derive from a variational principle, as seen in the previous chapter. This method is called the weighted residual method (section 3.2). As an example, we show how to apply this method to the scalar convection-diffusion equation, which will be our model problem for chapter 4.

The important concepts of convergence and stability we present in section 3.3. Indeed, unlike in chapter 2, convergence of discrete solutions obtained with the method of weighted residual is not automatic.

Eventually, applying the weighted residual method to a particular set of basis function, we obtain the finite element method. A short reminder of mathematical tools useful for the practical implementation of the finite element method in a computer program follows in section 3.4, 3.5 and 3.6.

3.2 The weighted residual method

We apply the method of the previous chapter by following a different path: we start from the partial differential equation in order to get the weak formulation. For example, let us start with the steady convection-diffusion equation where $\boldsymbol{u}$ is a given vector field:

$$\boldsymbol{u} \cdot \nabla T - \alpha \Delta T = 0 \tag{3.1}$$

We shall call the mathematical expression on the left of the equal sign the residual R. The residual is an *operator*: it maps a given function T to another function $(\boldsymbol{u} \cdot \nabla T - \alpha \Delta T)$. Solving equation (3.1) amounts to finding a function T that makes the residual to vanish.

In the previous chapter we pointed out that if we are looking for the solution T using a discretization method, we usually can, at best, find an approximate solution T_h in a finite-dimensional functional space $\mathtt{F}_{\text{pri}}$. Let N_i ($i \in [1, N]$) be a basis of the $\mathtt{F}_{\text{pri}}$ space. We can then write the following expression for T_h:

$$T_h(\boldsymbol{x}) = \sum_{i=1}^{N} T_i N_i(\boldsymbol{x}) \tag{3.2}$$

The unknowns for the problem at hand are now the N coefficients T_i, which are called the *degrees of freedom (dof)*. We must emphasize that for most of the time, we have:

$$R(T_h) = \boldsymbol{u} \cdot \nabla T_h - \alpha \Delta T_h \neq 0 \tag{3.3}$$

because T_h is only an approximation of the true solution T of problem (3.1). In order to find suitable values for the degrees of freedom T_i, several methods can be used. For instance, one could require one of the following:

- that the residual $R(T_h)$ vanishes at N suitably chosen points as in the *collocation* method;
- that the residual $R(T_h)$ be minimal in a chosen norm. We get methods of the *least-squares* type if we choose the $\mathcal{L}^2(\Omega)$ norm;
- that the residual $R(T_h)$ be *orthogonal* to a (dual) functional space $\mathtt{F}_{\text{dua}}$. In order to achieve this, we must work in a functional space where a scalar product can be defined (Hilbert space). This type of method is called *weighted residual* method.

In general the weighted residual method is applied in functional spaces that are subspaces of $\mathcal{L}^2(\Omega)$ with the scalar product associated to the $\mathcal{L}^2(\Omega)$ norm:

$$\langle u, v \rangle = \int_\Omega uv \, \mathrm{d}\Omega \tag{3.4}$$

When one makes the choice $\mathtt{F}_{\text{pri}} = \mathtt{F}_{\text{dua}}$, the weighted residual method is called a *Galerkin* method. We then have:

$$\forall j \in [1, N] \quad \langle R(T_h), N_j \rangle = 0 \tag{3.5}$$

Developing the expressions for R and T_h, using the linearity of the integral and integrating by parts on the Laplacian, one gets:

$$\forall j \in [1, N] \quad \sum_{i=1}^{N} T_i \left(\int_\Omega \boldsymbol{u} \cdot \nabla N_i N_j + \alpha \nabla N_i \nabla N_j \, \mathrm{d}\Omega - \int_{\delta\Omega} \alpha \nabla N_i \cdot \boldsymbol{n} N_j \, \mathrm{d}\delta\Omega \right) = 0 \tag{3.6}$$

After computing the volume integral (the boundary integral vanishes if we prescribe the degrees of freedom T_i on the boundary), we write:

$$\forall j \in [1, N] \quad (\mathsf{C} + \mathsf{R})_{ji} T_i = 0 \tag{3.7}$$

where C is the convection matrix and R the rigidity matrix seen in the previous chapter (equation (2.20)). Once again, we eventually get a linear system which can be solved with a computer. The discretization process just described is summarized in table 3.1. This table is to be compared with table 2.1 of the previous chapter.

Compared to the previous chapter method, the weighted residual method seems more general because it can be applied to any partial differential equation, not just those that can be derived from a variational principle. However, it seems to have some drawbacks:

Continous		Discrete		Linear system
$T \in \mathcal{H}^1_{\mathrm{D}}(\Omega)$		$T_h = \sum_{i=1}^{N} T_i N_i$		
		$\boxed{\text{No minimization principle}}$		
$\int_\Omega \boldsymbol{u} \cdot \nabla T U \,\mathrm{d}\Omega$ $+ \int_\Omega \alpha \nabla T \cdot \nabla U \,\mathrm{d}\Omega = 0 \quad \forall U$ $\Uparrow$ $\boldsymbol{u} \cdot \nabla T - \alpha \Delta T = 0$	$\Rightarrow$	$\int_\Omega \boldsymbol{u} \cdot \nabla T_h N_j \,\mathrm{d}\Omega$ $+ \int_\Omega \alpha \nabla T_h \cdot \nabla N_j \,\mathrm{d}\Omega = 0 \quad \forall N_j$	$\Rightarrow$	$(\mathsf{C} + \mathsf{R})\, \underline{T} = 0$

Table 3.1: Discretization of the convection-diffusion equation with a Galerkin method: follow the double arrows ⇒. Compare with table 2.1.

1. it gives no indication as to how to choose the primal functional space in which we look for the solution T and its approximation T_h, and the dual functional space for the continuous and discrete residuals;

2. it gives no indication as to what type of boundary conditions for the problem are suitable;

3. carrying out integration by parts on the Laplacian, thus lowering the regularity requirements on the N_i functions, seems to be quite artificial.

Moreover, the fact that we can apply the method does not guaranty that it will work. By work, we mean that the method should result in an invertible linear system *and* that the discrete solution T_h of this system approaches correctly the exact solution T. That is, we have to study the *convergence* of the method.

3.3 Convergence and stability equivalence

We state *Lax*'s equivalence theorem also known as the *fundamental theorem of numerical analysis*, following [Str07]:

Theorem 1 (Lax) *Stability is equivalent to convergence, for a consistent approximation to a well-posed linear problem.*

Let $Lu = f$ be the linear problem at hand where L denotes the linear operator associated with the problem, f is the problem input data (boundary conditions included) and u is the problem unknowns. Let $L_h U_h = f_h$ be the corresponding discrete problem.

Consistency $f_h \to f$ and $L_h\tilde{u} \to L\tilde{u}$ for smooth functions $\tilde{u}$ when $h \to 0$. This means that the problem L and the problem data f are correctly approximated.

Well-posed problem The inverse of L exists and is bounded: $\|u\| = \|L^{-1}f\| \leq C\|f\|$. Existence of the inverse of L means that for a given f, there is a unique solution u of the problem.

Stability The discrete inverses ${L_h}^{-1}$ are uniformly bounded: $\|U_h\| = \left\|{L_h}^{-1}f_h\right\| \leq C'\|f_h\|$, $\forall h$. Loosely, this means that the modulus of the smallest eigenvalue of L_h, λ_{1h}, is bounded below by a strictly positive constant independent of h.

Convergence $u - U_h \to 0$ when $h \to 0$.

We notice that the definition of the stability of the discrete problem is quite similar to the definition of the well-posedness of the continuous problem with an added difficulty coming from the discretization process h: the discrete problem must be *uniformly* well-posed.

A sketch of the theorem proof consists in adding and substracting $L_h^{-1}Lu = L_h^{-1}f$ from $u - U_h$:

$$u - U_h = L_h^{-1}\left(L_h u - Lu\right) + L_h^{-1}\left(f - f_h\right) \to 0 \tag{3.8}$$

Consistency controls the parenthesized terms (they tend to zero). Stability controls the discrete inverses ${L_h}^{-1}$ which act upon them. Well-posedness controls approximation of u by smooth functions $\tilde{u}$.

We point out that Lax's equivalence theorem is quite general: the primal and dual functional spaces and associated norms are not yet prescribed. We shall see in section 4.5.2 that choosing a norm can have important consequences. Stronger results than Lax's theorem can often be obtained for particular discretization methods: notably, results related to their convergence order.

Usually the weighted residual method is consistent for linear well-posed problems, like the scalar convection-diffusion and Stokes' problem, with suitable boundary conditions. However stability is not always verified. This fact will be brought forward by numerical examples in the following chapters: chapter 4 - for the convection-diffusion equation and chapter 8 - for Stokes' problem.

When the problem we want to approximate is non-linear, things could be more complicated. We will illustrate this on one of the most simple example of non-linear problem, Burgers' equation, in chapter 7. In the case of incompressible Navier-Stokes' equations in 3D, there is no general existence theorem[1] concerning their solution: this means that we do not know if even the continuous problem is well-posed. Of course, this does not prevent ourselves from seeking numerical solutions! Moreover, deep questions arise. Some of them are related to the transition to turbulence and to the modeling of turbulent flows (which may lead to better posed problems). These matters are the subject of current research. We shall mention that the Clay Mathematics Institute has set a 1000000$ prize for partial answers to these questions. On our more modest level, it is a sane practice to always put numerical results for fluid dynamics problem into question. In general, we can say that comparison with experimental data remains necessary.

[1] More precisely, mathematicians are looking for existence conditions such that solutions of Navier-Stokes' equations do not explode (i.e. stays bounded in the $\mathcal{L}^\infty$-norm) in finite time.

3.4 Basis functions

Once the discrete problem has been written (2.20) or (3.6), one has to choose a particular set of basis functions N_i. In one space dimension, several choices are possible: Fourier series decomposition, orthogonal polynomial basis (Legendre or Chebychev)... In space dimension higher than one, it is easy to generalize these basis on a cartesian mesh, but not so easy on an unstructured mesh.

In the nodal finite element method, the N_i basis is constructed in the following way:

Mesh The considered domain Ω is partitioned into geometrically simple elements: $\Omega = \bigcup_k \Omega_k$;

Nodal basis The degrees of freedom T_i coincide with the values of the function T_h on some nodes P_i of the mesh: $T_h(P_i) = T_i$;

Element-wise polynomial basis The restriction of any basis function N_i to an element Ω_k is polynomial in the space coordinates.

Rather than going into the details of the computation of the basis functions (see for instance [DLT12]), we will plot a particular N_i function of the simplest discrete functional spaces in one and two space dimensions, together with the interpolation T_h of an arbitrary function T in the same discrete functional space.

Due to the choice of element-wise polynomial basis functions, the discrete V subspace of $\mathcal{L}^2(\Omega)$ (square-integrable functions) will consist of element-wise continuous functions. However these functions can be discontinuous at the boundary between two elements. On the contrary, discrete W subspace of $\mathcal{H}^1(\Omega)$ (functions whose gradient is square-integrable) will consist of globally continuous functions.

The simplest discrete subspace $V_0 \subset \mathcal{L}^2(\Omega)$ is the space of element-wise constant functions. A basis of V_0 is the set of indicator functions for the elements (see figure 3.1 and 3.2).

The simplest discrete subspace $W_1 \subset \mathcal{H}^1(\Omega)$ is the space of element-wise linear functions. A basis of W_1 is the set of hat functions built on the elements' vertices (see figure 3.3 and 3.4).

The three properties used in building the N_i basis in the finite element method lead to interesting consequences. The second property (nodal basis) implies:

$$N_i(P_j) = \delta_{ij} \tag{3.9}$$

δ_{ij} is Kronecker's symbol. Its value is 1 if $i = j$ and 0 otherwise. Then, using the third property (element-wise polynomial basis), we can show that N_i vanishes on every element not having P_i as a vertex. This *locality* property is valuable for the subsequent computer treatment. In fact, a generic term A_{ij} of a discretized problem total matrix A will be equal to zero if vertices P_i and P_j are not part of the same element. This is because A_{ij} is an integral involving the two local functions N_i and N_j or their derivatives which are also local. Thus, the matrix A will mostly have zero terms: it is called a *sparse* matrix. On a computer, we can store only the non-zero terms in order to save memory.

However we notice that the N_i set of basis functions for W_1 is not an orthogonal basis:

$$\mathsf{M}_{ij} = \langle N_i, N_j \rangle = \int_\Omega N_i N_j \,\mathrm{d}\Omega \neq \delta_{ij} \tag{3.10}$$

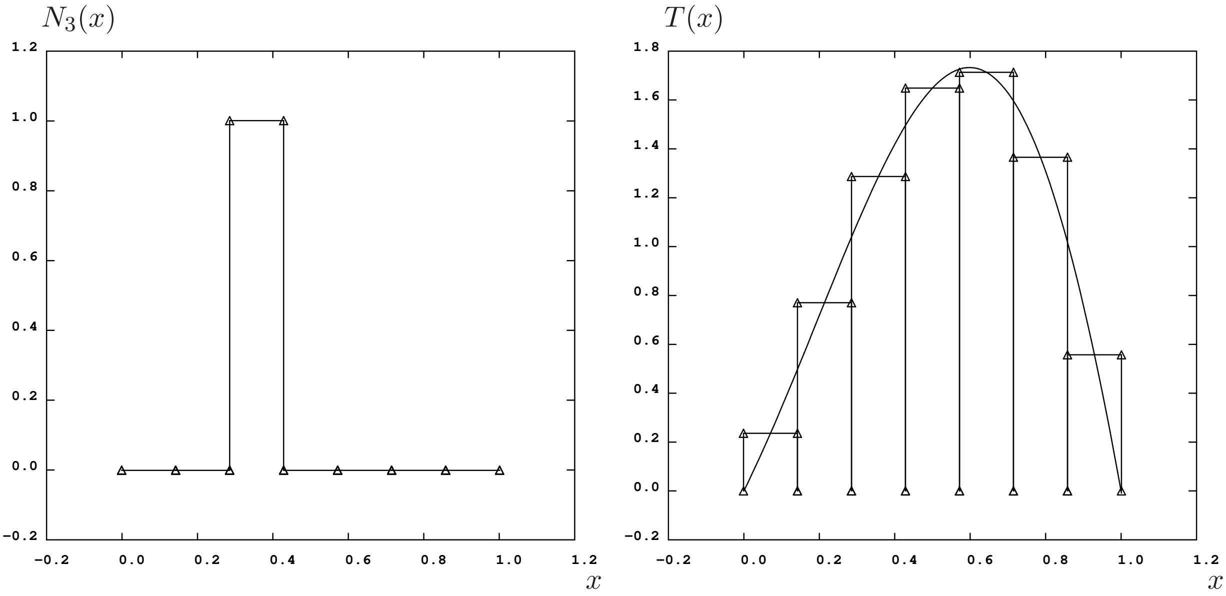


Figure 3.1: Sample functions in V_0 in 1D. Left: the N_3 basis function, indicator function of the third element. Right: an arbitrary function T (—) and its interpolate $T_h \in V_0$ ($\triangle$).

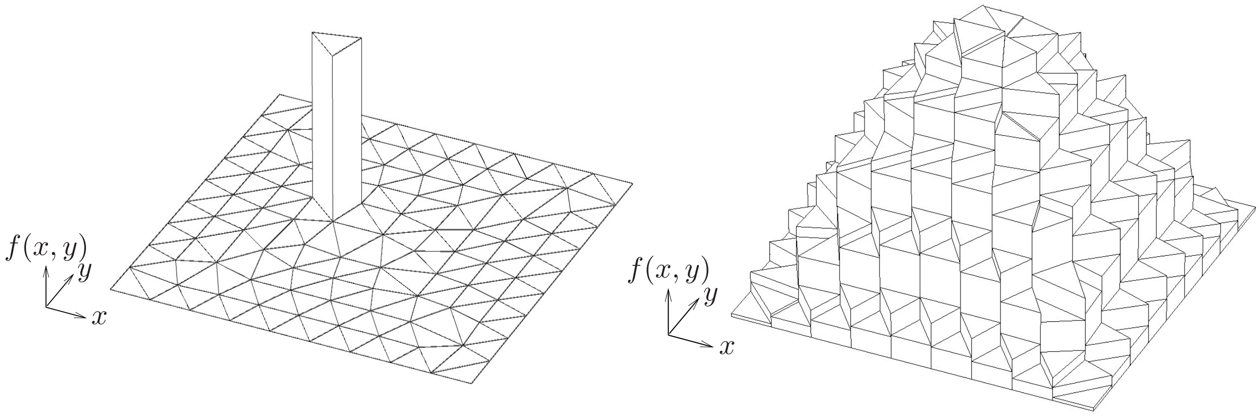


Figure 3.2: Sample functions in V_0 in 2D. Left: a N_i basis function, indicator function if the i^{th} element. Right: a function $T_h \in V_0$.

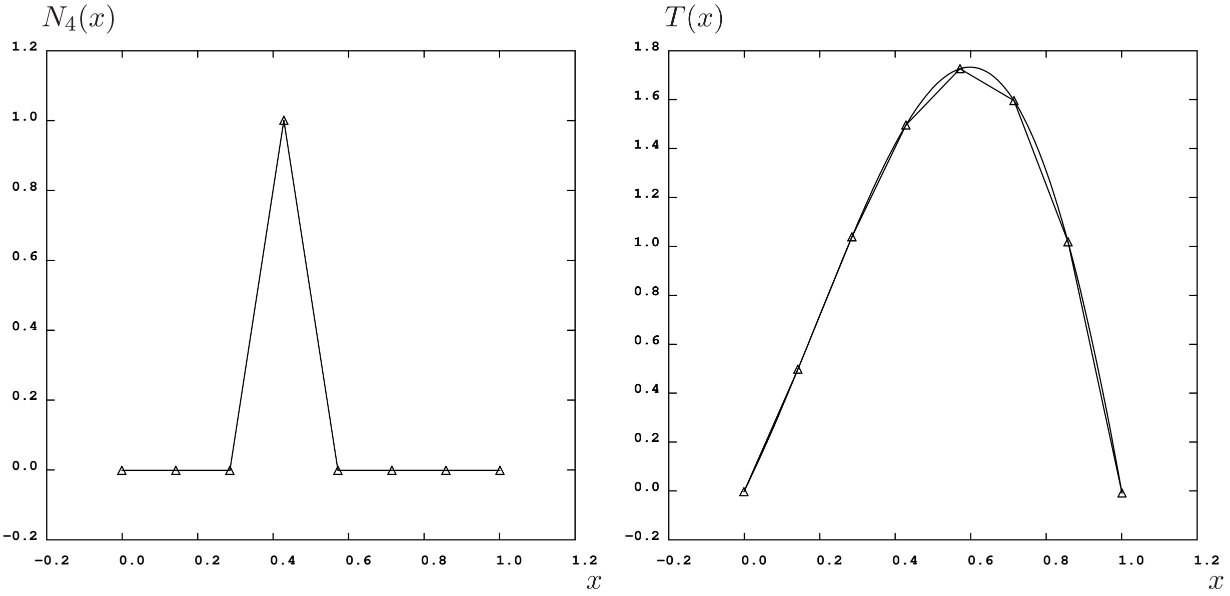


Figure 3.3: Sample functions in W_1 in 1D. Left: the N_4 basis function, hat function built on the 4^{th} vertex of the mesh. Right: an arbitrary function T (—) and its interpolate $T_h \in W_1$ ($\triangle$).

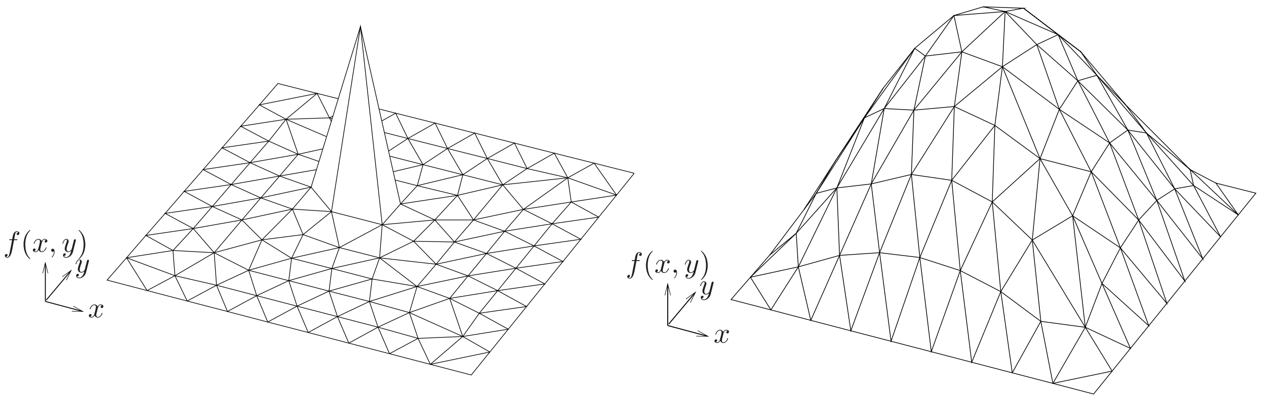


Figure 3.4: Sample functions in W_1 in 2D. Left: a N_j basis function, hat function built on the j^{th} vertex. Right: a function $T_h \in W_1$.

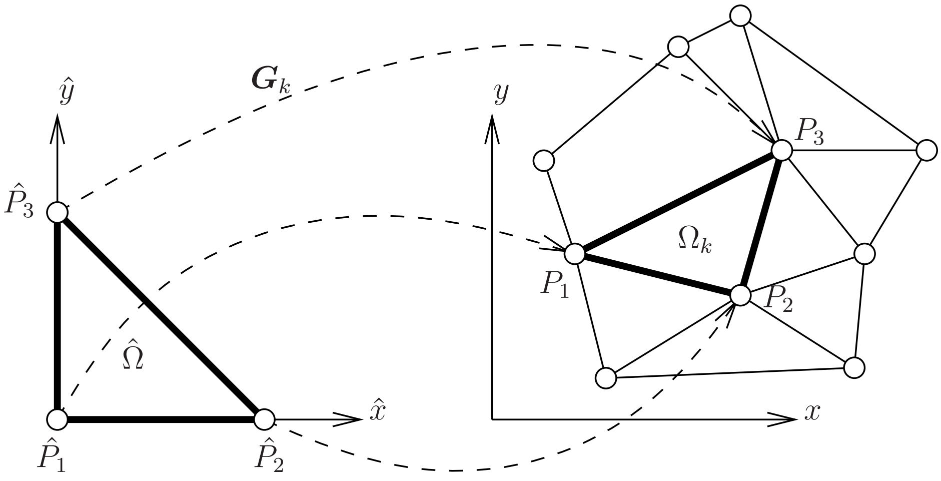


Figure 3.5: The geometric transformation $\boldsymbol{G}_k$ maps the reference element $\hat{\Omega}$ to an actual element Ω_k.

Thus the *mass matrix* M is not a diagonal matrix. This can be a drawback: for instance, even with an explicit finite-difference time discretization scheme, each time step involves a solution of a linear system with the mass matrix in the standard finite element method (see chapter 5). So-called mass matrix diagonalization techniques have been developed in order to avoid this drawback but they are outside of the scope of these lecture notes and must be used with precautions.

3.5 Reference element

As an example of practical calculation of a weighted residual integral, let us show how to compute $\int_\Omega \boldsymbol{u} \cdot \nabla T_h \,\mathrm{d}\Omega$:

$$\int_\Omega \boldsymbol{u} \cdot \nabla N_i N_j \,\mathrm{d}\Omega = \sum_k \int_{\Omega_k} \boldsymbol{u} \cdot \nabla N_i N_j \,\mathrm{d}\Omega_k \tag{3.11}$$

Experience shows that it is, in general, easier to compute the N_i and the integrals not directly on Ω_k, but rather on a reference domain $\hat{\Omega}$ where the coordinate system is orthogonal (cartesian) and where the elements have a simple fixed shape (straight edges). To do this, we use a geometric transformation $\boldsymbol{G}_k$ which maps the vertices of $\hat{\Omega}$ to the vertices of Ω_k (figure 3.5):

$$\boldsymbol{G}_k : \hat{\boldsymbol{x}} \to \boldsymbol{x} \tag{3.12}$$

If we assume that $\boldsymbol{G}_k$ is one to one (at least inside $\hat{\Omega}$), we can associate to any scalar-valued function f defined on an actual element Ω_k, a corresponding hat function defined on the reference element $\hat{\Omega}$:

$$\hat{f}(\hat{\boldsymbol{x}}) = f(\boldsymbol{G}_k(\hat{\boldsymbol{x}})) \tag{3.13}$$

In particular, for the basis functions, it holds:

$$\hat{N}_i(\hat{\boldsymbol{x}}) = N_i(\boldsymbol{G}_k(\hat{\boldsymbol{x}})) \tag{3.14}$$

The geometric transformations $\boldsymbol{G}_k$ can be chosen in a simple way using the basis functions for the reference element:

$$\boldsymbol{G}_k(\hat{\boldsymbol{x}}) = \sum_{i=1}^{N} \boldsymbol{x}_{P_i} \hat{N}_i(\hat{\boldsymbol{x}}) \tag{3.15}$$

where N is the number of vertices of Ω_k and $\boldsymbol{x}_{P_i}$ are the coordinates of Ω_k's vertices P_i. This is due to the fact that, similary to the actual basis functions N_i, the basis functions for the reference element satisfy:

$$\hat{N}_i(\hat{\boldsymbol{x}}_{\hat{P}_j}) = \delta_{ij} \tag{3.16}$$

Mapping the gradient of a function from an actual element to the reference element involves the *jacobian matrix* G of the geometric transformation $\boldsymbol{G}_k$. Indeed:

$$\frac{\partial \hat{f}}{\partial \hat{x}_j} = \sum_i \frac{\partial f}{\partial x_i} \frac{\partial (\boldsymbol{G}_k)_i}{\partial \hat{x}_j} = \frac{\partial f}{\partial x_i} \mathsf{G}_{ij} \tag{3.17}$$

Or, more succintly:

$$\hat{\nabla} \hat{f} = \mathsf{G}^t \nabla f \tag{3.18}$$

The previous formula is a generalization of the classical chain rule for functions of a scalar variable: $(f \circ g)' = f' \circ g \times g'$.

Eventually, the original integral mapped on the reference domain writes:

$$\int_{\Omega_k} \boldsymbol{u} \cdot \nabla N_i N_j \, \mathrm{d}\Omega_k = \int_{\hat{\Omega}} \hat{\boldsymbol{u}} \cdot (\mathsf{G}^{-t} \hat{\nabla} \hat{N}_i) \hat{N}_j \det \mathsf{G} \, \mathrm{d}\hat{\Omega} \tag{3.19}$$

An interesting feature of this mapping process is that we have separated information that can be computed once and for all (the basis functions $\hat{N}_i$ and their derivatives on the reference element) from those that change within each element Ω_k (the jacobian matrix G).

3.6 Quadrature formulae

Once the integrals have been expressed on the reference element, it remains to evaluate them. This can be done analytically in simple cases but, most often, we use quadrature formulae to compute the integrals:

$$\int_{\hat{\Omega}} \hat{f}(\hat{\boldsymbol{x}}) \, \mathrm{d}\hat{\Omega} \approx \sum_{i=1}^{r} w_i \hat{f}(\hat{\boldsymbol{x}}_{\hat{Q}_i}) \tag{3.20}$$

Here, $\hat{Q}_i$ are points of the reference domain where we evaluate function $\hat{f}$ (so-called *integration points*), $\hat{\boldsymbol{x}}_{\hat{Q}_i}$ their coordinates and w_i a weight assigned to each point.

You already know such quadrature formulae for approaching the value of an integral in one space dimension: they are names as the midpoint rule, the trapezoidal rule or Simpson's rule.

The most often used formulae are called *Gaussian quadrature*[2] or Gaussian cubature in space dimension higher than 1. They have the best possible order of precision (they exactly integrate polynomials up to a given order on the considered domain) while keeping the number of integration points at a minimum[3].

[2] These formulae' integration points are then called Gauss points.

[3] Note that finding Gaussian cubature formulae of arbitrary order is an open problem even for simple integration domains.

3.7 Summary

In this chapter we gave a short review on how to discretize a problem using the finite element method. The method itself only uses basic mathematical tools: linear interpolation (or Lagrange polynomials in the higher order case), change of variable formula, integration by parts, chain rule and quadrature rules.

Difficulties arise from the fact that the discrete problem is not necessarily well-posed even if the continuous problem is. Rigorous study of well-posedness of continuous problems is the application domain of *functional analysis*. Regarding discrete problems, functional analysis is also useful together with more algebraic techniques: Fourier mode analysis… These techniques are out of the scope of these lectures. In the following chapters we will mostly show, using practical examples, how some difficulties arise and describe some often used methods that have been developed to circumvent them.

Chapter 4

Convection-diffusion and upwinding

$$\begin{aligned}
\frac{\partial \boldsymbol{u}}{\partial t} + \boxed{(\boldsymbol{\nabla u}) \cdot \boldsymbol{u}} &= -\nabla p^* + \boxed{\nu \boldsymbol{\Delta u}} + \boldsymbol{s_u} \\
\nabla \cdot \boldsymbol{u} &= 0 \\
\frac{\partial T}{\partial t} + \boxed{\boldsymbol{u} \cdot \nabla T} &= \boxed{\alpha \Delta T} + s_T
\end{aligned}$$

In the previous chapter we described how the finite element method could be used to discretize a partial differential equation. We will now focus on the convergence property of the method for the particular case of the scalar convection-diffusion equation. We will show that is necessary to use *upwinding* in the finite element scheme in order to obtain convergence in the pure convection case and to prevent oscillations in the discrete solution in the convection-dominated case.

4.1 Model problem

We go back to our convection-diffusion model problem (3.1). We consider the one-dimensional case on the interval $[0, L]$ with Dirichlet boundary conditions:

$$\left\{\begin{array}{l} u\dfrac{\partial T}{\partial x} - \alpha\dfrac{\partial^2 T}{\partial x^2} = 0 \\ T|_{x=0} = 0 \\ T|_{x=L} = 1 \end{array}\right. \tag{4.1}$$

The exact solution to this problem is the following:

$$T(x) = \frac{1 - \exp\frac{xu}{\alpha}}{1 - \exp\frac{Lu}{\alpha}} \tag{4.2}$$

The denominator of this expression involves the *Péclet* number based on the domain length:

$$Pe_L = \frac{Lu}{2\alpha} \tag{4.3}$$

Whenever the Péclet number is close to zero, the solution is almost a straight line. Whenever the Péclet number is large, the solution is zero on almost all of the interval except in the neighborhood of the right boundary where the slope is very steep in order to satisfy the boundary condition $T|_{x=L} = 1$ (figure 4.1).

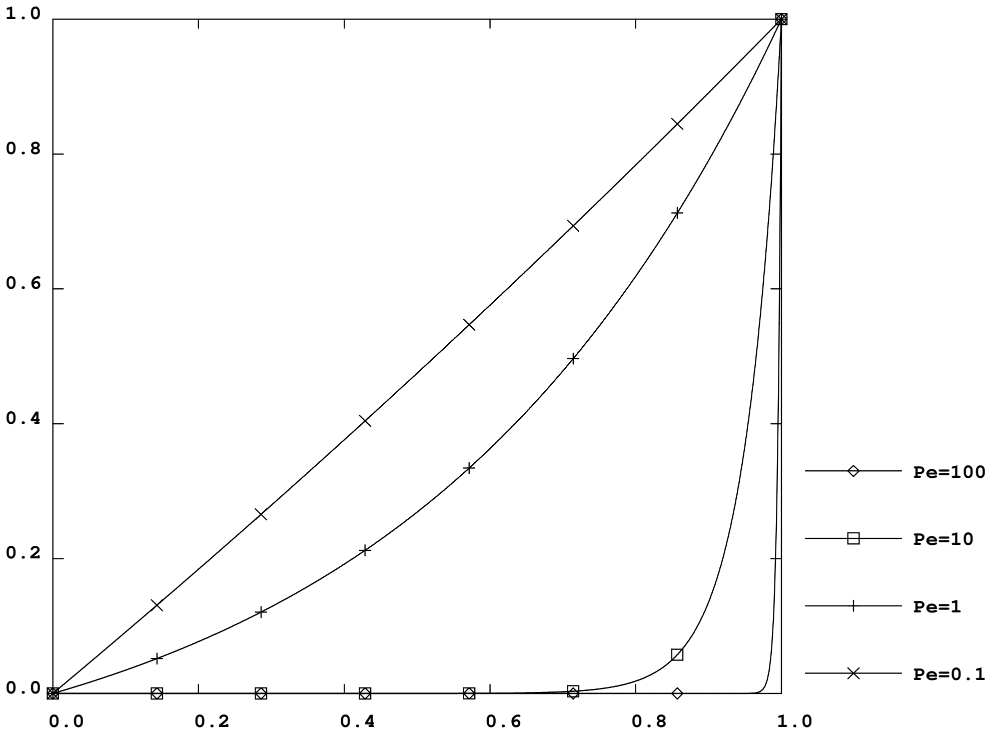


Figure 4.1: Exact solution $T(x)$ of problem (4.1) for various Péclet numbers Pe_L.

4.2 Centered spatial discretization

4.2.1 Equivalence between centered FDM and FEM

Centered Finite Difference Method (FDM)

Let us discretize our model problem (4.1) with a centered finite difference method of order 2. We use a constant spatial discretization step Δx. At a given point i, we have:

$$u\frac{T_{i+1}-T_{i-1}}{2\Delta x}-\alpha\frac{T_{i+1}-2T_i+T_{i-1}}{\Delta x^2}=0 \tag{4.4}$$

We already know that this centered discretization of the problem can lead to *oscillating* discrete solutions (see for instance [DP00]).

Three-point example

For example, let us see what happens when we use a two-element mesh ($\Delta x = L/2$). In this case, we have analytical expressions for the discrete solution:

$$\begin{cases} T_3 = 1 \quad \text{and} \quad T_1 = 0 \\ T_3 - T_1 - \dfrac{2\alpha}{u\Delta x}\left(T_3 - 2T_2 + T_1\right) = 0 \Rightarrow T_2 = \dfrac{1-Pe_{\Delta x}}{2} \end{cases} \tag{4.5}$$

where $Pe_{\Delta x}$ is the local *element-wise Péclet* number, computed with the local mesh size: $Pe_{\Delta x} = \frac{\Delta x u}{2\alpha}$. One can see that, when $Pe_{\Delta x} \to \infty$, then $T_2 \to -\infty$. In particular, whenever $Pe_{\Delta x} > 1$, T_2 becomes negative, out of the bounds given by the Dirichlet boundary conditions, namely $[0,1]$.

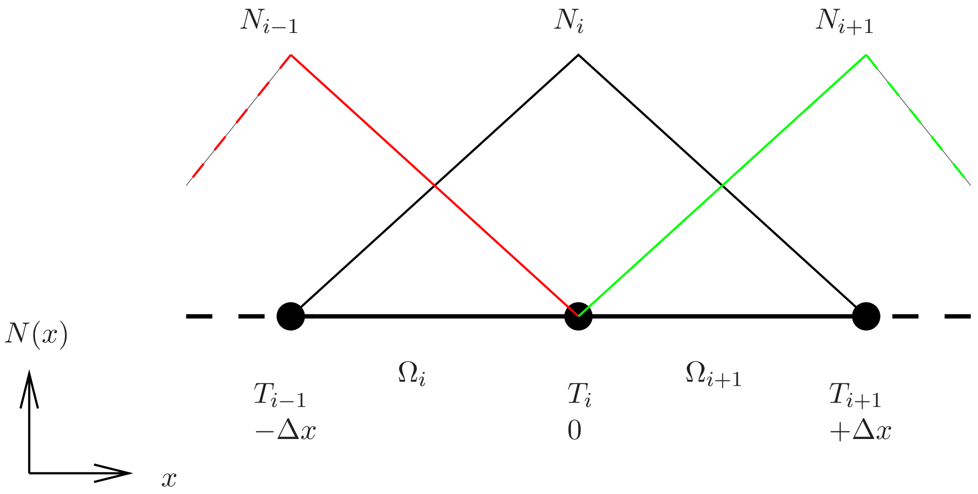


Figure 4.2: Basis functions supported by vertex i

We recognize the following important (sufficient) condition in order to obtain a non-oscillating discrete solution:

$$Pe_{\Delta x} \leq 1 \Rightarrow \Delta x \leq \frac{2\alpha}{u} \tag{4.6}$$

Finite Element Method (FEM)

Unfortunately, if we discretize the model problem (4.1) with the finite element method on a regular mesh, we get a scheme that is identical to the centered finite-difference one.

Let us show that this property holds for the convective term. On the i^{th} vertex, we have to compute integrals such as:

$$\int_\Omega \left(\sum_{j=1}^{n} uT_j \frac{\partial N_j}{\partial x} \right) N_i \, \mathrm{d}\Omega = I_i \tag{4.7}$$

This integral is non-zero for $j = \{i-1, i, i+1\}$ on the elements Ω_i and Ω_{i+1} (figure 4.2):

$$I_i = \left\{ \begin{array}{rcl} \displaystyle\int_{\Omega_i} uT_{i-1} \frac{\partial N_{i-1}}{\partial x} N_i \, \mathrm{d}\Omega_i & = & uT_{i-1} \left(-\frac{1}{\Delta x} \right) \frac{1}{2} \Delta x \\ + \displaystyle\int_{\Omega_i} uT_i \frac{\partial N_i}{\partial x} N_i \, \mathrm{d}\Omega_i & = & uT_i \left(+\frac{1}{\Delta x} \right) \frac{1}{2} \Delta x \\ + \displaystyle\int_{\Omega_{i+1}} uT_i \frac{\partial N_i}{\partial x} N_i \, \mathrm{d}\Omega_{i+1} & = & uT_i \left(-\frac{1}{\Delta x} \right) \frac{1}{2} \Delta x \\ + \displaystyle\int_{\Omega_{i+1}} uT_{i+1} \frac{\partial N_{i+1}}{\partial x} N_i \, \mathrm{d}\Omega_{i+1} & = & uT_{i+1} \left(+\frac{1}{\Delta x} \right) \frac{1}{2} \Delta x \end{array} \right\} = u \frac{T_{i+1} - T_{i-1}}{2} \tag{4.8}$$

This discrete expression for the convective term is identical to the centered finite-difference one (4.4) except for the Δx factor, due to integration.

Exercise 1 *Show that the finite element discretization of the diffusive term on a regular mesh is identical to the centered finite-difference one on a 1D regular mesh.*

Exercise 2 *Do the three-point example of section 4.2.1 with a unique quadratic finite element to discretize interval $[O, L]$ instead of the two equal-length linear finite elements.*

```
Peclet  = 10. ;
typdec = 'CENTREE' ;
typdec = 'SUPG' ;
*
rv = 'EQEX'
     'OPTI' 'EF' 'IMPL' typdec
     'ZONE' $mt 'OPER' 'KONV' 1. 'UN' 'ALF' 'INCO' 'TN'
     'OPTI' 'EF' 'IMPL' 'CENTREE'
     'ZONE' $mt 'OPER' 'LAPN' 'ALF' 'INCO' 'TN'
     'CLIM' gau 'TN' 'TIMP' 0.
     'CLIM' dro 'TN' 'TIMP' 1. ;
*
rv . 'INCO' = 'TABLE' 'INCO' ;
rv . 'INCO' . 'UN'  = 'KCHT' $mt 'VECT' 'SOMMET' (1. 0.) ;
rv . 'INCO' . 'ALF' = 'KCHT' $mt 'SCAL' 'CENTRE' ('/' 0.5 Peclet) ;
rv . 'INCO' . 'TN'  = 'KCHT' $mt 'SCAL' 'SOMMET' 0. ;
*
EXEC rv ;
```

Listing 4.1: CastɜM data file `convdif1d.dgibi` corresponding to problem (4.4).

4.2.2 Numerical results

Let us demonstrate the numerical behavior of the finite element discretization of the model problem. The significant part of the data file, in CastɜM's Gibiane language, corresponding to problem (4.1) is given on listing 4.1. We choose the following parameters:

$$\begin{cases} L = 1 \\ u = 1 \qquad \Rightarrow Pe_L = 10 \\ \alpha = 0.05 \end{cases} \tag{4.9}$$

If we numerically solve the problem with a regular six-element mesh, we get the result presented on figure 4.3. The solid line is the exact solution while the dashed line is the numerical solution. The numerical solution for this case exhibits oscillations. For a six-element mesh, the element-wise Péclet number value is: $Pe_{\Delta x} = Pe_L/6 \approx 1.7 > 1$. The next figure 4.4 shows what happens when we refine the mesh up to 10 (resp. 40) elements such that the element-wise Péclet number $Pe_{\Delta x}$ is 1 (resp. 0.25). The numerical solution gets closer to the exact solution and does not oscillate anymore. In practical cases the condition (4.6) $Pe_{\Delta x} < 1$ on the element-wise Péclet number is very restrictive. For example, for air at 300 K, we have $\alpha = 2.25 \cdot 10^{-5}\, m^2.s^{-1}$. Choosing the characteristic speed and length scales as $u = 1\, m.s^{-1}$ and $L = 1\, m$, we obtain a (global) Péclet number of: $Pe_L \approx 2 \cdot 10^4$. Using a computer, it is generally not practical to have such a number of elements per spatial dimension.

Note however that condition (4.6) is sufficient but not necessary. Figure 4.5 (left) shows that it is possible to get a non-oscillating numerical solution with a six-element mesh that does not satisfy condition (4.6) on every element. To achieve this, we have *adapted* the mesh to the anticipated solution. Mesh adaptation methods are an interesting way of trying to keep computational costs low. Their main drawback is perhaps their complexity because the mesh itself becomes an unknown.

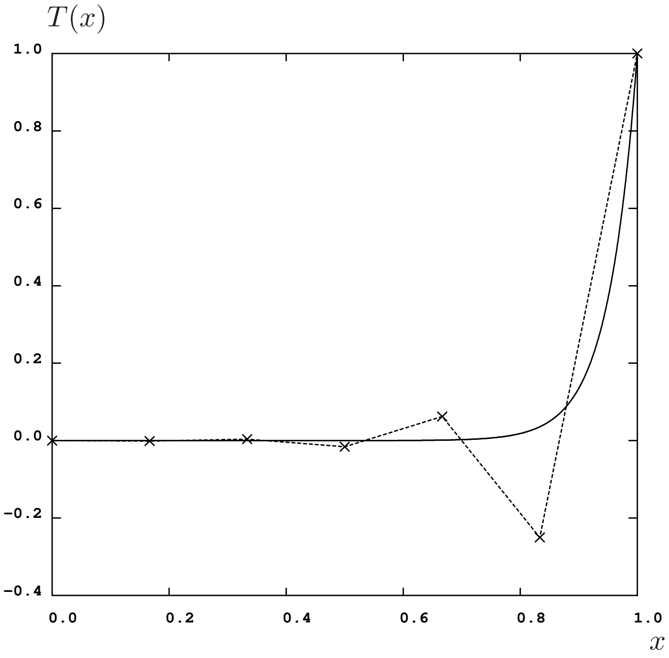


Figure 4.3: Problem (4.1)-(4.9) with $Pe_L = 10$. Regular six-element mesh. —: exact solution $T(x)$. ×: numerical solution $T_h(x)$.

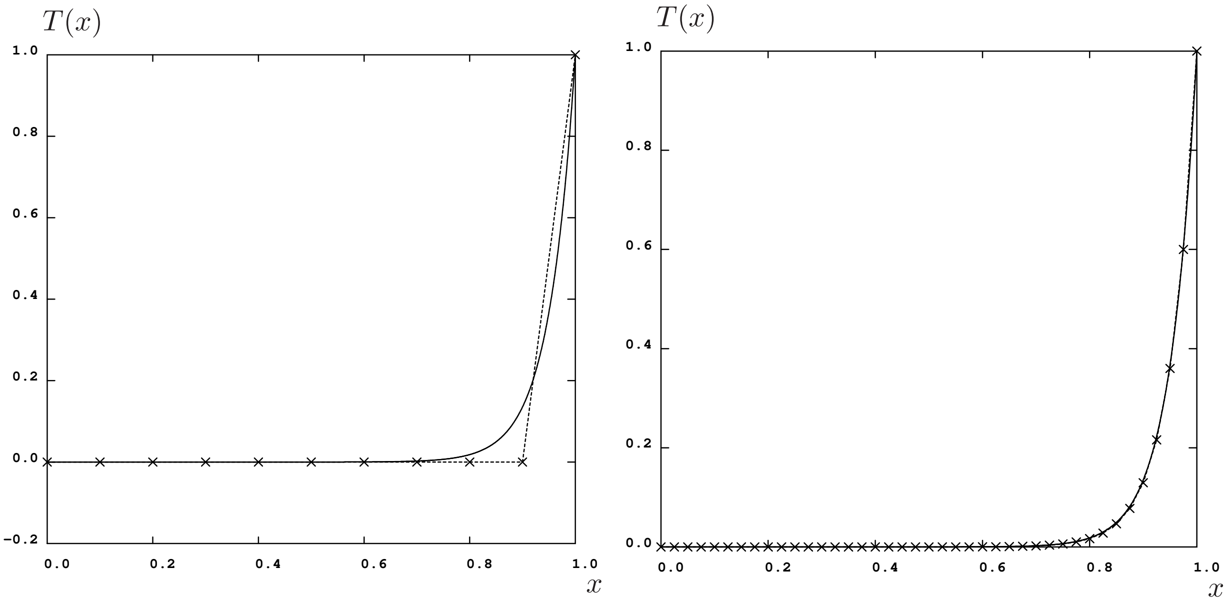


Figure 4.4: Problem (4.1)-(4.9) with $Pe_L = 10$. Left: 10-element regular mesh. Right: 40-element regular mesh. —: exact solution $T(x)$. ×: numerical solution $T_h(x)$.

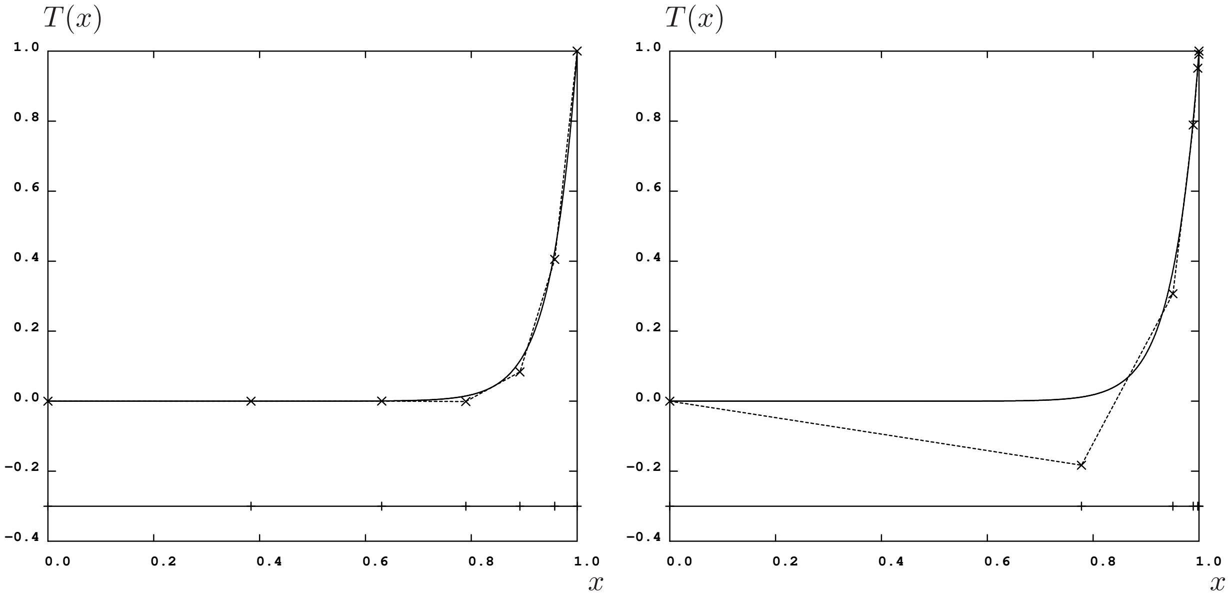


Figure 4.5: Problem (4.1)-(4.9) with $Pe_L = 10$. Six-element mesh refined in the high-variation zone of the solution. Left: correctly refined. Right: "too much" refined. —: exact solution. ×: numerical solution $T_h(x)$. +: mesh vertices.

Mesh adaptation requires some care. Figure 4.5 on the right shows what happens to the numerical solution with a badly adapted mesh. In this case the numerical solution is arguably as bad as in the regular mesh case.

Frequently, one will want to use numerical methods that have less stringent stability conditions than centered discretizations. So-called *upwind* methods achieve this at the expense of having generally a lower order of convergence.

4.3 Upwind spatial discretization

Let us consider another finite difference discretization of our model problem (4.1). We still use a constant spatial discretization step Δx and a second order centered finite difference formula for the diffusive term. However, we discretize the convective term with an *upwind* finite difference formula of order 1. Assuming $u \geq 0$, at a given point i, we have:

$$u\frac{T_i - T_{i-1}}{\Delta x} - \alpha\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} = 0 \tag{4.10}$$

This discretization does not generate oscillations in the solution. If we go back to our two-element mesh example ($\Delta x = L/2$), we get the following analytical expressions for the discrete solution:

$$\left\{\begin{array}{l} T_3 = 1 \quad \text{and} \quad T_1 = 0 \\ T_2 - T_1 - \frac{\alpha}{u\Delta x}\left(T_3 - 2T_2 + T_1\right) = 0 \Rightarrow T_2 = \dfrac{1}{2\left(1 + Pe_{\Delta x}\right)} \end{array}\right. \tag{4.11}$$

This time, when $Pe_{\Delta x} \to \infty$, $T_2 \to 0$ and for all Péclet numbers: $0 < T_2 < 1$ in the bounds given by the Dirichlet boundary conditions.

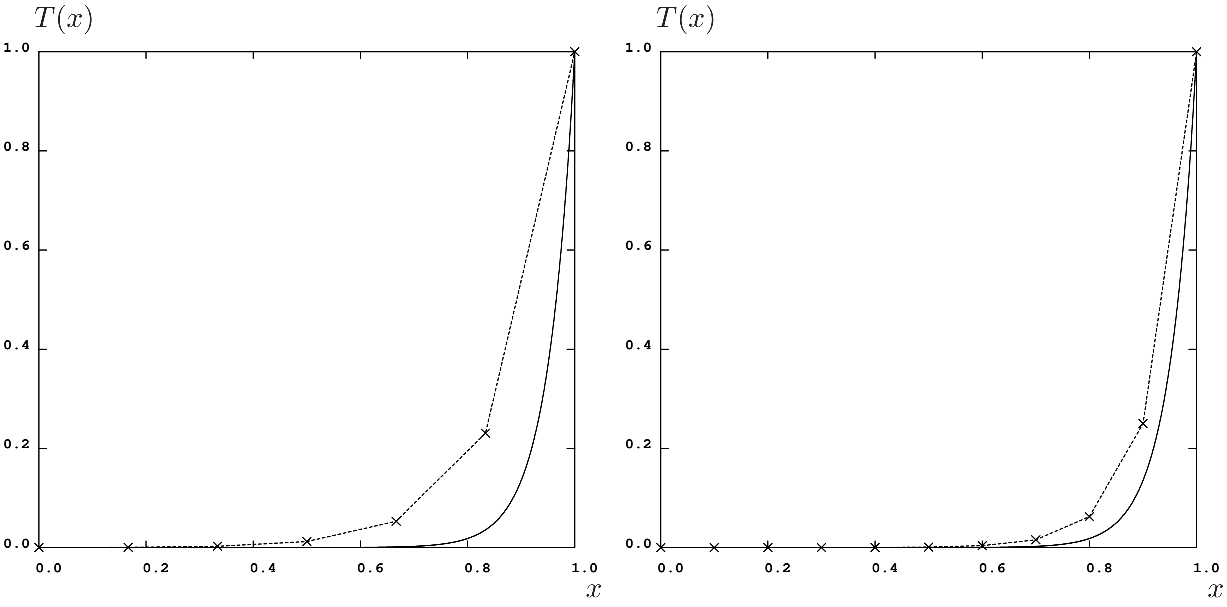


Figure 4.6: Problem (4.1)-(4.9) with $Pe_L = 10$. Finite element method + artifical diffusion $\tilde{\alpha}$ (4.12). Left: regular six-element mesh. Right: regular ten-element mesh. —: exact solution. ×: numerical solution.

If we compute the difference between the centered scheme and the upwind scheme, whatever the sign of u, we get:

$$\begin{aligned}\left[u\frac{\partial T}{\partial x}\right]_{\text{upwind}} - \left[u\frac{\partial T}{\partial x}\right]_{\text{centered}} &= -\frac{\Delta x|u|}{2}\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} \\ &= \left[-\frac{\Delta x\,|u|}{2}\frac{\partial^2 T}{\partial x^2}\right]_{\text{centered}}\end{aligned} \tag{4.12}$$

That is, on a regular mesh, a first-order upwind discretization of the convective term is formally equivalent to a second-order centered discretization of the convective term with an added *numerical diffusion* term with a diffusion coefficient equal to: $\tilde{\alpha} = \frac{\Delta x|u|}{2}$. Note that this diffusion coefficient tends to 0 when $\Delta x \to 0$: both the upwind and the centered discretization are consistent.

This idea is very important and is the foundation for a class of methods called *artificial diffusion* methods. This type of method is suitable for stabilizing the (centered) finite element method. Let us try it on our 1D model problem for a six-element and a ten-element mesh (figure 4.6). Compared to the previous results of figure 4.3 and figure 4.4 (left), we see no oscillations in the numerical solution with artificial diffusion. However, for the ten-element mesh, the latter solution also seems less precise: the gradient near the right boundary is lower than the exact one.

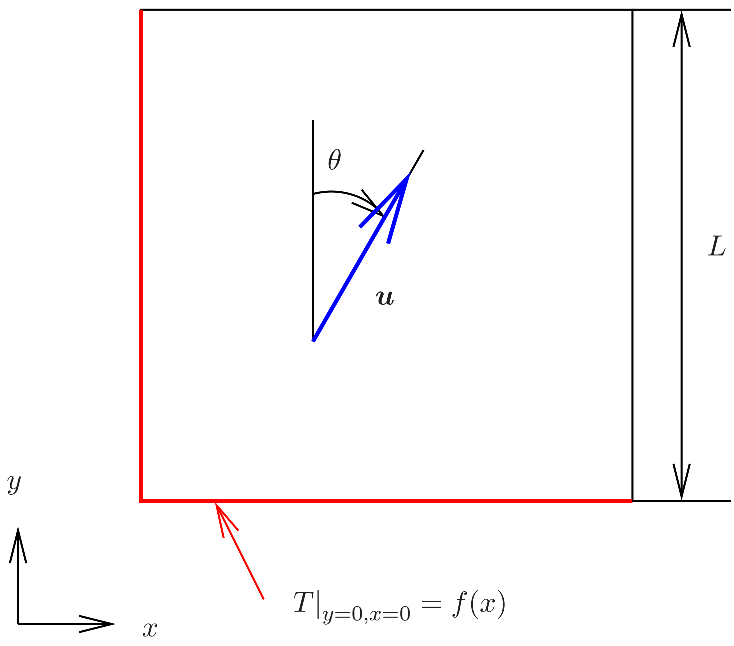


Figure 4.7: 2D convective model problem (4.13).

4.4 Multidimensional extension

4.4.1 2D convective model problem

Here we generalize the artifical diffusion idea to the multidimensional case. We consider the following 2D purely convective transport problem:

$$\begin{cases} \boldsymbol{u} \cdot \nabla T = 0 \quad \text{sur} \quad \Omega = [0, L] \times [0, L] \\ T|_{y=0} = f(x) \\ T|_{x=0} = f(0) \end{cases} \tag{4.13}$$

with $\boldsymbol{u}$ a given constant transport velocity vector. Figure 4.7 illustrates this problem. We choose the following parameters:

$$\begin{cases} L = 1.5 \\ \|\boldsymbol{u}\|^2 = 1 \\ \theta = 10^\circ \\ f(x) = \frac{(\tanh((x-0.5)/-0.15))+1}{2} \end{cases} \tag{4.14}$$

This problem is hyperbolic. One expects that the function given on the boundary with incoming velocity is simply *convected* in the direction given by $\boldsymbol{u}$ on the entire domain. Indeed, $\boldsymbol{u} \cdot \nabla T = 0$ means that the derivative of T is zero (T does not change) in the direction given by $\boldsymbol{u}$.

A sample of the Gibiane data file for problem (4.13) is given on listing 4.2.

We display on figure 4.8 the numerical result obtained with the standard (centered) finite element method on a regular cartesian grid. The numerical solution exhibits very big oscillations.

This is because the discrete problem is not well-posed: the corresponding matrix is not invertible. However, in the computer floating point arithmetics, this matrix is not found

```
angle = 10. ;
typdec = 'CENTREE' ; difart = 1.D-6 ; niter  = 1 ; omeg   = 1. ;
* alfa = U dx / 2 Pem avec Peclet critique de maille = 1
typdec = 'CENTREE' ; difart = 1. '/' (2 '*' nmail) ; niter=1 ; omeg=1. ;
typdec = 'SUPG'    ; difart = 0. ; niter  = 1 ; omeg   = 1. ;
typdec = 'SUPGDC'  ; difart = 0. ; niter = 15 ; omeg  = 0.7 ;
*
rv = 'EQEX' 'NITER' niter 'OMEGA' omeg
     'OPTI' 'EF' 'IMPL' typdec
     'ZONE' $mt 'OPER' 'KONV' 1. 'UN' 'ALF' 'INCO' 'TN'
     'OPTI' 'EF' 'IMPL' 'CENTREE'
     'ZONE' $mt 'OPER' 'LAPN' 'ALF' 'INCO' 'TN'
     'CLIM' mclim 'TN' 'TIMP' cclim
      ;
rv . 'INCO' = 'TABLE' 'INCO' ;
rv . 'INCO' . 'UN'  = 'KCHT' $mt 'VECT' 'SOMMET'
                           (('SIN' angle) ('COS' angle)) ;
rv . 'INCO' . 'ALF' = 'KCHT' $mt 'SCAL' 'CENTRE' difart ;
rv . 'INCO' . 'TN'  = 'KCHT' $mt 'SCAL' 'SOMMET' 0. ;
*
EXEC rv ;
```

Listing 4.2: CastɜM data file `conv2d.dgibi` corresponding to problem (4.13).

exactly singular because of the limited precision for the representation of real numbers. However, as it is nearly singular, we get these very big oscillations in the numerical solution.

4.4.2 Artificial diffusion method

A *first* idea in order to cure these oscillations consists in doing the same thing that we did in one space dimension: add an artificial diffusion term to our equation. We thus discretize:

$$\boldsymbol{u}\cdot\nabla T-\operatorname{div}\frac{h_{\boldsymbol{u}}\,|\boldsymbol{u}|}{2}\nabla T=0 \tag{4.15}$$

where $h_{\boldsymbol{u}}$ is a local mesh size in the direction given by $\boldsymbol{u}$ and $\tilde{\alpha}=\frac{h_{\boldsymbol{u}}|\boldsymbol{u}|}{2}$ is the same numerical diffusion coefficient as in 1D. In doing this, we obtain the result shown on figure 4.9 (left). We observe that there are no oscillations. However, if we compare the profile of the convected function on the inflow and on the outflow boundaries (figure 4.9 on the right), it can be seen that the profile has been notably *diffused* instead of being simply convected. This is clearly not satisfactory.

4.4.3 Streamline upwind diffusion method (SUPG)

A *second* idea consists in adding numerical diffusion, but this time only in the transport direction $\boldsymbol{u}$. In order to achieve this, we have to take a numerical diffusion which is not a scalar $\tilde{\alpha}$, but a second-order tensor denoted by A. The discretized problem writes:

$$\boldsymbol{u}\cdot\nabla T-\operatorname{div}\mathsf{A}\nabla T=\boldsymbol{u}\cdot\nabla T-\operatorname{div}\frac{h_{\boldsymbol{u}}\,|\boldsymbol{u}|}{2}\frac{\boldsymbol{u}\otimes\boldsymbol{u}}{|\boldsymbol{u}|^2}\nabla T=0 \tag{4.16}$$

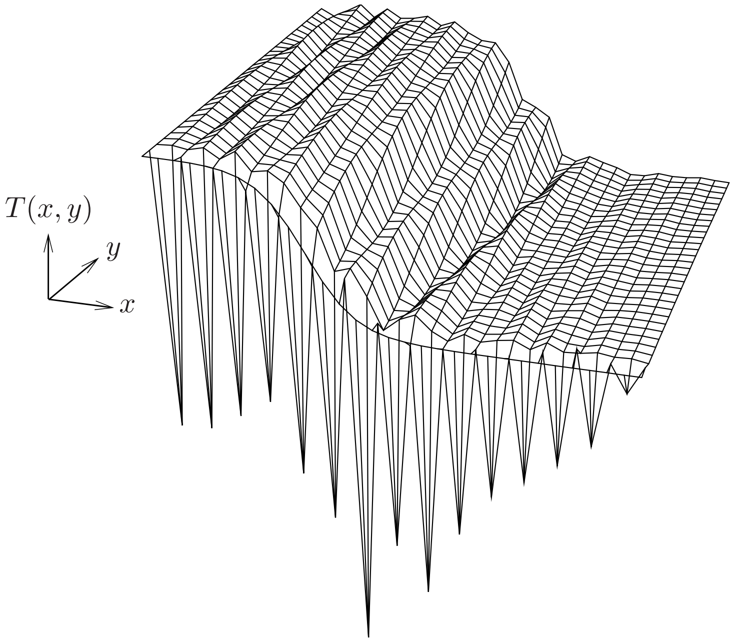


Figure 4.8: Problem (4.13)–(4.14). Standard finite element method (centered=`CENTREE`).

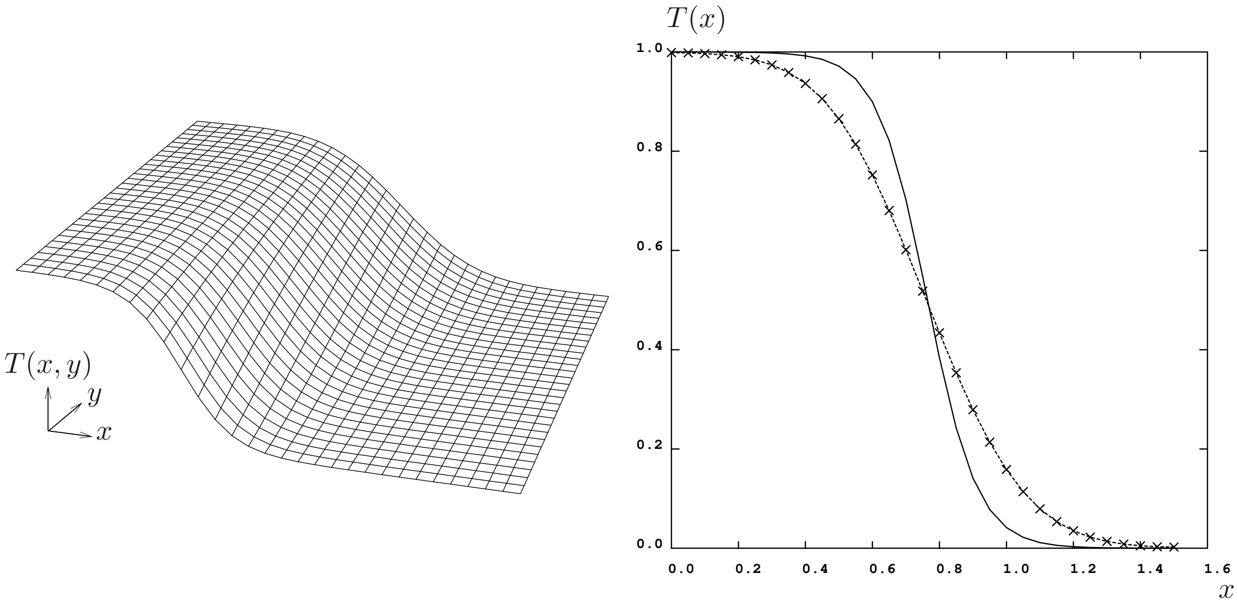


Figure 4.9: Problem (4.13)–(4.14). Finite element method + artificial diffusion (4.15). Left: $T(x,y)$. Right: inflow ($y = 0$) profile of T (—) and outflow ($y = 1$) profile $y = 1(\times)$.

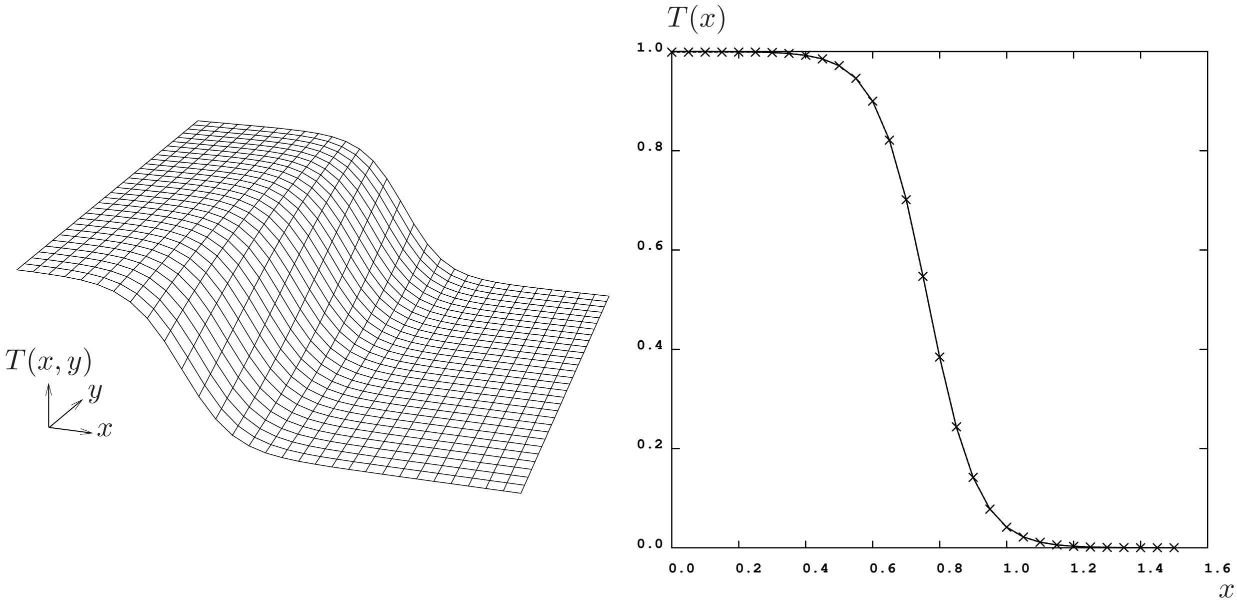


Figure 4.10: Problem (4.13)–(4.14). Finite element method + streamline upwinding term `SUPG` (4.17). Left: $T(x, y)$. Right: inflow ($y = 0$) profile of T (—) and outflow ($y = 1$) profile $y = 1(\times)$.

Why will the diffusion occur only in the direction of $\boldsymbol{u}$? We can explain this by looking at the eigencomponents of the $\frac{\boldsymbol{u}\otimes\boldsymbol{u}}{|\boldsymbol{u}|^2}$ tensor. If we write its components in a 2D cartesian basis, we get:

$$\frac{\boldsymbol{u}\otimes\boldsymbol{u}}{|\boldsymbol{u}|^2} = \frac{1}{|\boldsymbol{u}|^2}\begin{pmatrix} u_x^2 & u_x u_y \\ u_y u_x & u_y^2 \end{pmatrix} \tag{4.17}$$

It is easy to check that this tensor has only one non-zero eigenvalue. This eigenvalue is equal to 1 and the corresponding eigenvector is $\boldsymbol{u}$. Thus, if we compute the product $\frac{\boldsymbol{u}\otimes\boldsymbol{u}}{|\boldsymbol{u}|^2}\nabla T$, the result contains only the part of ∇T that is parallel to $\boldsymbol{u}$.[1]

This is the correct generalization of the artificial diffusion method in space dimension greater than 1. It is often called the Streamline Upwind (`SU`) method by analogy with the 1D case. This type of method has been popularized, in particular by Hughes et al. [Hug87], in the beginning of the 1980's under the name `SUPG` (Streamline Upwind Petrov Galerkin; this is the name used in Cast3M). `SUPG` can be seen as a variant of the `SU` method. Since then, an impressive amount of research has been carried out and published on this subject.

Figure 4.10 illustrates the numerical results obtained with the streamline upwind method. On the left, we can see that no oscillations remain. On the right, we can see that the inflow and outflow profile are nearly identical.

4.4.4 Asymptotic preservation of the consistency order

The purely convective problem (4.13) is a particular case of the convection-diffusion problem:

$$\boldsymbol{u}\cdot\nabla T - \operatorname{div}\alpha\nabla T = 0 \tag{4.18}$$

[1] That is we perform the projection of ∇T on the vector subspace generated by $\boldsymbol{u}$.

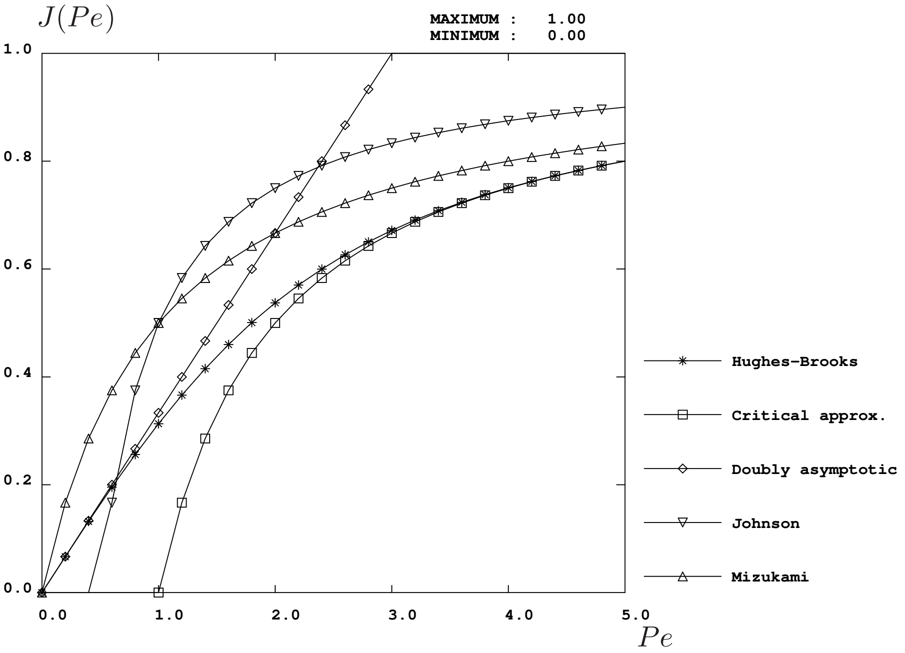


Figure 4.11: Weighting functions $J(Pe)$ of the numerical diffusion term in (4.19) proposed by various authors.

For the latter problem we can propose an enhancement of the SUPG method: we multiply the numerical diffusion tensor A by a weighting function J that depends on the local mesh Péclet number $Pe_{h_{\boldsymbol{u}}} = \frac{h_{\boldsymbol{u}} u}{2\alpha}$. We have already seen that upwinding allows a gain in *stability* at the expense of a loss in *precision*: the upwind discretization of the convective term is first-order in space whereas the centered discretization is second-order. The main idea behind the present enhancement is to use the centered method wherever the stability condition $Pe_{h_{\boldsymbol{u}}} < 1$ holds and the upwind method otherwise. The discretized problem becomes:

$$\begin{aligned}\boldsymbol{u} \cdot \nabla T - \operatorname{div} \mathsf{B}(Pe_{h_{\boldsymbol{u}}}) \nabla T - \operatorname{div} \alpha \nabla T = \boldsymbol{u} \cdot \nabla T - \operatorname{div} \frac{h_{\boldsymbol{u}} |\boldsymbol{u}|}{2} J(Pe_{h_{\boldsymbol{u}}}) \frac{\boldsymbol{u} \otimes \boldsymbol{u}}{|\boldsymbol{u}|^2} \nabla T \\ - \operatorname{div} \alpha \nabla T = 0\end{aligned} \tag{4.19}$$

where J should satisfy: $J(Pe) \to 1$ when $Pe > 1$ and $J(Pe) \to 0$ when $Pe \to 0$. Figure 4.11 displays several choices of J that have been proposed in the literature.

4.5 Remaining oscillations

4.5.1 The Gibbs phenomenon

At this point we can ask to ourselves: is the SUPG method the last word in the finite element discretization of convection-diffusion problem? This is not quite the case; figure 4.12 shows that remaining oscillations are still present in some cases. The problem solved here is still the purely convective one (4.13) but we have changed the smooth tanh profile

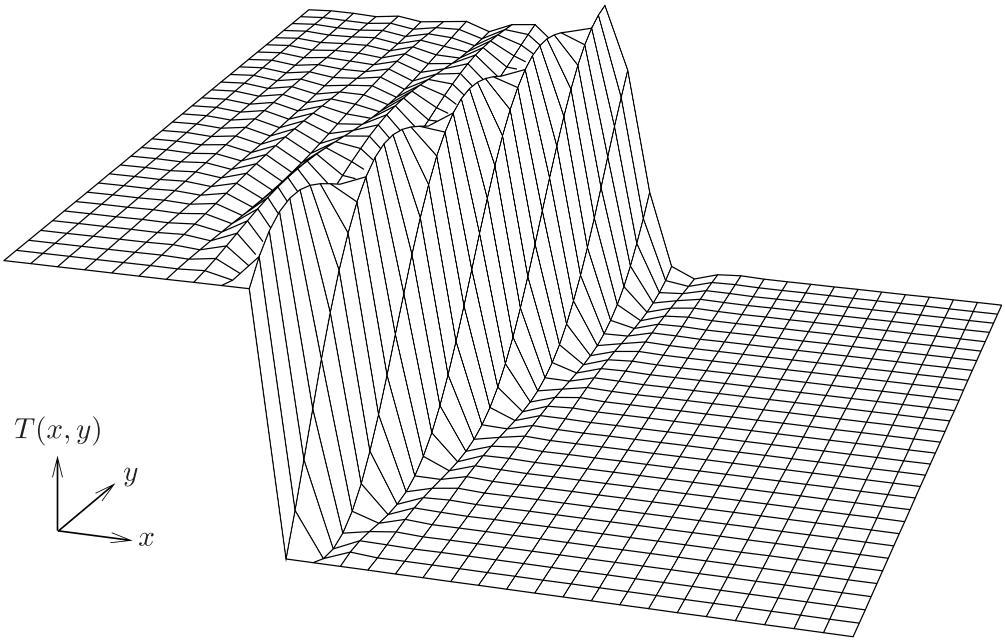


Figure 4.12: Problem (4.13)–(4.20). Discontinuous step-like boundary condition. Finite element method + streamline upwinding term `SUPG` (4.16).

on the incoming velocity boundary for a less regular step-like profile:

$$\begin{cases} L = 1.5 \\ \|\boldsymbol{u}\|^2 = 1 \\ \theta = 10^\circ \\ f(x) = 1 \quad \text{if} \quad x < 0.5 \\ f(x) = 0 \quad \text{if} \quad x \geq 0.5 \end{cases} \tag{4.20}$$

In fact, these remaining oscillations are not a deficiency unique to the `SUPG` method: we will see them appear on several other occasions (section 4.5.2 and 5.2). These oscillations are the consequence of an approximation problem: we are trying to approximate a discontinuous solution with continuous functions. Frequently the solution of such an approximation is oscillatory: this is called *Gibbs'* phenomenon. You may already know Gibbs' phenomenon in the context of Fourier approximation: if one tries to approximate a step-like (Heaviside) function with a finite Fourier series, the latter oscillates near the discontinuity. Furthermore, these oscillations do not vanish as the series' order is increased.

Then shall we choose to give up on continuity in order to get better solutions? In the standard (conforming) finite element method, we cannot give up on continuity because we seek for the solution in a subspace of $\mathcal{H}^1(\Omega)$ in order to be able to discretize Laplacian terms[2].

Other discretization methods choose to give up on continuity: this is the case in *Finite Volume* methods where the solution is taken to be element-wise constant (or, better said, a degree of freedom is defined as the mean value of the solution on a control volume). This is also the case in *Discontinuous Galerkin* Finite Element Methods (DGFEM) where the solution is polynomial in an element but discontinuous at the boundary between elements.

[2] Problem (4.13) does not have any Laplacian term but its approximation stabilized by the `SUPG` method (4.16) does!

However in these methods, whenever the unknowns gradient is needed at a boundary (if there are diffusive terms to discretize for example), the definition of this gradient is not that natural because of the discontinuity.

4.5.2 Shock approximation

We shall now look more closely at the problem of approximating a shock-like function with a discontinuity. In section 3.3, we noted that Lax's theorem left several choices open: the choice of the approximation functional spaces as well as the choice of norms. In this subsection, we will keep a continuous functional space for the approximation space and investigate various possibilities for the norm.

Best approximation of a given function

First, let us write the general problem of finding the best approximation to a given function g in the $\mathcal{L}^2$-norm:

$$\min_{T\in\mathcal{L}^2(\Omega)} A(T) = \min_{T\in\mathcal{L}^2(\Omega)} \int_\Omega \frac{1}{2}\|T-g\|^2 \,\mathrm{d}\Omega \tag{4.21}$$

For the functional A to be computable, it makes sense to require that both T and g be in $\mathcal{L}^2$. Using the method of chapter 2, we can write the minimization condition:

$$\delta_U A(T) = \int_\Omega TU - gU \,\mathrm{d}\Omega = 0 \quad \forall U \in \mathcal{L}^2(\Omega) \tag{4.22}$$

Obviously, the solution to this problem is $T = g$! However, if we choose T in a continuous subspace of $\mathcal{L}^2$ and if g is discontinuous, the solution is not $T = g$ because T and g belong to different functional spaces.

Choosing T in a discrete space of continuous functions and discretizing problem (4.22) leads to:

$$\begin{aligned}\int_\Omega T_h N_i - gN_i \,\mathrm{d}\Omega &= \sum_j T_j \Big(\int_\Omega N_j N_i \,\mathrm{d}\Omega\Big) - \int_\Omega gN_i \,\mathrm{d}\Omega \\ &= \sum_j T_j \mathsf{M}_{ji} - g_i = 0 \quad \forall i \in \Omega\end{aligned} \tag{4.23}$$

The problem to be solved involves the mass matrix M_{ji}. T_h is the best approximation of g in the $\mathcal{L}^2$-norm sense.

Best approximation of a shock by a continuous function

Let us now focus on the approximation problem for a specific function $g(x)$:

$$\begin{cases} g(x) = 1 & \text{if} \quad x > 1 \\ g(x) = 0 & \text{if} \quad x < 1 \end{cases} \tag{4.24}$$

For the sake of simplicity, we restrict ourselves to the $\Omega = [0, 1]$ interval and choose the simplest continuous functional space we can think of. We thus consider the one-parameter

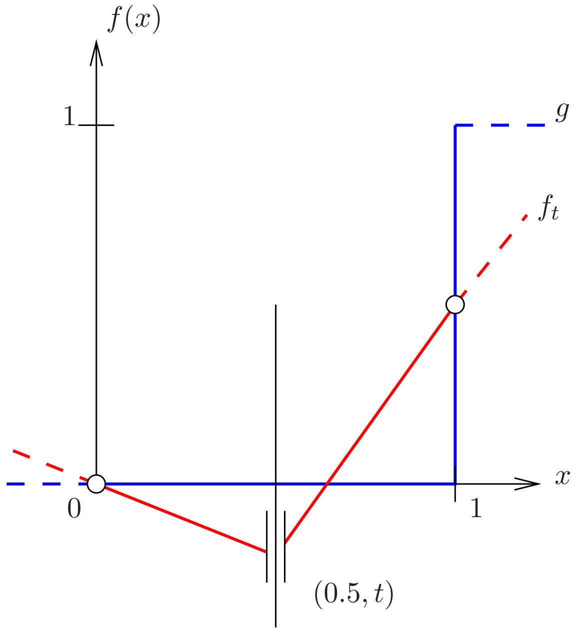


Figure 4.13: Best approximation of a shock function g by a continuous one-parameter t functionf_t.

t family of functions $f_t(x)$, which are piecewise linear and go through the following points: $(0,0)$, $(0.5,t)$ and $(1,0.5)$[3]:

$$\begin{cases} f_t(x) = 2tx \quad \text{if} \quad x \in [0, 0.5] \\ f_t(x) = (1-2t)x + (2t-0.5) \quad \text{if} \quad x \in [0.5, 1] \end{cases} \tag{4.25}$$

Figure 4.13 plots f_t and g. First, let us solve the best approximation problem in the $\mathcal{L}^2$-norm:

$$\min_{t\in\mathbb{R}} I(t) = \min_{t\in\mathbb{R}} \int_\Omega (f_t - g)^2 \, \mathrm{d}\Omega \tag{4.26}$$

We can solve the problem graphically by plotting $I(t)$, see figure 4.14. The minimum of I is obtained for $t = -0.25$, that is for a function $f_{-0.25}$ which *oscillates*. Next, we solve the best approximation problem in the $\mathcal{L}^1(\Omega)$-norm:

$$\min_{t\in\mathbb{R}} J(t) = \min_{t\in\mathbb{R}} \int_\Omega |f_t - g| \, \mathrm{d}\Omega \tag{4.27}$$

$J(t)$ is also plotted on figure 4.14. The minimum is J obtained for $t = 0$, that is for a function f_0 which is *monotonous*. These results hold if we discretize the best approximation problem with more than two elements (figure 4.15).

An important fact is that functional $J(t)$ is minimal at a point at which it is *not derivable*. Thus, we can not apply the techniques of chapter 2 to solve $\mathcal{L}^1$ minimization problems because these techniques rely on the functional derivative. Solving $\mathcal{L}^1$ minimization problems directly requires tools from *non-differentiable optimization*. The $\mathcal{L}^1$-norm is not linked to a simple scalar product and we can not formulate the equivalent of a weighted residual method leading to a linear system: the best approximation problem in the $\mathcal{L}^1$-norm is *non-linear*.

[3] This last point is chosen for symmetry reasons.

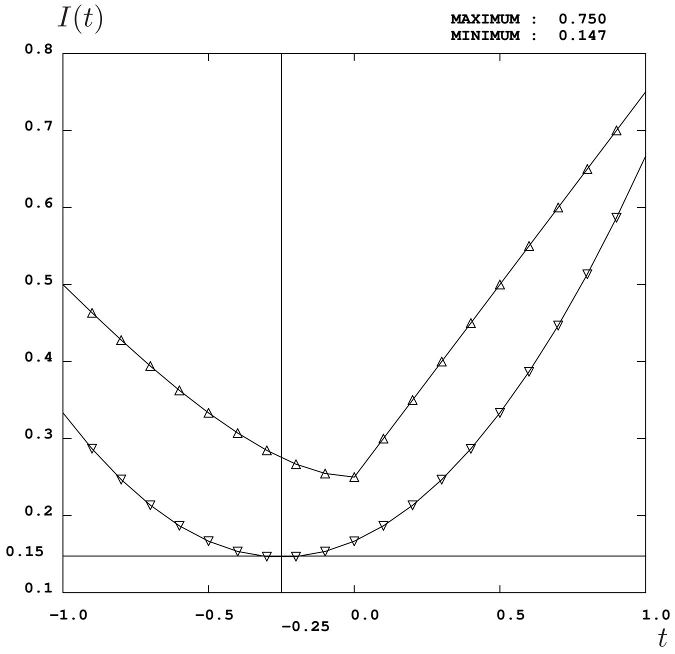


Figure 4.14: Plot of the functionals $I(t) = \int_\Omega (f_t - g)^2 \, \mathrm{d}\Omega$ ($\mathcal{L}^2$-norm: ∇) and $J(t) = \int_\Omega |f_t - g| \, \mathrm{d}\Omega$ ($\mathcal{L}^1(\Omega)$-norm : $\triangle$).

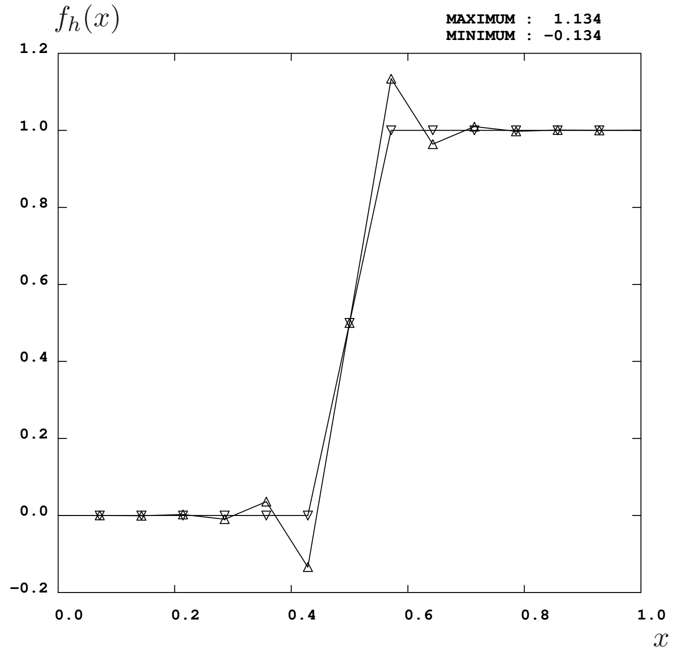


Figure 4.15: Best approximation of a shock function by a discrete, continuous, piecewise linear function in the sense of the $\mathcal{L}^1$ (∇) and $\mathcal{L}^2$ ($\triangle$) norms.

Nevertheless we can go back to a differentiable optimization problem if we accept to *regularize* the $\mathcal{L}^1$-norm with a suitably chosen small parameter ϵ:

$$\min_{t\in\mathbb{R}} K(t) = \min_{t\in\mathbb{R}} \int_\Omega \sqrt{(f_t - g)^2 + \epsilon}\, \mathrm{d}\Omega \tag{4.28}$$

Indeed, we can now use standard techniques like functional derivation to solve this problem. Note however that this is still a non-linear problem.

Exercise 3 *Solve graphically the best approximation problem of g with f_t in the regularized $\mathcal{L}^1_\epsilon$-norm and look at the influence of the ϵ parameter.*

4.5.3 Godunov's theorem

In fact the results obtained on the elementary exemple of the previous subsection are quite general. We have the following theorem, due to Godunov:

Theorem 2 (Godunov) *Numerical schemes for solving partial differential equations having the property of monotonicity (in the sense of not generating new extrema) are of order 1 at most.*

In our elementary exemple we use piecewise linear functions, able to approximate smooth solutions up to order 2. Solutions found with a linear scheme (using the $\mathcal{L}^2$ norm) are not monotonous.

We wanted to bring forward the question of the norm in best approximation problems because, in some cases, we seek for solutions that must remain between prescribed bounds. For instance, this is the case in the field of image processing. In these cases the use of the $\mathcal{L}^1$ norm is mandatory.

In the field of fluid mechanics these methods are useful in some problems, for example, the approximation and transport of interfaces (*level-set* methods for capturing an interface between two immiscible liquids…). In so-called *shock-capturing* methods a property related to monotonicity is frequently required: the *TVD* (Total Variation Diminishing) property. The total variation of a differentiable function f is defined as the $\mathcal{L}^1$-norm of its gradient:

$$TV(f) = \int_\Omega |\nabla f|\, \mathrm{d}\Omega \tag{4.29}$$

We already met this concept of total variation in section 2.1.1, where it was shown that it allowed to select monotonous functions (in 1D) going from one point to another.

In these lecture notes, we will stay within the $\mathcal{L}^2(\Omega)$-norm framework for practical reasons: even though we might obtain oscillatory solutions, they are simple to compute (linear system solution) and they satisfy boundary conditions and conservation properties in a quite natural way. We will deal with these two latter properties extensively in chapter 9.

For more on optimization principles in the $\mathcal{L}^1$-norm, the interested reader can refer to Strang [Str07].

4.6 Shock capturing: the SUPGDC method

Many authors have proposed enhancements of the SUPG method for the case where the solution exhibits strong variations, as in figure 4.12. For instance, Hughes et al. [Hug87] suggest to add a so-called Discontinuity Capturing (DC) term to the SUPG numerical scheme. Once again, this is a numerical diffusion term. But, this time, it is designed to act only in the direction of the solution's gradient. We define $\boldsymbol{u}_{\|}$ as the projection of the transport velocity onto the solution's gradient:

$$\boldsymbol{u}_{\|}(T) = \left(\frac{\boldsymbol{u} \cdot \nabla T}{|\nabla T|^2} \right) \nabla T \tag{4.30}$$

The discretized problem in the SUPGDC method writes:

$$\begin{aligned} \boldsymbol{u} \cdot \nabla T - \operatorname{div} \mathsf{B} \nabla T - \operatorname{div} \mathsf{C} \nabla T - \operatorname{div} \alpha \nabla T = \boldsymbol{u} \cdot \nabla T & \\ - \operatorname{div} \frac{h_{\boldsymbol{u}} |\boldsymbol{u}|}{2} J(Pe_{h_{\boldsymbol{u}}}) \frac{\boldsymbol{u} \otimes \boldsymbol{u}}{|\boldsymbol{u}|^2} \nabla T & \\ - \operatorname{div} \frac{h_{\boldsymbol{u}_{\|}} \left|\boldsymbol{u}_{\|}\right|}{2} J(Pe_{h_{\boldsymbol{u}_{\|}}}) \frac{\boldsymbol{u}_{\|} \otimes \boldsymbol{u}_{\|}}{\left|\boldsymbol{u}_{\|}\right|^2} \nabla T & \\ - \operatorname{div} \alpha \nabla T = 0 & \end{aligned} \tag{4.31}$$

The solution of our model problem of purely convective transport (4.13) with the SUPGDC method is shown on figure 4.16. Almost no oscillations remain. However, the shock is notably diffused (on six elements) at the domain outflow boundary.

It is also important to notice that the discontinuity capturing term is a *non-linear* function of the unknown variable T. Indeed, the added numerical diffusion tensor coefficient depends on T via $\boldsymbol{u}_{\|}(T)$[4]. This is compatible with Godunov's theorem: we are trying to devise a monotonous method of formal order 2. Thus it necessitates a non-linear method.

4.7 Summary

The following table summarizes the properties of the various discretization methods for the linear convection term $\boldsymbol{u} \cdot \nabla T$ that we have discussed in this chapter.

Method	Formal convergence order	Numerical oscillations	Linearity
CENTREE	2	possible if $Pe_h > 1$	linear
SUPG	2 if $Pe_h < 1$ 1 if $Pe_h > 1$	possible if solution has shocks	linear
SUPGDC	2 if $Pe_h < 1$ 1 if $Pe_h > 1$	few	non-linear

Up to this date, the default method used in the CastƐM code is SUPGDC. However, we advise the user to try instead the methods in the increasing order of complexity (which is also the decreasing order of precision): CENTREE, SUPG and SUPGDC.

Whenever possible, the user should compute an order of magnitude of the local mesh Peclet number Pe_h and keep in mind that the condition $Pe_h < 1$ is not a necessary one: it is perfectly possible to obtain an oscillation-free solution even if $Pe_h > 1$.

[4] The numerical treatment of this non-linearity requires the use of the methods of chapter 6.

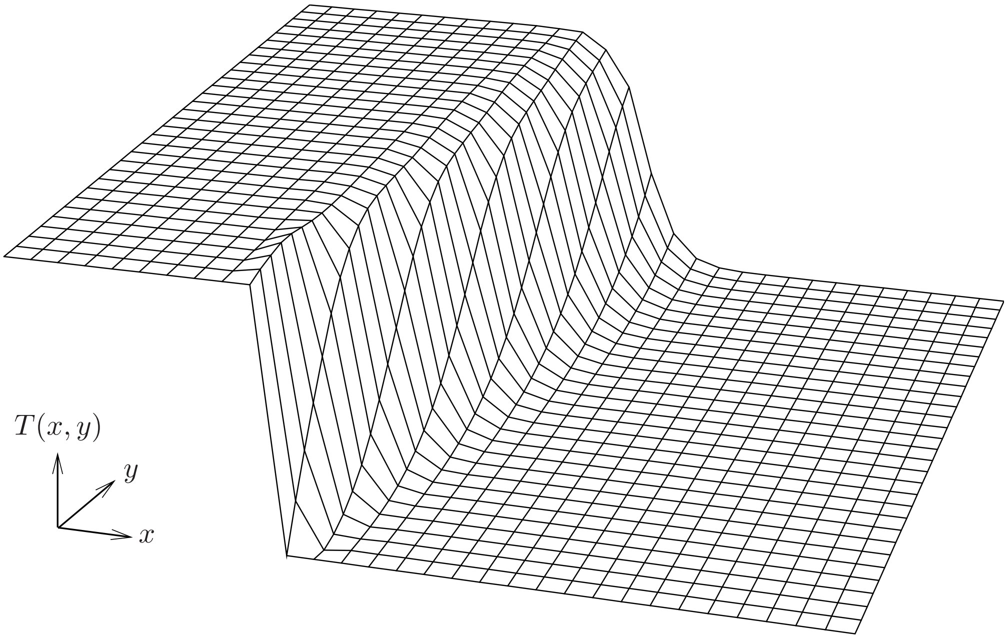


Figure 4.16: Problem (4.13)–(4.20). Discontinuous step-like boundary condition. Finite element method + streamline upwinding and discontinuity capturing terms SUPGDC (4.31).

Chapter 5

Time discretization

$$\begin{aligned}\boxed{\frac{\partial \boldsymbol{u}}{\partial t}} + (\boldsymbol{\nabla u}) \cdot \boldsymbol{u} &= -\nabla p^* + \boxed{\nu \boldsymbol{\Delta u}} + \boldsymbol{s_u} \\ \nabla \cdot \boldsymbol{u} &= 0 \\ \boxed{\frac{\partial T}{\partial t}} + \boldsymbol{u} \cdot \nabla T &= \boxed{\alpha \Delta T} + s_T\end{aligned}$$

> Prediction is very difficult, especially about the future.
>
> Niels Bohr

Up to now, we have only considered stationary problems, without a time derivative $\frac{\partial}{\partial t}$. In this chapter we discuss what happens when we add and discretize this time derivative term. We only consider implicit time discretizations (section 5.1). The model problem we focus on is the unsteady scalar diffusion equation, or heat equation.

Having a time derivative means that we are dealing with an *evolution problem.* In these problems, we have to specify an *initial* condition. We discuss the choice of an initial condition in section 5.2.

5.1 Time discretization

5.1.1 Choosing a time discretization scheme

Explicit time discretization schemes (see [DP00] for an introduction) are generally simpler to implement in a computer code because one doesn't have to solve a linear system at each time step. However, in general, the time step magnitude cannot be chosen arbitrarily large and must satisfy some stability criterion.

In the context of incompressible fluid mechanics, one frequently chooses an implicit time discretization scheme for the Navier-Stokes equations. Indeed:

- The mass conservation equation $\nabla \cdot \boldsymbol{u} = 0$ does not have a time derivative term $\frac{\partial}{\partial t}$. It can be considered as a constraint (see chapter 8) that must be satisfied at every time step. As such, it has an implicit character[1].

- When the diffusive terms in the equations are dominant, stability criteria for explicit methods determine a time step value that can be very small.
- In the finite element method, even with an explicit time discretization scheme, due to the non-orthogonality of the basis functions, one must solve a linear system with the mass matrix at each time step[2].

5.1.2 Implicit time discretization

For the sake of simplicity and robustness, we will consider implicit finite difference time discretization schemes: either the backward difference formula of order 1 (BDF1 or Implicit Euler) or the backward difference formula of order 2 (BDF2):

$$\left\{\begin{aligned} &\frac{\partial T}{\partial t} - S(T) \approx \frac{{T_h}^{k+1} - {T_h}^{k}}{\Delta t} - S({T_h}^{k+1}) \quad \text{Implicit Euler} \\ &\frac{\partial T}{\partial t} - S(T) \approx \frac{3{T_h}^{k+1} - 4{T_h}^{k} + {T_h}^{k-1}}{2\Delta t} - S({T_h}^{k+1}) \quad \text{BDF2} \end{aligned}\right. \tag{5.1}$$

Here, $S(T)$ denotes the stationary part (without time derivative) of the problem at hand.

The matrix of the space and time discretized problem is the sum of the stationary part and a mass matrix multiplied by $1/\Delta t$:

$$\int_\Omega \frac{{T_h}^{k+1} - {T_h}^{k}}{\Delta t} N_i - S'({T_h}^{k+1}, N_i)\,\mathrm{d}\Omega = 0 \quad \forall N_i \tag{5.2}$$

Expanding ${T_h}^{k+1}$ and ${T_h}^{k}$[3] on the basis functions leads to:

$$\sum_j {T_j}^{k+1}\left(\int_\Omega \frac{1}{\Delta t} N_j N_i - S''(N_j, N_i)\,\mathrm{d}\Omega\right) - \sum_j {T_j}^{k}\left(\int_\Omega \frac{1}{\Delta t} N_j N_i\,\mathrm{d}\Omega\right) = 0 \quad \forall N_i \tag{5.3}$$

Eventually, we have to solve the linear system with $\boldsymbol{T}^{k+1}$ as the unknown vector:

$$\left(\frac{\mathsf{M}}{\Delta t} + \mathsf{S}\right)\boldsymbol{T}^{k+1} = \frac{\mathsf{M}}{\Delta t}\boldsymbol{T}^{k} \tag{5.4}$$

where $\mathsf{M}_{ij} = \int_\Omega N_i N_j\,\mathrm{d}\Omega$ is the mass matrix and S is the matrix of the stationary part.

The total matrix $\mathsf{T} = \frac{\mathsf{M}}{\Delta t} + \mathsf{S}$ is generally better conditioned than the sole stationary part S because of greater diagonal dominance. That is, if the discretization is stable for the stationary problem, it will also be stable for the unsteady problem with an implicit finite-difference time discretization.

[1] We can partially get rid of this implicit character by uncoupling the mass and momentum conservation equations via the use of so-called *projection* methods [EG02]. Such methods are out of the scope of these lecture notes.

[2] In order not to solve the linear system involving the mass matrix, so-called *diagonalization* or mass-lumping techniques have been proposed. However, some care must be exercised in using such techniques.

[3] ${T_h}^{k}$ is a known term, it is the unknown's value found at the previous time step or the initial condition if $k = 0$.

5.2 Initial condition

We discuss here the case of a *stiff* initial condition (shock-like). It may even be the case where the initial condition *does not satisfy the boundary conditions or other constraints* (for instance, the $\nabla \cdot \boldsymbol{u} = 0$ constraint of the Stokes or Navier-Stokes' problem). The latter case arises quite often in practice, either because one has been careless about this issue or because it is not easy to build an initial condition that satisfies all the constraints.

In order to deal with this issue, we consider the following unsteady 1D diffusion problem, with α set to 1 and with the Dirichlet boundary conditions:

$$\left\{\begin{array}{l} \dfrac{\partial T}{\partial t} - \operatorname{div} \alpha \nabla T = 0 \quad \text{sur} \quad [0,1] \\ T|_{\boldsymbol{x}=0} = 1 \\ T|_{\boldsymbol{x}=1} = 0 \end{array}\right. \tag{5.5}$$

This problem is an easy one in the sense that the Laplacian operator, as seen in chapter 2, is a regularizing operator.

Discretizing in space and time, we get:

$$\int_\Omega \left\{ \frac{{T_h}^{k+1} - {T_h}^k}{\Delta t} N_i + \alpha \nabla {T_h}^{k+1} \cdot \nabla N_i \right\} \mathrm{d}\Omega = 0 \quad \forall N_i \in \Omega \setminus \delta\Omega \tag{5.6}$$

Thus, we have to solve the linear system with $\boldsymbol{T}^{k+1}$ as an unknown:

$$\left(\frac{\mathsf{M}}{\Delta t} + \alpha \mathsf{R} \right) \boldsymbol{T}^{k+1} = \frac{\mathsf{M}}{\Delta t} \boldsymbol{T}^k \tag{5.7}$$

with the mass matrix $\mathsf{M}_{ij} = \int_\Omega N_i N_j \,\mathrm{d}\Omega$ and the rigidity matrix $\mathsf{R}_{ij} = \int_\Omega \nabla N_i \nabla N_j \,\mathrm{d}\Omega$.

Let us choose the following initial condition:

$${T_h}^0 = 0 \quad \text{on} \quad [0,1] \tag{5.8}$$

This condition *does not satisfy* the boundary conditions of problem (5.5).

The significant part of the Gibiane data file for problem (5.5) is given on listing 5.1.

Figure 5.1 displays the numerical result for a large time step on the left ($\Delta t = 10^{-1}$) and for a small time step on the right ($\Delta t = 10^{-4}$). With a large time step, no oscillation is observed. However, the solution is not time-accurate. With a small time step, some unwanted[4] oscillations occur in the first time step which are then damped due to the Laplacian's regularizing effect.

In fact, it is easy to understand why these oscillations occur. If we write the weak form of the problem at the first time step and let the time step tend to zero, $\Delta t \to 0$:

$$\left\{\begin{array}{l} \displaystyle\int_\Omega \left({T_h}^1 - {T_h}^0 \right) N_i \,\mathrm{d}\Omega = 0 \quad \forall N_i \in \Omega \setminus \delta\Omega \\ T_h|_{\boldsymbol{x}=0} = 1 \\ T_h|_{\boldsymbol{x}=1} = 0 \end{array}\right. \tag{5.9}$$

[4] These oscillations are unwanted because they violate the maximum principle for the heat equation which states that the solution should remain in the bounds given by the boundary conditions ($[0,1]$ in the case at hand).

```
dt = 1.D-1 ;    dt = 1.D-4 ;

....

rv = 'EQEX' 'NITER' 1 'OMEGA' 1. 'ITMA' nitma
     'OPTI' 'EF' 'IMPL' 'CENTREE'
     'ZONE' $mt 'OPER' 'DFDT' 1. 'CNM1' dt 'INCO' 'CN'
     'OPTI' 'EF' 'IMPL' 'CENTREE'
     'ZONE' $mt 'OPER' 'LAPN' 1. 'INCO' 'CN'
     'CLIM' gau 'CN' 'TIMP' 1.
     'CLIM' dro 'CN' 'TIMP' 0. ;
*
rv . 'INCO' = 'TABLE' 'INCO' ;
rv . 'INCO' . 'CN'  = 'KCHT' $mt 'SCAL' 'SOMMET' cini ;
rv . 'INCO' . 'CNM1' = 'KCHT' $mt 'SCAL' 'SOMMET' cini ;
```

Listing 5.1: CastɜM data file `convdif1d.dgibi` corresponding to problem (5.5).

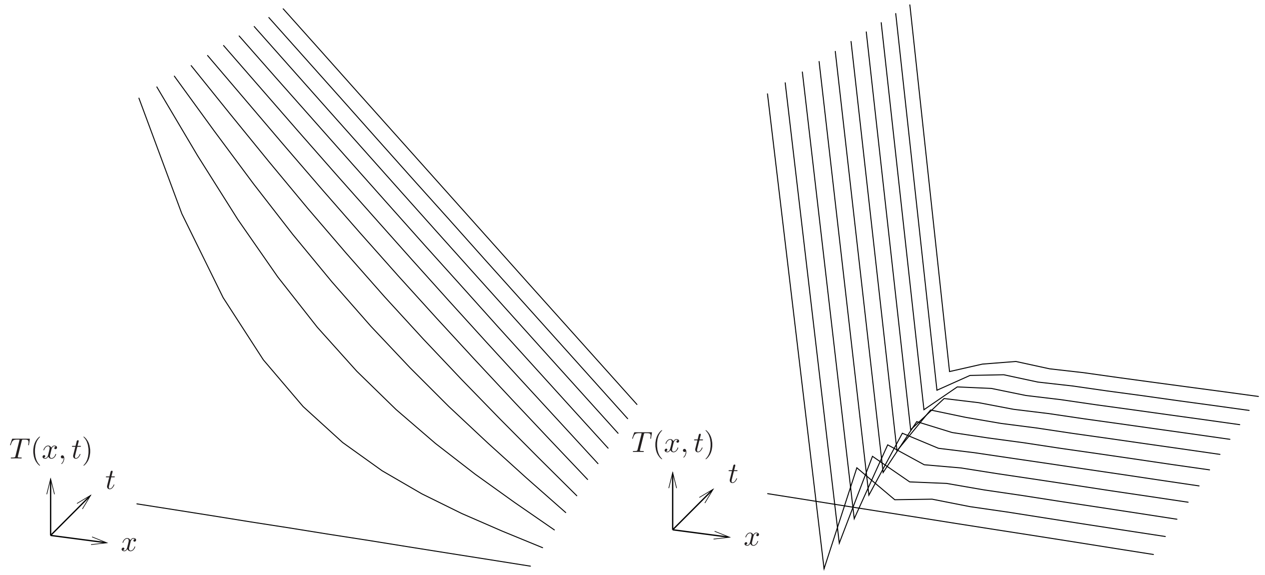


Figure 5.1: Problem (5.5) with an inconsistent initial condition (5.8). Left: $\Delta t = 10^{-1}$. Right: $\Delta t = 10^{-4}$.

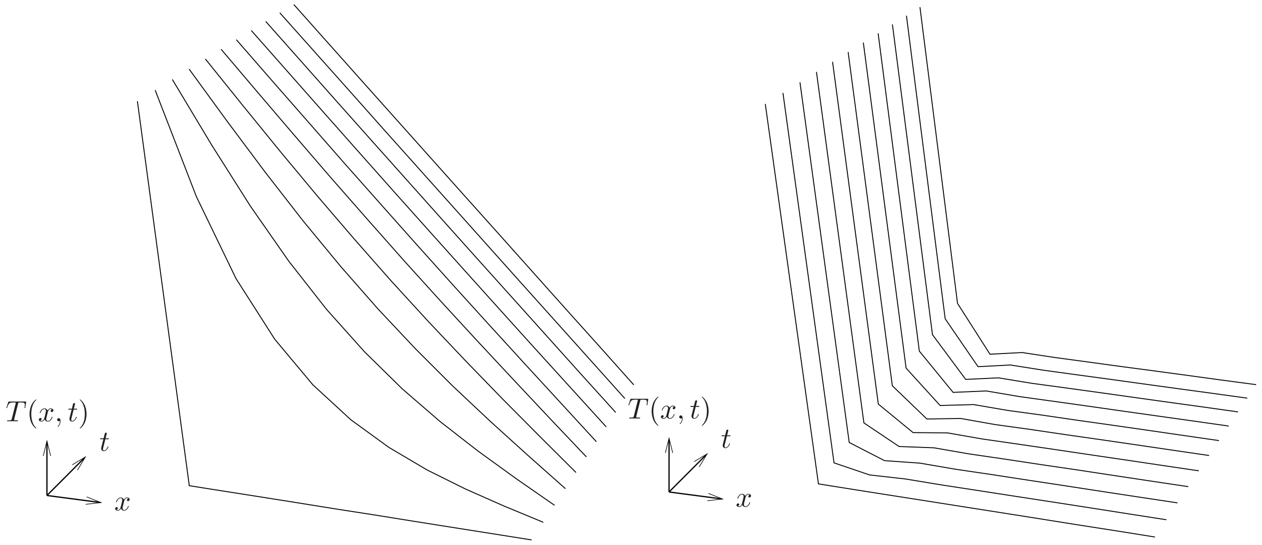


Figure 5.2: Problem (5.5) with an initial condition consistent with the boundary conditions (5.10). Left: $\Delta t = 10^{-1}$. Right: $\Delta t = 10^{-4}$.

Formally, as ${T_h}^0 = 0$, we recover exactly the shock approximation problem of section 4.5.2. The oscillations are due to the fact that the finite element scheme tries to approximate a shock-like function with a continuous function. In the $\mathcal{L}^2$ norm sense, the best approximation is an oscillating function.

Now, if we choose an initial condition which satisfies the boundary conditions:

$$\begin{cases} {T_h}^0 = 1 - \dfrac{x}{\Delta x} \quad \text{sur} \quad [0, \Delta x] \\ {T_h}^0 = 0 \quad \text{sur} \quad [\Delta x, 1] \end{cases} \tag{5.10}$$

where Δx is the first element's length, we obtain the solutions displayed on figure 5.2. With a large time step, the solution is almost identical to the one obtained in the previous case. The Laplacian already has regularized (i.e. forgotten) the initial discontinuity.

With a small time step, the oscillations are much weaker in the case of an initial condition compatible with the boundary conditions. This is because the initial condition is more regular than the shock-like condition and is better approximated by continuous functions in the $\mathcal{L}^2$ norm sense.

For more complex problems, where diffusion effects are not dominant, or where non-linearities are present, the unwanted oscillations due to a bad choice of initial condition can be much longer to dump. They could even be amplified and render the numerical solution totally useless.

5.3 Summary

Upon discretizing in time with an implicit finite-difference method, the time derivative operator leads to a mass matrix divided by the time step Δt that is added to the matrix of the stationary problem, together with a right-hand side.

When the initial condition is inconsistent (i.e. does not satisfy the boundary conditions and/or other constraints) or singular (badly approximated by the basis functions), some oscillations can occur in the numerical solution.

Chapter 6

Solution method for non-linear PDEs

$$\begin{aligned}
\frac{\partial \boldsymbol{u}}{\partial t} + \boxed{(\boldsymbol{\nabla u}) \cdot \boldsymbol{u}} &= -\nabla p^* + \boxed{\nu \boldsymbol{\Delta u}} + \boldsymbol{s_u} \\
\nabla \cdot \boldsymbol{u} &= 0 \\
\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T &= \alpha \Delta T + s_T
\end{aligned}$$

In this chapter, we focus on the numerical resolution of *non-linear* problems. In particular, we study two methods of solution: Newton's method and the method of fixed point iteration (Picard's method). Basically, these methods consist in, first, devising a suitable *linearisation* of the non-linear problem at hand around a given estimate of the solution, and second, solving this linear problem. The solution of the linear problem then serves as a new estimate for the next iteration of the method. A suitable linearisation should make the iteration to converge to the solution of the non-linear problem. In order to linearize a partial differential equation, we will show how to compute the derivative of an operator.

6.1 Newton's method: zero of a function

We recall here the basic principle of Newton's method for finding a zero of a derivable real-valued function f of a single real variable x. We are looking for the solution of:

$$f(x) = 0 \tag{6.1}$$

The main idea of Newton's method is to transform the non-linear problem (6.1), which is not analytically solvable in general, into a sequence of simpler linear problems which we know how to solve. An iteration of Newton's method involves the following steps:

1. an *initial estimate* x_0 of the solution;

2. a *linearization* g of the non-linear function f around the initial estimate x_0: the graph of g is the tangent line to the graph of f at the point of abscissa x_0;

$$g(x) = f'(x_0)(x - x_0) + f(x_0) \tag{6.2}$$

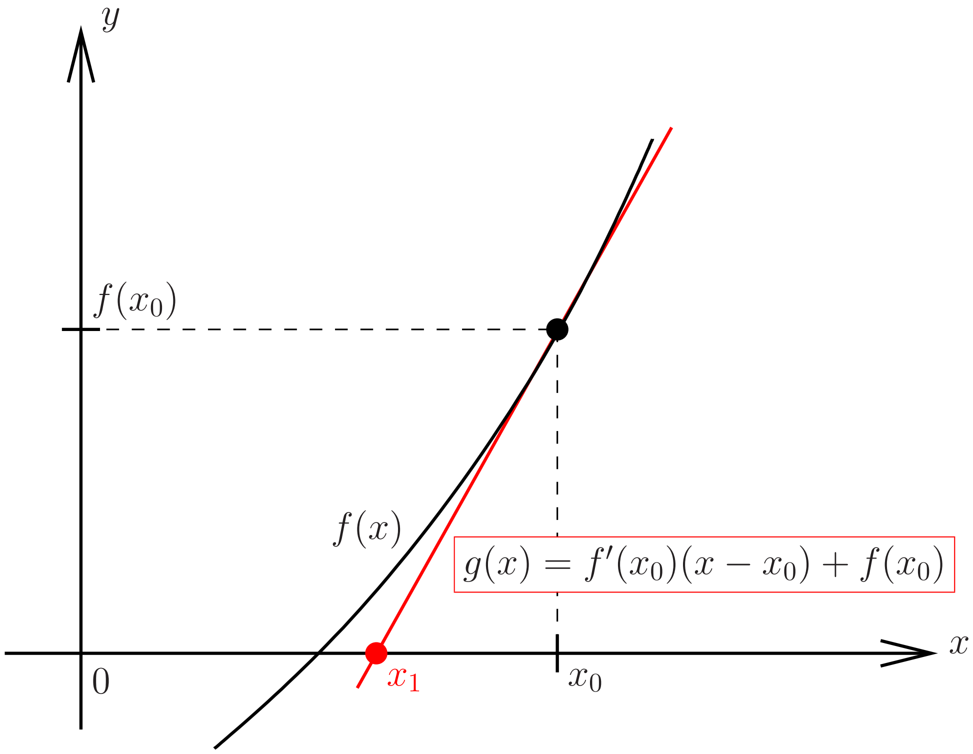


Figure 6.1: An iteration of Newton's method. x_0: initial estimate of the solution. x_1: new estimate of the solution.

3. the *resolution* of the tangent problem:

$$g(x_1) = 0 \tag{6.3}$$

   which gives a new estimate x_1. When Newton's method converge, x_1 will be closer to the solution of (6.1) than x_0.

An iteration of Newton's method is sketched on figure 6.1. Newton's method is then iterated n times until $f(x_n)$ is close enough to 0.

The resolution of the tangent problem (6.3) can be written into two algebraically-equivalent forms:

- an *incremental* form where the unknown Δx is thought as the increment of variable x:

  1. $\Delta x = \frac{1}{f'(x_0)} \times (-f(x_0))$;
  2. $x_1 = x_0 + \Delta x$.

- a *non-incremental* form where the unknown is x_1 itself:

  1. $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$.

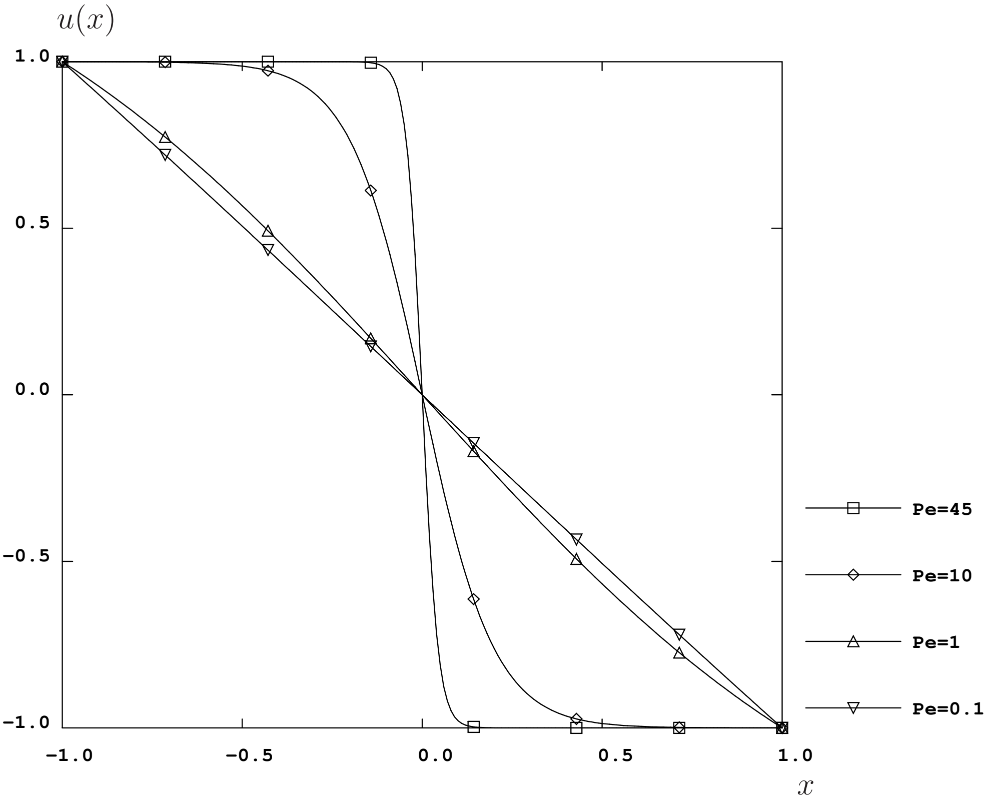


Figure 6.2: Solution $u(x)$ to problem (6.4) as a function of the Péclet number Pe.

6.2 Newton's method: zero of a non-linear PDE

The method of the previous section is generalizable to non-linear partial differential equations, and also to any non-linear problem for which a suitable *derivative* can be defined, such that a solvable tangent problem can be defined.

6.2.1 A model problem

As a model problem we choose the following 1D non-linear convection-diffusion equation defined on the $[-1, 1]$ interval with Dirichlet boundary conditions:

$$\begin{cases} u\dfrac{\partial u}{\partial x} - \dfrac{1}{Pe}\dfrac{\partial^2 u}{\partial x^2} = 0 \\ u\big|_{x=-1} = 1 \\ u\big|_{x=1} = -1 \end{cases} \tag{6.4}$$

This problem is the non-linear extension of the convection-diffusion problem (4.1). This problem solution is similar to an hyperbolic tangent function. Its slope at $x = 0$ increases with the Péclet number Pe, as displayed in figure 6.2.

6.2.2 Operator derivative

The model problem is formally written as:

$$R(u) = 0 \tag{6.5}$$

where R is an *operator*. It is an object which, given a function, provides as an output another function. Here, R, given the unknown function $u(x)$, returns the following function: $u\frac{\partial u}{\partial x} - \frac{1}{Pe}\frac{\partial^2 u}{\partial x^2}$. R is frequently called a *residual* operator. It results in a null-valued function if we provide it with the solution u of the problem. In the context of solid mechanics, R is also named the *disequilibrium* operator.

In chapter 2 we defined the functional derivative. In the same way we define the operator derivative:

$$\delta_v R(u) = \lim_{\epsilon \to 0} \frac{\mathrm{d}}{\mathrm{d}\epsilon} R(u + \epsilon v) \tag{6.6}$$

This formula reads: the operator derivative of R at "point" u along the "direction" v equals... In the formula, derivation with respect to ϵ is carried out while u and v are kept fixed.

The operator derivative is also an operator which, given two functions u and v, provides as an output another function $\delta_v R(u)$. An important fact is that $\delta_v R(u)$ is *linear* with respect to v.

Applying definition (6.6) to our model problem (6.4), one gets:

$$\delta_v R(u) = \lim_{\epsilon \to 0} \frac{\mathrm{d}}{\mathrm{d}\epsilon} \left(u\frac{\partial u}{\partial x} + \epsilon \left(v\frac{\partial u}{\partial x} + u\frac{\partial v}{\partial x} \right) + \epsilon^2 v\frac{\partial v}{\partial x} - \frac{1}{Pe}\frac{\partial^2 u}{\partial x^2} - \epsilon\frac{1}{Pe}\frac{\partial^2 v}{\partial x^2} \right) \tag{6.7}$$

that is:

$$\delta_v R(u) = \left(v\frac{\partial u}{\partial x} + u\frac{\partial v}{\partial x} \right) - \frac{1}{Pe}\frac{\partial^2 v}{\partial x^2} \tag{6.8}$$

The third term in the right-hand side does not depend on the derivation "point" u. This is due to the fact that $-\frac{1}{Pe}\frac{\partial^2}{\partial x^2}$ is a linear operator.

6.2.3 Newton's iteration

Now, we can present the iteration process of Newton's method for the solution of the non-linear problem (6.5):

1. *initial estimate* of the thought solution u_0;

2. *linearization* S of the residual R around the initial estimate u_0:

$$S(u) = \delta_{(u-u_0)} R(u_0) + R(u_0) \tag{6.9}$$

3. *resolution* of the tangent problem:

$$S(u_1) = 0 \tag{6.10}$$

which gives a new estimate u_1. When Newton's method converges, u_1 will be closer to the solution of (6.5) than u_0.

Newton's method is then iterated n times until $R(u_n)$ is close enough to the null-valued function.

The resolution of the tangent problem (6.3) can be written into two algebraically-equivalent forms:

- an incremental form where the unknown function Δu is the increment of the function u:

    1. Solve $\delta_{\Delta u} R(u_0) = -R(u_0)$;
    2. $u_1 = u_0 + \Delta u$.

- a non-incremental form where the unknown function is u_1 itself:

    1. Solve $\delta_{u_1} R(u_0) = -R(u_0) + \delta_{u_0} R(u_0)$.

Remark 1 *Going from the incremental form to the non-incremental form uses the fact that $\delta_v R(u)$ is linear in v.*

Remark 2 *Looking more closely at the incremental form, we can see that in the scalar case, the resolution of the tangent problem involves a* division *by $f'(x_0)$ of the right-hand side $(-f(x_0))$. In the PDE case, the tangent problem involves the* resolution of a non-homogeneous linear PDE *$\delta_{\Delta u} R(u_0) = -R(u_0)$.*

6.2.4 Application to the model problem

Let us apply Newton's method to the model problem (6.4). In incremental form the first iteration writes:

1. given u_0;
2. solve the following linear PDE to find the unknown Δu:

$$\begin{cases} \Delta u \dfrac{\partial u_0}{\partial x} + u_0 \dfrac{\partial \Delta u}{\partial x} - \dfrac{1}{Pe} \dfrac{\partial^2 \Delta u}{\partial x^2} = -u_0 \dfrac{\partial u_0}{\partial x} - \dfrac{1}{Pe} \dfrac{\partial^2 u_0}{\partial x^2} \\ \Delta u\big|_{x=-1} = 1 - u_0\big|_{x=-1} \\ \Delta u\big|_{x=1} = -1 - u_0\big|_{x=1} \end{cases} \tag{6.11}$$

3. new estimate: $u_1 = u_0 + \Delta u$.

In non-incremental form, the first iteration writes:

1. given u_0;
2. solve the following linear PDE to find unknown u_1:

$$\begin{cases} u_1 \dfrac{\partial u_0}{\partial x} + u_0 \dfrac{\partial u_1}{\partial x} - \dfrac{1}{Pe} \dfrac{\partial^2 u_1}{\partial x^2} = +u_0 \dfrac{\partial u_0}{\partial x} \\ u_1\big|_{x=-1} = 1 \\ u_1\big|_{x=1} = -1 \end{cases} \tag{6.12}$$

The incremental and non-incremental form are algebraically equivalent but, in practice, each has its advantages:

Non-incremental form: in general, there are less terms to compute and the incrementation step is not needed;

Incremental form: this form explicitly makes use of the increment Δu and the residual R. The norm of these quantities allows one to estimate respectively the error on the unknown (primal error) and the error on the equation (dual error). These are very useful indicators of the convergence behavior of Newton's method.

In the Cast3M code, the incremental form is generally preferred in the context of solid mechanics (`PASAPAS` procedure) while the non-incremental form is used in the context of fluid mechanics (`EXEC` procedure, see chapter 10).

6.3 Picard's method

In general, Newton's method exhibits fast convergence to the solution of a given non-linear problem when the initial estimate is sufficiently close to this solution. In the opposite case, one frequently resorts to variants of Newton's method which are more robust, that is less sensitive to the choice of the initial estimate.

One such method is the method of *fixed-point* iteration, also called *Picard*'s method. This method can be interpreted as a Newton's method where the exact derivative ($f'(x_0)$ or $\delta R(u_0)$) is replaced with an approximation. For example, in the case of the model problem (6.4), instead of the exact expression (6.8), one can use:

$$\widetilde{\delta_v} R(u) = u\frac{\partial v}{\partial x} - \frac{1}{Pe}\frac{\partial^2 v}{\partial x^2} \tag{6.13}$$

We have discarded the first term of the exact linearization of the $u\frac{\partial u}{\partial x}$ operator and we have kept the second term[1].

6.3.1 Incremental form

In incremental form, an iteration of Picard's method writes:

1. Solve $\widetilde{\delta_{\Delta u}} R(u_0) = -R(u_0)$;
2. $u_1 = u_0 + \Delta u$.

An important fact is that only the left-hand side in step 1 has been modified: if Picard's method converge, it converges to a solution u such that $R(u) = 0$, identically to Newton's method!

6.3.2 Non-incremental form

In non-incremental form, the first fixed-point iteration applied to the model problem (6.4) writes:

1. given u_0;

[1] The choice of the terms to keep is frequently done heuristically, based on experience, or dictated by other considerations (complexity, number of terms to compute...).

2. solve the following linear PDE to find unknown u_1 :

$$\begin{cases} u_0 \dfrac{\partial u_1}{\partial x} - \dfrac{1}{Pe} \dfrac{\partial^2 u_1}{\partial x^2} = 0 \\ u_1\big|_{x=-1} = 1 \\ u_1\big|_{x=1} = -1 \end{cases} \tag{6.14}$$

One can see that a fixed-point iteration is equivalent to solving the problem (6.4) where the non-linear term $u\frac{\partial u}{\partial x}$ has been replaced with $u_0\frac{\partial u}{\partial x}$: the convection speed has been fixed to the value of the initial estimate u_0, hence the name of fixed-point iteration.

Picard's method in non-incremental form is the method used in the `EXEC` procedure to solve non-linear equations. More details are given in chapter 10.

6.4 Numerical examples

We examine what has just been presented by solving the model problem (6.4) using Newton's method or Picard's method and choosing the following parameters:

- regular 1D mesh of 100 elements;
- spatial discretization with linear finite elements;
- $Pe = 10$ or $Pe = 45$;
- initial estimate: $u_0(x) = -x$.

Once the problem has been discretized, using for instance Newton's method in incremental form (6.11), the first iteration is written in matrix form:

1. given $\boldsymbol{u}_0$;
2. solve the following linear system to find unknown $\Delta\boldsymbol{u}$:

$$\mathsf{N}_{\boldsymbol{u}_0} \Delta\boldsymbol{u} = \boldsymbol{b}(\boldsymbol{u}_0) \tag{6.15}$$

3. new estimate: $\boldsymbol{u}_1 = \boldsymbol{u}_0 + \Delta\boldsymbol{u}$.

The matrix $\mathsf{N}_{\boldsymbol{u}_0}$ varies from iteration to iteration because it depends on $\boldsymbol{u}_0$.

On figure 6.3 we display the approximate solution u_i found by each method has a function of the number of non-linear iterations i, for a Péclet number $Pe = 10$. The two methods converge to the same solution but we observe a difference in behavior for the first iterations where Newton's method computes approximate solutions out of the interval $[-1, 1]$.

On figure 6.4 we plot the error $\max |u_i - u_\infty|$ as a function of the iteration number i for Newton's method and Picard's method for a Péclet number $Pe = 10$. We notice the faster convergence of Newton's method compared to Picard's method once the estimate is sufficiently close to the solution (here $i \geq 2$).

On figure 6.5 we plot the error for a larger Péclet number $Pe = 45$: now Newton's method diverges, whereas the more robust Picard's method still converges.

The difference in behavior observed for Newton's and Picard's method on this particular example are also frequently observed in other problems.

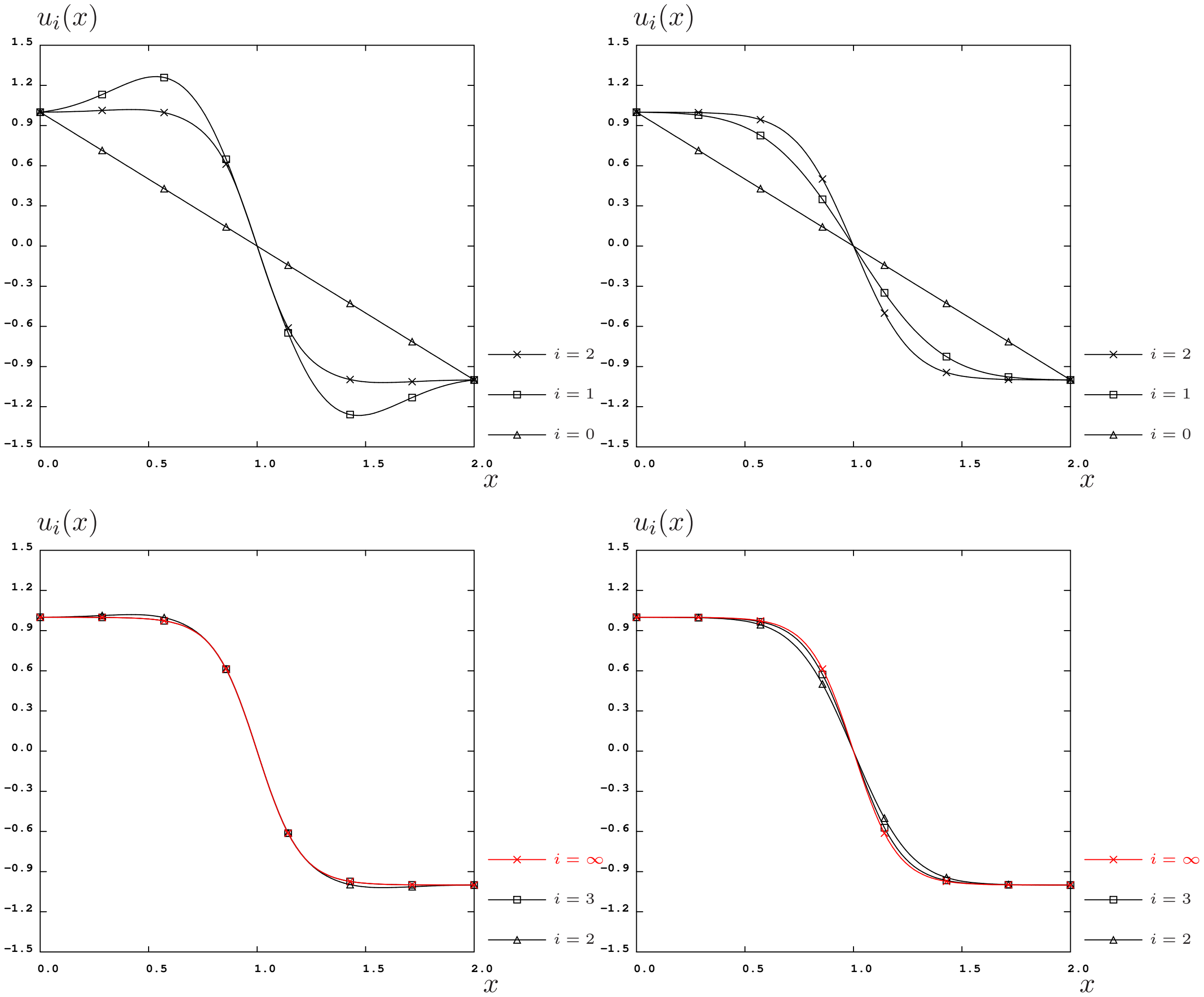


Figure 6.3: Approximate solution u_i to problem (6.4) for a Péclet number $Pe = 10$ at iteration number i. Left: Newton's method. Right: Picard's method. Up: $i < 3$. Down: $i > 2$.

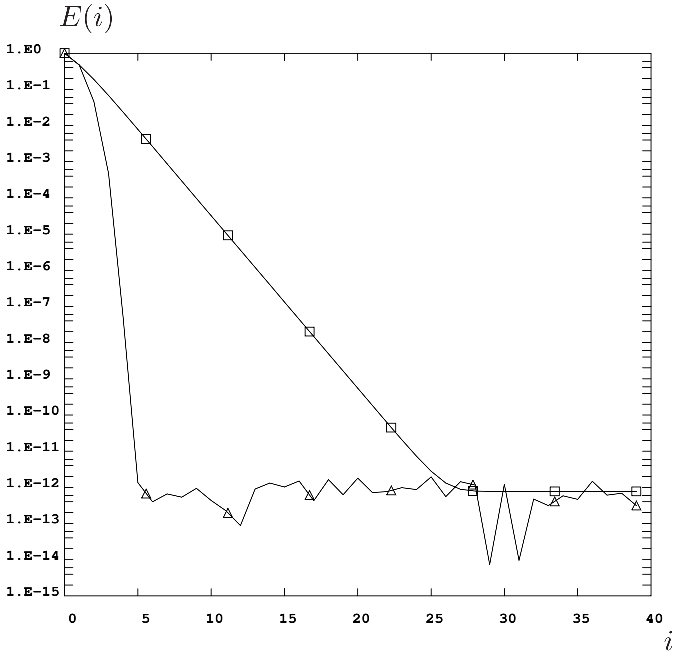


Figure 6.4: Error $E = \max |u_i - u_\infty|$ as a function of the iteration number i for the model problem (6.4) and Péclet number $Pe = 10$. □: Picard's method. △: Newton's method.

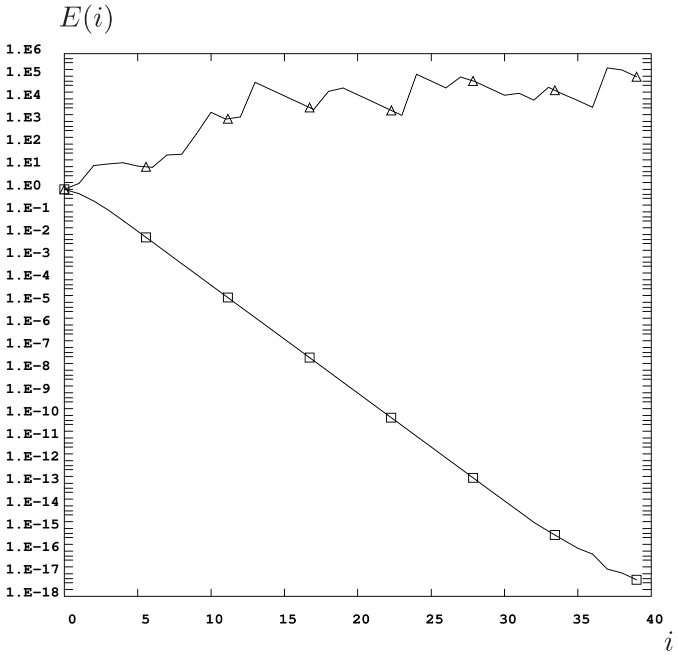


Figure 6.5: Error $E = \max |u_i - u_\infty|$ as a function of the iteration number i for the model problem (6.4) and Péclet number $Pe = 45$. □: Picard's method. △: Newton's method.

6.5 Summary

In order to solve a non-linear problem, we need an initial estimate of the solution. Then we linearize the problem around this initial estimate. The resolution of this simplified linear problem gives us a new, hopefully better, estimate of the solution. This process is iterated until sufficient accuracy is reached.

The linearization process involves the definition of a derivative: for PDEs, the operator derivative is an appropriate one.

When the linearization is exact, the method is called Newton's method. Its main feature is fast convergence to the solution for a close enough initial estimate.

When the linearization is approximate, we obtain other methods such as Picard's method of fixed-point iterations. These methods are frequently more robust and less costly per iteration than Newton's method at the price of a slower speed of convergence to the solution.

Chapter 7

Shock formation

$$\begin{aligned}
\boxed{\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\,\cdot \boldsymbol{u}} &= -\nabla p^* + \nu \boldsymbol{\Delta u} + \boldsymbol{s_u} \\
\nabla \cdot \boldsymbol{u} &= 0 \\
\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T &= \alpha \Delta T + s_T
\end{aligned}$$

7.1 Burgers' equation

We now turn to one of the simplest non-linear unsteady problem: the unsteady *Burgers'* equation. We can write it in several ways, for example in *conservative* form:

$$\frac{\partial u}{\partial t} + \nabla \cdot \frac{u^2}{2} = \frac{\partial u}{\partial t} + \nabla \cdot f(u) = 0 \tag{7.1}$$

where $f(u)$ is called a flux. The flux depends on the extensive unknown quantity u which is conserved. The fact that equation $\frac{\partial u}{\partial t} + \nabla \cdot f(u) = 0$ expresses the conservation of u will appear more clearly when we write the corresponding weak integral form (7.4).

Burgers' equation can also be written in *non-conservative* form, which is equivalent to the conservative form as far as PDE are concerned. In 1D, the non-conservative form is:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{7.2}$$

We see that the unknown u plays two roles in the non-linear convection term $u\frac{\partial u}{\partial x}$:

- a passive role as a convected unknown $u\dfrac{\partial \boxed{u}}{\partial x}$;
- an active role as a convecting velocity $\boxed{u}\dfrac{\partial u}{\partial x}$.

This dual role leads to interesting properties.

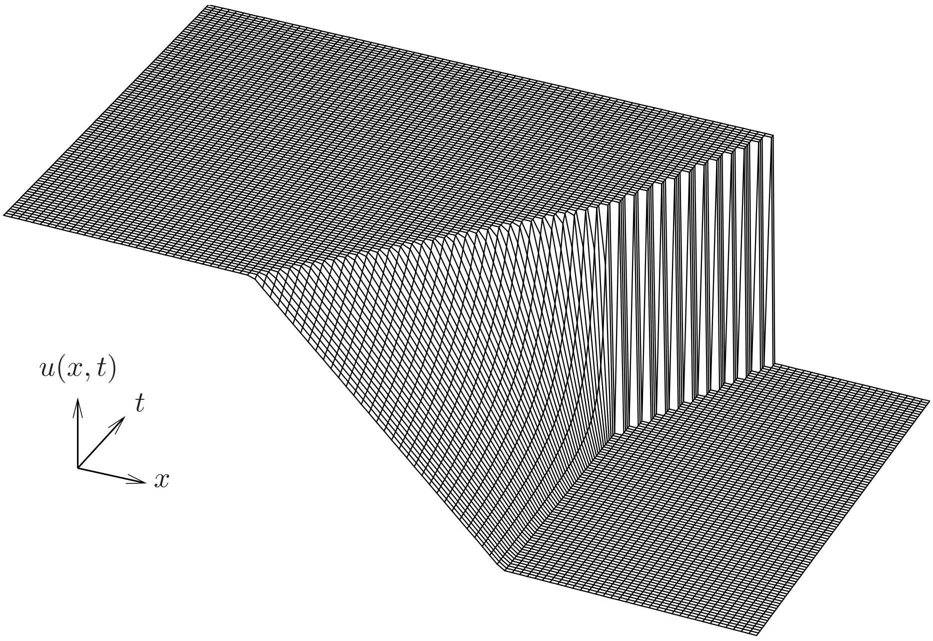


Figure 7.1: Exact solution $u(x,t)$ to problem (7.2) (Burgers' equations) with initial condition (7.3). Shock formation.

7.2 Shock formation

For example, let us consider the following initial condition:

$$\begin{cases} u^0(x) = 1 \quad \text{if} \quad x < 0 \\ u^0(x) = 1 - x \quad \text{if} \quad 0 \leq x < 1 \\ u^0(x) = 0 \quad \text{if} \quad x \geq 1 \end{cases} \tag{7.3}$$

The exact solution to the unsteady 1D Burgers' equation with this initial condition in space and time is displayed on figure 7.1. We can describe qualitatively what is going on:

- the left part of the solution, with $u = 1$, will travel to the right with unit velocity;
- the right part of the solution, with $u = 0$, will remain steady.

When the left and right part of the solution meet, the *shock* formation takes place.

This shock is somewhat problematic because it means that the solution becomes discontinuous and we cannot make sense of the partial derivative $\frac{\partial u}{\partial x}$ in the PDE (7.2) any more. Then, in the vicinity of the shock we have to get back to the more general weak form of the conservation equation. This weak form is obtained by integration in space and time of the PDE (7.1)[1]. It gives:

$$\int_t^{t+\Delta t} \int_x^{x+\Delta x} \frac{\partial u}{\partial t} + \nabla \cdot f(u) \mathrm{d}t\mathrm{d}x = 0 \quad \forall t, x, \Delta t, \Delta x \tag{7.4}$$

[1] In fact, as we did in chapter 3 for the weighted residual method, we are following a kind of reversed path. Indeed, it could be argued that the weak form is the more physical way of expressing conservation. The PDE form is deduced from the weak form under regularity hypothesis that don't hold in the vicinity of the shock. However, the PDE form has the advantage of being more concise and easier to manipulate algebraically.

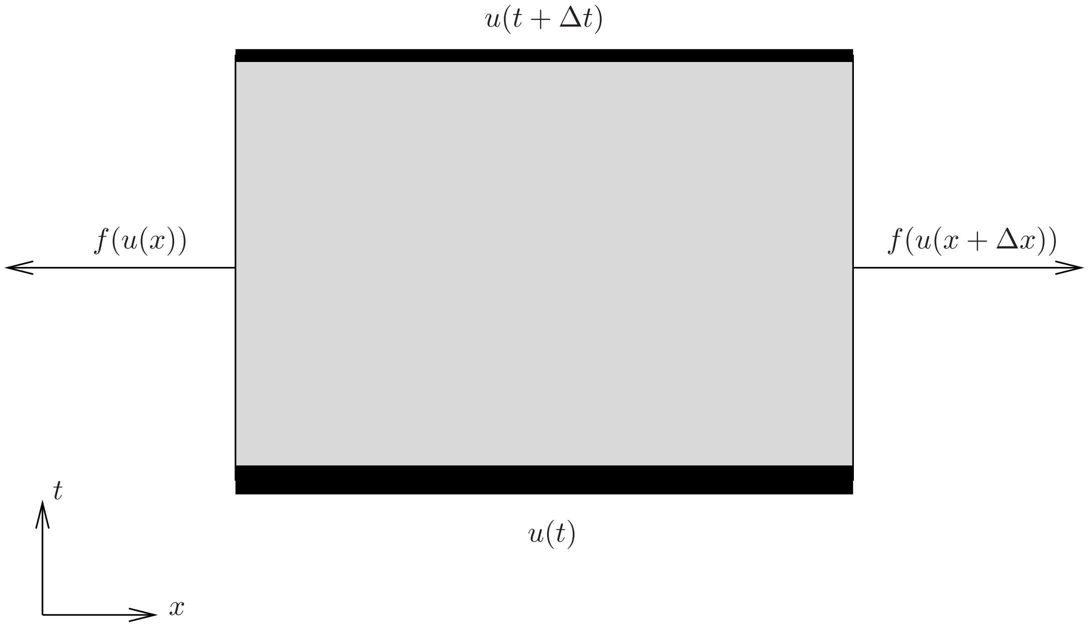


Figure 7.2: Balance of the extensive quantity u on the space-time interval $[x, x+\Delta x] \times [t, t+\Delta t]$ (equation (7.5))

or:

$$\int_x^{x+\Delta x} u(t+\Delta t) - u(t)\mathrm{d}x + \int_t^{t+\Delta t} f(u(x+\Delta x)) - f(u(x))\mathrm{d}t = 0 \quad \forall t, x, \Delta t, \Delta x \tag{7.5}$$

which reads: the increase in the extensive quantity u, present in an arbitrary space interval $[x, x+\Delta x]$, for an arbitrary time interval $[t, t+\Delta t]$, is equal to the quantity of u (the flux of u) flowing through the boundaries, at x and $x+\Delta x$, within the same time interval. This balance is represented in figure 7.2.

The weak form of the conservation equation still holds when the solution is a shock wave and allows us to write *jump relations*. These relations establish a link between the velocity of the shock and the left and right state around the shock. They are called *Rankine-Hugoniot relations*. We obtain them by writing the conservation equation on a space-time interval $[x, x+dx] \times [t, t+dt]$ in the presence of a shock, traveling at velocity s, where the solution goes from the upwind (left) value u_{L} to the downwind value (right) u_{R} (figure 7.3):

$$\begin{aligned} \int_x^{x+\Delta x} u(t+\Delta t) - u(t)\mathrm{d}x + \int_t^{t+\Delta t} f(u(x+\Delta x)) - f(u(x))\mathrm{d}t = 0 \quad \forall t, x, \Delta t, \Delta x \\ \Rightarrow dt(s(u_{\mathrm{L}} - u_{\mathrm{R}})) + dt(f(u_{\mathrm{R}}) - f(u_{\mathrm{L}})) = 0 \\ \Rightarrow s = \frac{f(u_{\mathrm{L}}) - f(u_{\mathrm{R}})}{u_{\mathrm{L}} - u_{\mathrm{R}}} = \frac{u_{\mathrm{L}} + u_{\mathrm{R}}}{2} \end{aligned} \tag{7.6}$$

Thus, once the shock is formed, it travels with a velocity equal to the arithmetic mean of the upwind and downwind velocities. Practically, we can use the PDE form of the conservation equation where the solution is smooth and elsewhere, in the presence of a shock identified by its location, we can characterize its velocity with the help of Rankine-Hugoniot's relations.

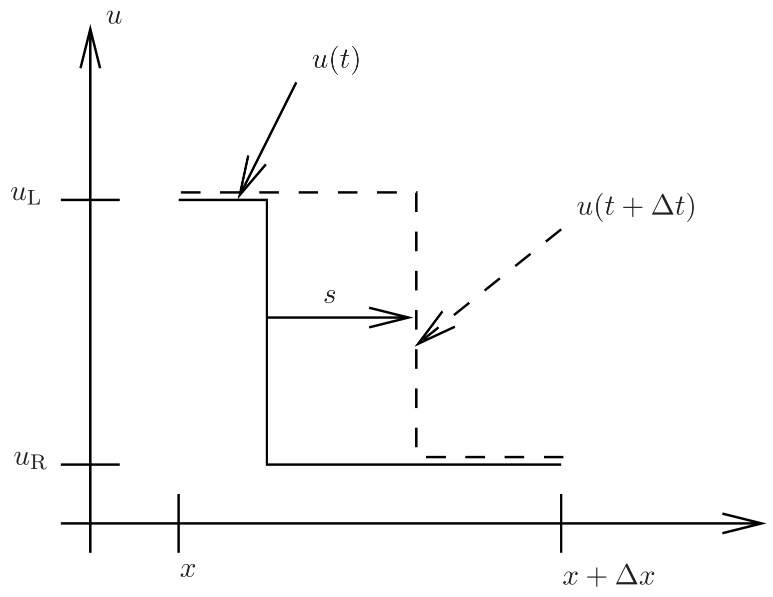


Figure 7.3: Variation of the extensive quantity u on an interval $[x, x+\Delta x]$ in the presence of a shock (equation (7.6)).

7.3 Rarefaction wave

However, that is not all we have to say about Burgers' equation. Let us now consider the following initial condition which is a shock in reverse:

$$\left\{\begin{array}{lll} u^0(x) = 0 & \text{if} & x < 0 \\ u^0(x) = 1 & \text{if} & x \geq 0 \end{array}\right. \tag{7.7}$$

One can easily check that the two functions plotted on figure 7.4 are solutions of Burgers' problem. Indeed, they verify PDE (7.2) where they are continuous and, across the non-physical shock, Rankine-Hugoniot's relations hold. Burgers' problem is thus *ill-posed* in the sense that it has more than one solution. Heuristically, this means that some physical conditions are still lacking.

Exercise 4 *In fact, Burgers' problem admits an infinity of solutions with non-physical shocks. Find another one.*

One way to make Burgers' problem better posed, so that it has a unique solution, is to add supplementary *entropy conditions* [Lev02] and to consider the vanishing-viscosity limit of the following well-posed problem:

$$\lim_{\epsilon \to 0} \left(\frac{\partial u}{\partial t} + \nabla \cdot \frac{u^2}{2} - \epsilon \Delta u = 0 \right) \tag{7.8}$$

In this case, we obtain the solution displayed on the right of figure 7.4 which is called a *rarefaction wave*.

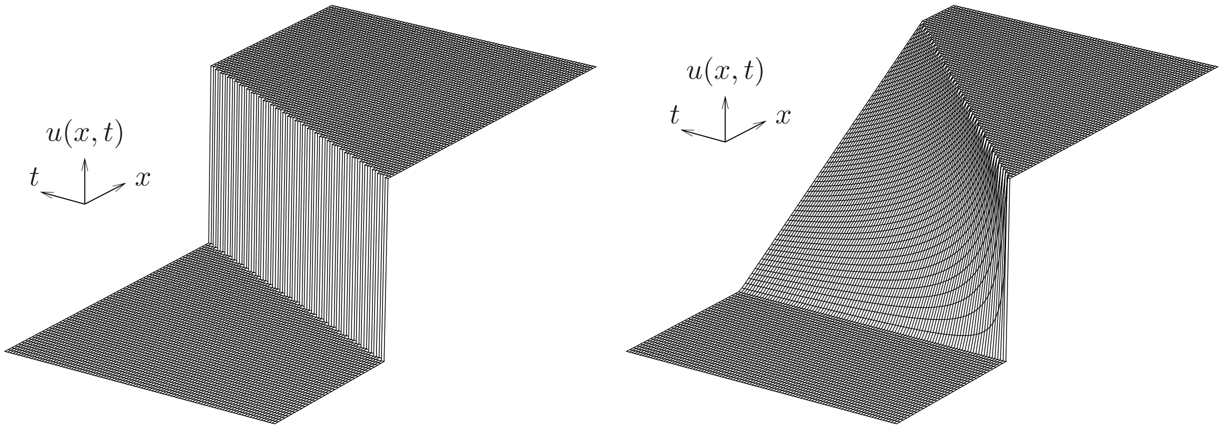


Figure 7.4: Two exact solutions $u(x,t)$ to problem (7.1) (Burgers’ equation) with an initial condition (7.7). Left: non-physical shock. Right: rarefaction wave.

```
typdec = ’CENTREE’ ;
typdec = ’SUPG’ ;
nmail = 10 ; cfl = 0.5 ; dt = ’/’ cfl nmail ; tfinal = 2. ;
nitma = ’ENTIER’ (’/’ tfinal dt) ;
*
rv = ’EQEX’ ’NITER’ niter ’OMEGA’ omeg ’ITMA’ nitma
     ’OPTI’ ’EF’ ’IMPL’ ’CENTREE’
     ’ZONE’ $mt ’OPER’ ’DFDT’ 1. ’CNM1’ dt ’INCO’ ’CN’ ;
     ’ZONE’ $mt ’OPER’ MAJUN
     ’OPTI’ ’EF’ ’IMPL’ typdec
     ’ZONE’ $mt ’OPER’ ’KONV’ 1. ’UN’ ’ALF’ ’INCO’ ’CN’
     ’OPTI’ ’EF’ ’IMPL’ ’CENTREE’
     ’ZONE’ $mt ’OPER’ ’LAPN’ ’ALF’ ’INCO’ ’CN’
     ’CLIM’ gau ’CN’ ’TIMP’ cgau
     ’CLIM’ dro ’CN’ ’TIMP’ cdro ;
```

Listing 7.1: Cast3M data file `burgers1d.dgibi` corresponding to problem (7.2).

7.4 Numerical examples

We now consider the numerical solutions to Burgers’ problem when we discretize equation (7.2) with the finite element method[2].

The significant part of the Gibiane data file for problem (7.2) is given on listing 7.1.

Shock We first deal with the shock formation case, corresponding to the initial condition (7.3). Figure 7.5 displays the numerical results obtained with a centered discretization for the convective term (on the left) and an upwind SUPG discretization (on the right). The centered discretization of the convective term leads to an oscillating solution as soon as the shock is formed. These oscillations are amplified due to the problem non-linearity and contaminate all the computational domain, making the numerical solution so obtained

[2] Prior to applying the weighted residual method, we need to discretize Burgers’ problem in time (chapter 5) and to *linearize* the non-linear convection term (chapter 6). The entire algorithm is described in chapter 10.

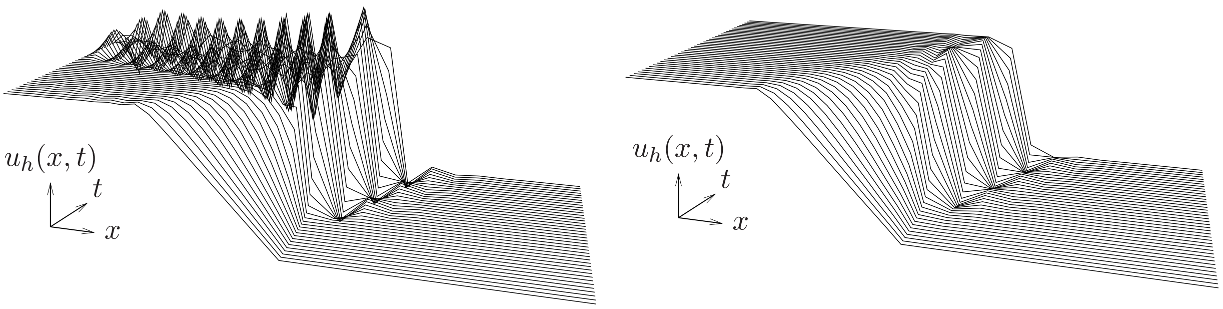


Figure 7.5: Numerical solution $u_h(x,t)$ to problem (7.2) (Burgers' equation) for an initial condition (7.3) giving rise to a shock formation. Left: centered discretization (CENTREE) for the convective term. Right: upwind discretization (SUPG) for the convective term.

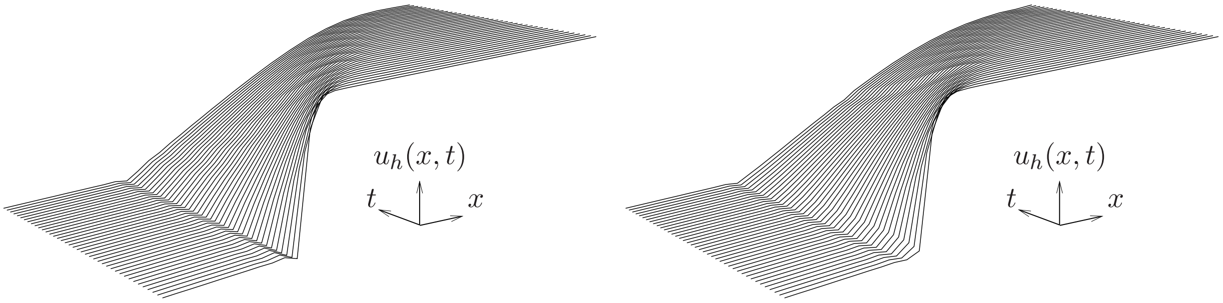


Figure 7.6: Numerical solution $u_h(x,t)$ to problem (7.2) (Burgers' equation) for an initial condition (7.7) giving rise to a rarefaction wave. Left: centered discretization (CENTREE) for the convective term. Right: upwind discretization (SUPG) for the convective term.

almost useless. If we use an upwind discretization (SUPG), some oscillations still remains because of the Gibbs' phenomenon (see section 4.5) but they are always located in the vicinity of the shock.

Rarefaction wave We now deal with the rarefaction wave case, corresponding to the initial condition (7.7). The numerical results are shown on figure 7.6. Both the centered and upwind discretization of the convective term give rise to the rarefaction wave solution. For the upwind (SUPG) discretization, the added numerical diffusion helps the numerical scheme to select the correct rarefaction wave solution. In the case of the centered discretization, we notice that a small oscillation remains and that it is not damped in time. This oscillation is located near the place where the shock in reverse was initially prescribed. This is in agreement with the remark we made at the end of section 5.2 where the importance of the correct prescription of the initial condition, particularly for non-linear problems, was emphasized.

There remains much to say about the solution of PDE related to conservation laws. In particular, one important PDE system of conservation laws is *Euler's equations* which corresponds to Navier-Stokes' equations without *diffusive terms*. For the interested reader, we mention the book of Leveque [Lev02].

The main point of this chapter is that non-linearity can lead to singular solutions, like shocks, even if the initial condition is regular.

7.5 Midway summary

Let us summarize some important issues that we have dealt with in this chapter and the previous ones:

1. a centered discretization (for example the standard finite element method) of a purely convective problem is, in general, not *stable*. One needs to add some upwinding, i.e. numerical diffusion, for the discrete problem to be well-posed;

2. the finite element method with continuous approximation basis functions approximates optimally the problem solution in the $\mathcal{L}^2$-norm: the discrete solution can oscillate when the exact solution undergoes fast variation (Gibbs' phenomenon). This happens *even if* the discrete problem is well-posed.

3. the Laplacian operator is a regularizing operator: it has an important role in the well-posedness of discretizations for pure convection problems. It also smooths a singular (shock-like) initial condition in time in the context of unsteady diffusion problems.

4. non-linearities in the equations may produce singular solutions even when the initial condition is regular.

In practice, on the model problems we have solved up to now, the finite element has two different kind of issues:

1. the stability of the discrete problem. If the discrete problem is not stable, the discrete solution can be spoiled by unbounded oscillations;

2. the approximation of singular functions (shock-like) by regular (continuous) basis functions. In fact, shock-like in the present context is a discretization-dependent notion. Once a mesh has been fixed, every function that can't be adequately represented (interpolated) on this mesh can be qualified as shock-like. If the exact solution is such a function, then the numerical solution can undergo oscillations, but these remain bounded in general.

The two issues are in fact *separate*: the stability of the discrete problem is a necessary condition if we want to have *convergence* of the discrete solution to the exact solution. The second issue warns us when the mesh is inadequate for *correctly* approximating the exact solution.

It is also important to look at the characteristic scale of each phenomenon in order to determine how they will interact:

- convective effects dominate on large space scale and small time scale. They can also have a stiffening effect on the solution in the non-linear case;

- on the opposite, diffusive effects dominate on small space scale and large time scale. They have a regularizing effect on the solution.

When the characteristic scales of the involved phenomena are different, we are dealing with a multiscale problem. In the context of fluid mechanics, *shock* formation, *boundary layer* problems and *turbulence* are common examples of such multiscale problems. The discretization process also adds its own scales to the studied problem: a space scale equal to the mesh elements' size and a time scale equal to the time-step size. This means that a given discretization may not be able to capture all the scales of the studied problem: the phenomena below the discretization scale will be cut off. Thus, we might need to *model* these phenomena because we are not able to *capture* them.

In fact, this multiscale character is what makes fluid mechanics so interesting and varied. It is also the source of the many difficulties encountered in its mathematical analysis and its physical and numerical modeling.

Chapter 8

Stokes' problem

$$\begin{aligned}\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla} \boldsymbol{u}) \cdot \boldsymbol{u} &= \boxed{-\nabla p^* + \nu \boldsymbol{\Delta} \boldsymbol{u}} + \boldsymbol{s}_{\boldsymbol{u}} \\ \boxed{\nabla \cdot \boldsymbol{u}} &= 0 \\ \frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T &= \alpha \Delta T + s_T\end{aligned}$$

In this chapter, we focus on the discretization of *Stokes'* problem. Similar to Dirichlet's problem of chapter 2, Stokes' problem can be interpreted as an optimization problem. But this time, we have to deal with a *constrained* minimization problem instead of the simpler unconstrained minimization in the Dirichlet case. This has important implications on the stability of the discretized Stokes' problem.

Thus the presentation of Stokes' problem in this chapter closely mimics the presentation of Dirichlet's problem in chapter 2 with an added subsection on stability (8.2.3).

8.1 The continuous Stokes problem

8.1.1 Stokes' functional

Stokes' problem is closely related to Navier's equations (2.29)-(2.30), that we recall here:

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \boldsymbol{f} = 0 \tag{8.1}$$

$$\boldsymbol{\sigma} = \lambda(\nabla \cdot \boldsymbol{u})\mathsf{I} + \mu \left(\boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla}^t \boldsymbol{u}\right) \tag{8.2}$$

When the first Lamé coefficient tends to infinity: $\lambda \to \infty$, given the expression of Navier's functional (2.25), we have: $\nabla \cdot \boldsymbol{u} \to 0$. If we define a new unknown: $p = \lambda \nabla \cdot \boldsymbol{u}$, which can be identified with a pressure, the stress tensor writes:

$$\boldsymbol{\sigma} = p\mathsf{I} + \mu \left(\boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla}^t \boldsymbol{u}\right) \tag{8.3}$$

Notice that the meaning of $\boldsymbol{u}$ has changed. In the context of linear elasticity, $\boldsymbol{u}$ was the displacement (defined with respect to a reference configuration and expected to be small). In the context of fluid mechanics, $\boldsymbol{u}$ is the local velocity of the fluid. A solid deforms, a fluid flows.

The optimization principle we write now involves the following functional, which depends on two functions of the space variables $\boldsymbol{u}$ and p:

$$I(\boldsymbol{u},p) = \int_\Omega \frac{\mu}{2}\|\boldsymbol{\nabla}\boldsymbol{u}\|^2 - p\nabla\cdot\boldsymbol{u}\,\mathrm{d}\Omega \tag{8.4}$$

This functional is linked to Stokes' problem. It does not have a minimum (it is not a sum of squares as in Dirichlet's functional) but it does have a *saddle-point*. Its structure is that of a constrained minimization problem. In fact, the problem solution $(\boldsymbol{u},p)$ minimizes $\int_\Omega \frac{\mu}{2}\|\boldsymbol{\nabla}\boldsymbol{u}\|^2\,\mathrm{d}\Omega$ under the constraint $\nabla\cdot\boldsymbol{u} = 0$. This constraint is prescribed with the help of the variable p which is called a *Lagrange multiplier*.

8.1.2 Saddle-point condition

Under suitable assumptions on the regularity of I, a necessary condition for $(\boldsymbol{u},p)$ to be a saddle-point is to write that the functional derivative of I vanishes at $(\boldsymbol{u},p)$:

$$\delta_{(\boldsymbol{v},q)}I(\boldsymbol{u},p) = 0 \quad \forall(\boldsymbol{v},q) \tag{8.5}$$

By definition of the functional derivative, we have:

$$\begin{aligned}\delta_{(\boldsymbol{v},q)}I(\boldsymbol{u},p) = \lim_{\epsilon\to 0}\frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big[&\int_\Omega \frac{\mu}{2}\|\boldsymbol{\nabla}\boldsymbol{u}\|^2 - p\nabla\cdot\boldsymbol{u}\,\mathrm{d}\Omega \\ &+\epsilon\int_\Omega \mu\boldsymbol{\nabla}\boldsymbol{u}:\boldsymbol{\nabla}\boldsymbol{v} - q\nabla\cdot\boldsymbol{u} - p\nabla\cdot\boldsymbol{v}\,\mathrm{d}\Omega \\ &+\epsilon^2\int_\Omega \frac{\mu}{2}\|\boldsymbol{\nabla}\boldsymbol{v}\|^2 - q\nabla\cdot\boldsymbol{v}\,\mathrm{d}\Omega\,\Big]\end{aligned} \tag{8.6}$$

Eventually:

$$\delta_{(\boldsymbol{v},q)}I(\boldsymbol{u},p) = \int_\Omega \mu\boldsymbol{\nabla}\boldsymbol{u}:\boldsymbol{\nabla}\boldsymbol{v} - p\nabla\cdot\boldsymbol{v} - q\nabla\cdot\boldsymbol{u}\,\mathrm{d}\Omega = 0 \quad \forall(\boldsymbol{v},q) \tag{8.7}$$

8.1.3 Equivalent partial differential equation

In order to better grasp the meaning of equation (8.7), we can rewrite it into a more or less equivalent partial differential equation (PDE). To achieve this we use integration by parts formulae to rewrite the saddle-point condition in the form:

$$\int_\Omega \mathrm{PDE}_{\boldsymbol{u}} \times \boldsymbol{v} + \mathrm{PDE}_p \times q\,\mathrm{d}\Omega = 0 \quad \forall(\boldsymbol{v},q) \tag{8.8}$$

From which we will infer: $\mathrm{PDE}_{\boldsymbol{u}} = 0$ and $\mathrm{PDE}_p = 0$ by virtue of the localization theorem.

Using integration by parts on the terms $\int_\Omega \mu\boldsymbol{\nabla}\boldsymbol{u}:\boldsymbol{\nabla}\boldsymbol{v}\,\mathrm{d}\Omega$ and $\int_\Omega -p\nabla\cdot\boldsymbol{v}\,\mathrm{d}\Omega$ in equation (8.7), one gets:

$$\begin{aligned}\int_\Omega -\mu\boldsymbol{\Delta}\boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}\Omega + \int_{\delta\Omega}(\mu\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{n})\cdot\boldsymbol{v}\,\mathrm{d}\delta\Omega + \int_\Omega \nabla p\cdot\boldsymbol{v}\,\mathrm{d}\Omega - \int_{\delta\Omega} p\boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}\delta\Omega \\ - \int_\Omega q\nabla\cdot\boldsymbol{u}\,\mathrm{d}\Omega = 0\end{aligned} \tag{8.9}$$

For the sake of simplicity, we suppose that Dirichlet boundary conditions for $\boldsymbol{u}$ are prescribed on the entire boundary $\delta\Omega$. Then, $\boldsymbol{v}$ vanishes on $\delta\Omega$ so that the boundary integrals in the previous expression also vanish.

Other boundary conditions for Stokes' problem are discussed in the dedicated chapter 9. Eventually we obtain the following system of PDE:

$$\left\{ \begin{array}{rcl} -\mu\boldsymbol{\Delta u} + \nabla p & = & 0 \\ \nabla \cdot \boldsymbol{u} & = & 0 \end{array} \right. \tag{8.10}$$

The first line of this system expresses *momentum* conservation whereas the second line expresses *mass* conservation.

8.2 The discrete Stokes problem

8.2.1 Discrete functional spaces

In order to discretize Stokes' problem, we need to choose discrete finite element functional spaces for $\boldsymbol{u}$ and p. We have not discussed yet which functional spaces are suitable for Stokes' problem. However, given the expression of Stokes functional (8.4), it seems natural to seek for $\boldsymbol{u}$ in a discrete subspace of $\left(\mathcal{H}^1_{\mathrm{D}}(\Omega)\right)^3$ (because of the term $\|\boldsymbol{\nabla u}\|^2$) and to seek for p in a discrete subspace of $\mathcal{L}^2(\Omega)$ (because of the term $p\nabla \cdot \boldsymbol{u}$). Then, if $\boldsymbol{N}_i$ are the basis functions for $\boldsymbol{u}$ functional space (its dimension being $n_{\boldsymbol{u}}$) and P_k are the basis functions for p functional space, we write:

$$\boldsymbol{u}_h(\boldsymbol{x}) = \sum_{i=1}^{n_{\boldsymbol{u}}} \boldsymbol{u}_i \boldsymbol{N}_i(\boldsymbol{x}) \tag{8.11}$$

$$p_h(\boldsymbol{x}) = \sum_{k=1}^{n_p} p_k P_k(\boldsymbol{x}) \tag{8.12}$$

8.2.2 Discrete saddle-point condition

The discrete saddle-point condition corresponding to (8.9) is:

$$\delta_{(\boldsymbol{N}_j,P_l)} I(\boldsymbol{u}_h,p_h) = \int_\Omega \mu \boldsymbol{\nabla u}_h : \boldsymbol{\nabla N}_j - p_h \nabla\cdot\boldsymbol{N}_j - P_l \nabla\cdot\boldsymbol{u}_h \,\mathrm{d}\Omega = 0 \quad \forall(\boldsymbol{N}_j,P_l) \tag{8.13}$$

Then, expanding $\boldsymbol{u}_h$ and p_h on the basis of their test functions, we get a linear system for the $\boldsymbol{u}_i$ and p_k degrees of freedom:

$$\begin{aligned} \delta_{(\boldsymbol{N}_j,P_l)} I(\boldsymbol{u}_h,p_h) &= \sum_i \boldsymbol{u}_i \int_\Omega \mu \boldsymbol{\nabla N}_i : \boldsymbol{\nabla N}_j - P_l \nabla\cdot\boldsymbol{N}_i \,\mathrm{d}\Omega \\ &\quad + \sum_k p_k \int_\Omega -P_k \nabla\cdot\boldsymbol{N}_j \,\mathrm{d}\Omega = 0 \quad \forall(\boldsymbol{N}_j,P_l) \end{aligned} \tag{8.14}$$

Defining the matrices $\mathsf{R}_{ji} = \int_\Omega \boldsymbol{\nabla N}_j : \boldsymbol{\nabla N}_i \,\mathrm{d}\Omega$ and $\mathsf{B}_{li} = \int_\Omega -P_l \nabla\cdot\boldsymbol{N}_i \,\mathrm{d}\Omega$, equation (8.14) takes the following form:

$$\mathsf{A}_h = \begin{pmatrix} \mathsf{R} & \mathsf{B}^t \\ \mathsf{B} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_h \\ p_h \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{8.15}$$

A_h is the complete matrix of Stokes' problem, R is the rigidity matrix and B the divergence matrix. Matrix A_h is symmetric: the $(1,2)$-block corresponding to the pressure gradient in the momentum equation is the transpose of the $(2,1)$-block associated to the zero-divergence constraint. The $(2,2)$-block is void. This particular structure of A_h is typical of saddle-point problems: the diagonal block of the Lagrange multiplier unknowns (here p) is void. In the minimization problem case (chapter 2), the matrix of the discrete system was symmetric and positive-definite. All its eigenvalues were positive real. In the saddle-point case, the matrix A_h is still symmetric but it is indefinite with $n_{\boldsymbol{u}}$ positive eigenvalues and n_p negative eigenvalues.

8.2.3 Stability of the discrete problem

In order to apply Lax's theorem (section 3.3), we first have to assure that the continuous Stokes' problem (8.9) is well-posed. Second, we need the discrete problem (8.15) to be stable, that is A_h should be invertible and the eigenvalue of smallest modulus should be uniformly bounded by a constant $\beta > 0$ independent of the discretization parameter h.

The well-posedness of the continuous Stokes' problem is given, for instance, in [EG02]. The demonstration relies on the verification of a so-called inf–sup condition (also known as Ladyzhenskaya–Brezzi–Babuska, or LBB, condition):

$$\inf_{q \in \mathcal{L}^2_0(\Omega)} \sup_{\boldsymbol{v} \in \left(\mathcal{H}^1_0(\Omega)\right)^3} \int_\Omega \frac{q \nabla \cdot \boldsymbol{v}}{\|\boldsymbol{v}\|_{\mathcal{H}^1} \|q\|_{\mathcal{L}^2}} \,\mathrm{d}\Omega \geq \beta \tag{8.16}$$

Similarly, the discrete problem stability involves discrete inf–sup conditions, with $\boldsymbol{u}_h$ and p_h instead of $\boldsymbol{v}$ and q. Practically, this means that the finite element spaces for $\boldsymbol{u}_h$ and p_h cannot be chosen at random: they must be *compatible*.

8.2.4 Compatible finite elements

The simplest finite element

The simplest conforming element that we can consider involves the finite element space of section 3.4:

$$\boldsymbol{u}_h \in W_1 \subset \mathcal{H}^1(\Omega) \tag{8.17}$$

$$p_h \in V_0 \subset \mathcal{L}^2(\Omega) \tag{8.18}$$

This finite element is pictured in figure 8.1. Besides its simplicity, it has an interesting property of *local mass conservation*. Indeed, if we look at the k^{th} line of the divergence matrix B:

$$\int_\Omega \nabla \cdot \boldsymbol{u}_h P_k \,\mathrm{d}\Omega = 0$$

Recalling that P_k is the indicator function of element Ω_k and using an integration by parts formula, we are led to:

$$\int_{\Omega_k} \nabla \cdot \boldsymbol{u}_h \,\mathrm{d}\Omega_k = \int_{\delta\Omega_k} \boldsymbol{u}_h \cdot \boldsymbol{n} \,\mathrm{d}\delta\Omega_k = 0$$

This equality expresses the fact that, if the discrete solution $\boldsymbol{u}_h$ exists, it verifies local mass conservation[1].

[1] We will discuss in greater detail the conservation properties of the finite element method in the dedicated chapter 9.

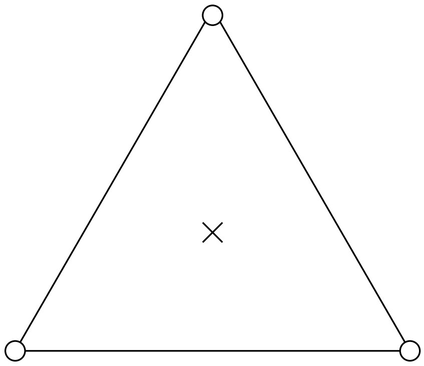


Figure 8.1: The simplest conforming element for Stokes' problem. ○: velocity nodes. ×: pressure nodes.

Unfortunately, this finite element *is not stable*: it leads to a matrix A_h which *is not invertible*. We will give a numerical example of the use of an unstable element in section 8.2.5.

Cast3M finite elements

The main finite elements of Cast3M used in solving (Navier)-Stokes' problems are shown in table 8.1. We advocate the use of the `QUAF/CENTREP1` family of elements. Indeed, these elements are stable, spatially accurate (third-order for velocity and second order for pressure), available for every simple geometric shapes in 2D and 3D and have the same local mass-conservation property as the simple element seen in 8.2.4. However, these nice properties come at a price: these elements are rather costly, which means that computing and solving the linear systems with matrix A_h is memory and CPU-time consuming.

8.2.5 Numerical examples

We consider the following Stokes' problem defined on the square $\Omega = [0,1] \times [0,1]$:

$$\left\{ \begin{array}{rcl} -\mu \boldsymbol{\Delta u} + \nabla p &=& 0 \\ \nabla \cdot \boldsymbol{u} &=& 0 \end{array} \right. \tag{8.19}$$

with the Dirichlet boundary conditions for $\boldsymbol{u}$:

$$\boldsymbol{u}|_{\delta\Omega} = \begin{pmatrix} 4x(1-x)y \\ 0 \end{pmatrix} \tag{8.20}$$

In addition, the pressure is prescribed in a randomly chosen point of the domain. This is due to the fact that, for (Navier)-Stokes' problem in a closed domain, pressure is only defined up to a constant. Then, we need to prescribe a value for this constant in order to have a well-posed problem. We discuss thoroughly the reason for this under-determination of the pressure in section 9.4.4.

The significant part of the data file, in Cast3M's Gibiane language, corresponding to problem (8.19)-(8.20) is given on listing 8.1.

If we use the unstable $\mathbb{Q}_1/\mathbb{P}_0$ element, we obtain the result displayed on figure 8.2. The velocity field seems globally correct despite some small oscillations in the bottom left

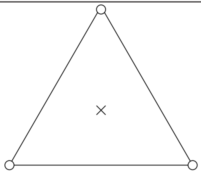
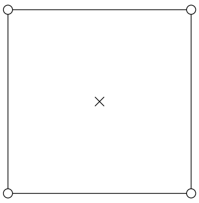
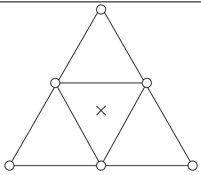
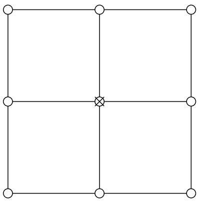
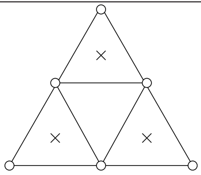
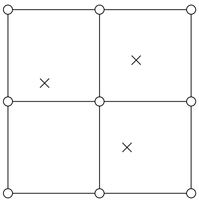
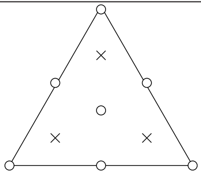
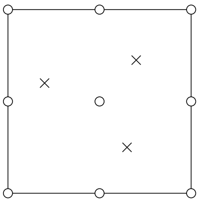
Element	Interpolation $\boldsymbol{u}_h/p_h$	CastɜM name	Stable?
	$\mathbb{P}_1/\mathbb{P}_0$	`LINE/CENTRE`	no
	$\mathbb{Q}_1/\mathbb{P}_0$	`LINE/CENTRE`	no
	$4\mathbb{P}_1/\mathbb{P}_0$	`MACRO/CENTREP0`	yes
	$4\mathbb{Q}_1/\mathbb{P}_0$	`MACRO/CENTREP0`	yes
	$4\mathbb{P}_1/\mathbb{P}_1^{\text{disc}}$	`MACRO/CENTREP1`	no
	$4\mathbb{Q}_1/\mathbb{P}_1^{\text{disc}}$	`MACRO/CENTREP1`	yes
	$\mathbb{P}_2^+/\mathbb{P}_1^{\text{disc}}$	`QUAF/CENTREP1`	yes
	$\mathbb{Q}_2/\mathbb{P}_1^{\text{disc}}$	`QUAF/CENTREP1`	yes

Table 8.1: Main 2D finite elements of CastɜM used in solving (Navier-)Stokes' problems. ○: velocity degrees of freedom ($\boldsymbol{u}_h$). ×: pressure degrees of freedom (p_h).

```
* kvit = ’LINE’ ;   kpre = ’CENTRE’ ;
kvit = ’QUAF’ ;  kpre = ’CENTREP1’ ;
*
$mt = ’MODE’ _mt ’NAVIER_STOKES’ kvit ;
*
mclim = bas ’ET’ dro ’ET’ hau ’ET’ gau ;
xm ym = ’COORDONNEE’ mclim ;
cux   = ’NOMC’ ’UX’ (xm ’*’ (’-’ xm 1.) ’*’ ym ’*’ -4.) ;
mp1 = ’POIN’ (’DOMA’ $mt kpre) 1 ;
*
rv = ’EQEX’ ’NITER’ 1 ’OMEGA’ 1.D0 ’ITMA’ 1
     ’OPTI’ ’EF’ ’IMPL’ ’CENTREE’ kpre
     ’ZONE’ $mt ’OPER’ ’KBBT’ 1. ’INCO’ ’UN’ ’PN’
     ’OPTI’ ’EF’ ’IMPL’ ’CENTREE’
     ’ZONE’ $mt ’OPER’ ’LAPN’ 1. ’INCO’ ’UN’
     ’CLIM’ ’UN’ ’UIMP’ mclim cux
     ’CLIM’ ’UN’ ’VIMP’ mclim 0.
     ’CLIM’ ’PN’ ’TIMP’ mp1 0. ;
*
rv . ’INCO’ = ’TABLE’ ’INCO’ ;
rv . ’INCO’ . ’UN’  = ’KCHT’ $mt ’VECT’ ’SOMMET’ (0. 0.) ;
rv . ’INCO’ . ’PN’  = ’KCHT’ $mt ’SCAL’ kpre 0. ;
*
EXEC rv ;
```

Listing 8.1: CastɜM data file `infsup.dgibi` corresponding to problem (8.19)-(8.20).

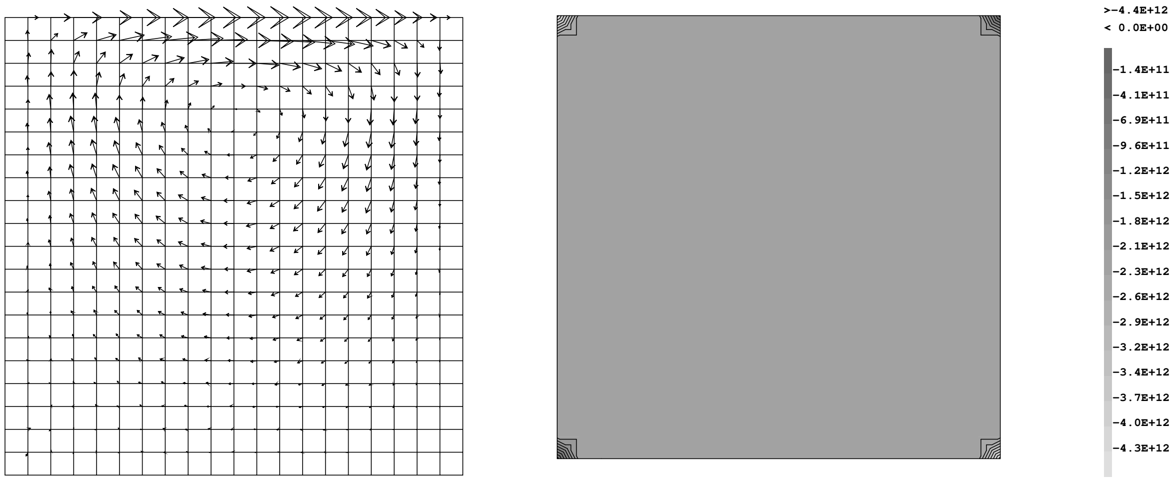


Figure 8.2: Numerical solution of Stokes' problem (8.19)-(8.20) with square `LINE/CENTRE` finite elements. Left: velocity field $\boldsymbol{u}_h$. Right: pressure field p_h.

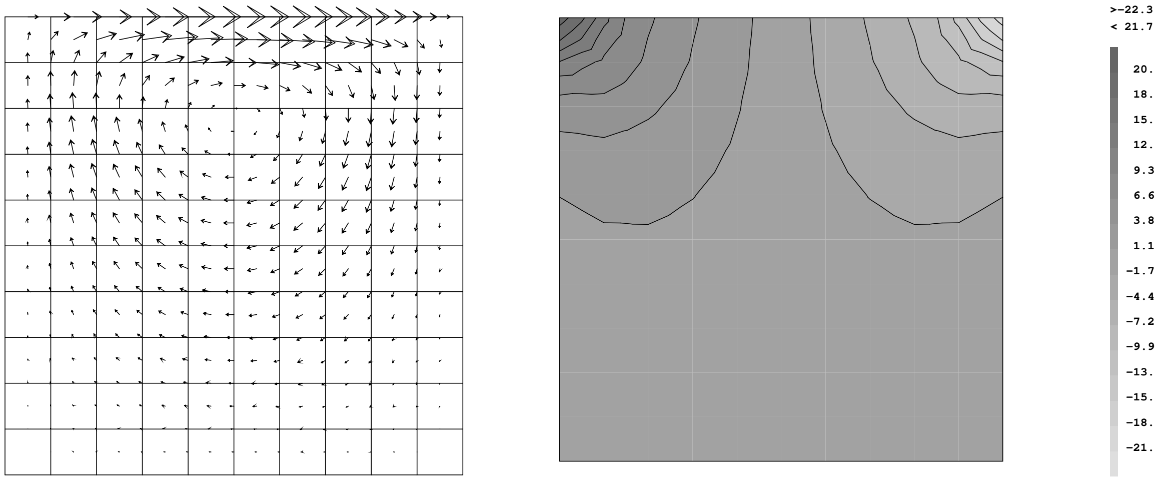


Figure 8.3: Numerical solution of Stokes' problem (8.19)-(8.20) with square `QUAF/CENTREP1` finite elements. Left: velocity field $\boldsymbol{u}_h$. Right: pressure field p_h.

corner. However, the pressure field has an order of magnitude of 10^{12} in the corners, which cannot be correct. These very high values are due to the near-singularity of the matrix A_h build from unstable finite elements. In floating-point arithmetic, a matrix is rarely exactly singular, because of the limited precision of the real numbers' representation. However, as A_h is really close to being singular, solving the linear system leads to some pressure unknowns having a large magnitude.

On using stable $\mathbb{Q}_2/\mathbb{P}_1^{\text{disc}}$ elements, we get the result shown in figure 8.3, which is correct in both the velocity and pressure unknowns.

Exercise 5 *Verify numerically by refining the mesh that the convergence order for the velocity and pressure unknows is correct.*

8.3 Summary

In this chapter, we have shown that Stokes' problem can be expressed as a constrained minimization (saddle-point) problem. The pressure unknown p is then interpreted as the Lagrange multiplier of the mass-conservation constraint $\nabla \cdot \boldsymbol{u} = 0$.

From the matrix point of view, the Stokes' linear system has a particular structure: $\mathsf{A}_h = \begin{pmatrix} \mathsf{R} & \mathsf{B}^t \\ \mathsf{B} & 0 \end{pmatrix}$ which is typical of constrained optimization problems.

The discrete stability condition also constrains the choice of suitable finite elements for the velocity $\boldsymbol{u}_h$ and pressure p_h unknowns.

Chapter 9

Boundary conditions and conservation

$$\begin{aligned}
\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u}) \cdot \boldsymbol{u} &= -\nabla p^* + \nu \boldsymbol{\Delta u} + \boldsymbol{s_u} \\
\nabla \cdot \boldsymbol{u} &= 0 \\
\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T &= \alpha \Delta T + s_T \\
\boxed{\text{Boundary conditions?}} &
\end{aligned}$$

In this chapter we study in greater detail the boundary conditions for the various model problems seen in the previous chapters.

First, in section 9.1, we consider a variation on Dirichlet's problem of chapter 2: Neumann's problem. We focus on an important property of the continuous Neumann problem, which is preserved by the finite-element method: the *global conservation* property. Intertwined with this global conservation property, we exhibit a *compatibility condition* on the prescribed boundary flux.

Then, in section 9.2, we show how the global conservation property can be expressed for a steady diffusion problem when the boundary conditions are more general than in Neumann's problem. We do this by introducing *Lagrange multipliers* for the prescription of Dirichlet boundary conditions.

In section 9.3 we consider a steady convection-diffusion equation. We have the choice to integrate by parts the convective term or to leave it as it is. Depending on this choice, we obtain one of two possible Neumann (natural) boundary conditions.

The following section 9.4 is dedicated to Stokes' problem. We exhibit the essential, natural and mixed boundary conditions for this problem, together with the possibly connected compatibility conditions. The viscous term can be written in different forms. These expressions are equivalent from the PDE point of view. However, they give rise to different natural boundary conditions when we consider the associated weak form.

Eventually, in section 9.5 we perform some computations on a steady Navier-Stokes problem. We illustrate numerically the various natural boundary conditions obtained for different formulations of the convective and viscous terms.

9.1 Neumann's problem with source terms

9.1.1 The continuous Neumann problem

We start from Dirichlet's problem (2.3) and make some changes:

$$\min_{T\in\mathcal{H}^1(\Omega)} I(T) = \min_{T\in\mathcal{H}^1(\Omega)} \int_\Omega \frac{\alpha}{2}\|\nabla T\|^2 - sT \,\mathrm{d}\Omega - \int_{\delta\Omega} qT \,\mathrm{d}\delta\Omega \tag{9.1}$$

This time, $T(\boldsymbol{x}) \in \mathcal{H}^1(\Omega)$ instead of $T(\boldsymbol{x}) \in \mathcal{H}^1_{\mathrm{D}}(\Omega)$ as in (2.3): this means that we have not prescribed any Dirichlet condition for T on the boundary of $\delta\Omega$. s is a function defined on Ω which has the meaning of a *volume* source term: this will be shown explicitly in the corresponding PDE. Similarly, q is a function defined on the boundary $\delta\Omega$ which has the meaning of a *surface* source term.

The minimization condition for I writes:

$$\delta_U I(T) = \int_\Omega \alpha\nabla T\cdot\nabla U \,\mathrm{d}\Omega - \int_\Omega sU \,\mathrm{d}\Omega - \int_{\delta\Omega} qU \,\mathrm{d}\delta\Omega = 0 \quad \forall U \in \mathcal{H}^1(\Omega) \tag{9.2}$$

Assuming that the solution of the continuous problem T is sufficiently regular, we can use an integration by parts on the previous formula. This leads to:

$$\int_\Omega (-\alpha\Delta T - s)\times U \,\mathrm{d}\Omega + \int_{\delta\Omega} (\alpha\nabla T\cdot\boldsymbol{n} - q)\times U \,\mathrm{d}\delta\Omega = 0 \tag{9.3}$$

First, let us choose U in the set of functions defined on Ω with zero value on $\delta\Omega$. With such a U, the second integral in (9.3) is zero which implies that the first integral is also zero. The only way that the first integral be zero for any such U is that:

$$-\alpha\Delta T - s = 0 \quad \text{on} \quad \Omega \tag{9.4}$$

Second, let us now choose U in the set of functions defined on Ω with zero value everywhere *except* on $\delta\Omega$. With such a U, the first integral in (9.3) is zero which implies that the second integral is also zero. The only way that the second integral be zero for any such U is that:

$$\alpha\nabla T\cdot\boldsymbol{n} - q = 0 \quad \text{on} \quad \delta\Omega \tag{9.5}$$

Then q has the meaning of an incoming flux through $\delta\Omega$. Indeed, the thermal flux is defined as: $\boldsymbol{q} = -\alpha\nabla T$. Thus the boundary condition (9.5) is equivalent to: $\boldsymbol{q}\cdot\boldsymbol{n} = -q$. $\boldsymbol{n}$ was conventionally defined as the outgoing normal to domain Ω in the integration by parts formula that we used which means that q is truly an incoming flux.

The boundary condition on the unknown's flux (9.5) is called a *Neumann* condition or *natural* condition. Conversely, the boundary condition on the unknown itself, as described in chapter 2, is called a *Dirichlet* condition or *essential* condition.

9.1.2 The discrete Neumann problem

The discrete Neumann problem writes:

$$\min_{T_h\in\mathtt{H}^1(\Omega)} I(T_h) = \min_{T_h\in\mathtt{H}^1(\Omega)} \int_\Omega \frac{\alpha}{2}\|\nabla T_h\|^2 - sT_h \,\mathrm{d}\Omega - \int_{\delta\Omega} qT_h \,\mathrm{d}\delta\Omega \tag{9.6}$$

The minimization condition for the discrete problem is:

$$\delta_{N_i} I(T_h) = \int_\Omega \alpha \nabla T_h \cdot \nabla N_i - s N_i \, \mathrm{d}\Omega - \int_{\delta\Omega} q N_i \, \mathrm{d}\delta\Omega = 0 \quad \forall N_i \in \mathtt{H}^1(\Omega) \quad \text{i.e.} \quad \forall i \in \Omega \tag{9.7}$$

Notice that, contrary to the continuous case, it is not possible to integrate (9.7) by parts because T_h is not regular enough. Indeed, the gradient of T_h is in general discontinuous at the interface between two elements (see figure 3.3 for example). Therefore it would not be possible to give a meaning to the integral $\int_\Omega (-\alpha \Delta T_h) \times N_i \, \mathrm{d}\Omega$.

However, the second integral in (9.3) can be defined, even in the discrete case, since $\int_{\delta\Omega} (\alpha \nabla T_h \cdot \boldsymbol{n} - q) \times N_i \, \mathrm{d}\delta\Omega = 0$ is computable. But, in fact, this does not allow us to conclude that $\alpha \nabla T_h \cdot \boldsymbol{n} - q = 0$ on $\delta\Omega$ because N_i, for i being a boundary node, is not zero on $\Omega \setminus \delta\Omega$: it is a hat function. Let us repeat it, in general:

$$\alpha \nabla T_h \cdot \boldsymbol{n} - q \neq 0 \quad \text{on} \quad \delta\Omega \tag{9.8}$$

Now, can we still say that q is an incoming flux once the problem has been discretized? Hopefully, the answer is yes. The fact that we cannot use integration by parts in the discrete problem does not prevent, provided that the discretized problem is stable, the convergence of T_h to the solution T of the continuous problem (9.2). And, if T is sufficiently regular for applying integration by parts, q still has the meaning of an incoming flux, at convergence.

One says that the Neumann condition $\alpha \nabla T \cdot \boldsymbol{n} - q = 0$ is prescribed *weakly* when formulation (9.7) is used. On the other hand, the Dirichlet condition $T = T_0$ is said to be prescribed *strongly* in the discrete formulation (2.19): $T = T_0$ holds exactly at the boundary nodes located on $\delta\Omega$.

In section 9.2.2, we try to make our point on weak boundary conditions clearer by considering a simple example. For a more in-depth analysis, the reader can refer to the classic textbook by Strang and Fix [SF88].

Eventually, expanding T_h on the N_i basis, we obtain the linear system with the unknowns T_j and a right-hand side b:

$$\begin{aligned}\int_\Omega \alpha \nabla T_h \cdot \nabla N_i - s N_i \, \mathrm{d}\Omega - \int_{\delta\Omega} q N_i \, \mathrm{d}\delta\Omega &= \sum_j T_j \Big(\int_\Omega \nabla N_j \cdot \nabla N_i \, \mathrm{d}\Omega \Big) - b_i(s, q) \\ &= \sum_j T_j \mathsf{R}_{ji} - b_i(s, q) \\ &= 0 \quad \forall i \end{aligned} \tag{9.9}$$

In matrix form:

$$\mathsf{R} \boldsymbol{T} - \boldsymbol{s} - \bar{\boldsymbol{q}} = \boldsymbol{0} \tag{9.10}$$

where R is the rigidity matrix $\mathsf{R}_{ji} = \int_\Omega \nabla N_j \cdot \nabla N_i \, \mathrm{d}\Omega$, $\boldsymbol{s}$ is the contribution of the volume source term to the right-hand side $\boldsymbol{s}_i = \int_\Omega s N_i \, \mathrm{d}\Omega$ and $\bar{\boldsymbol{q}}$ is the contribution of the surface source term (incoming flux) to the right-hand side $\bar{\boldsymbol{q}}_i = \int_{\delta\Omega} q N_i \, \mathrm{d}\delta\Omega$.

9.1.3 Indeterminacy of the unknown and compatibility condition

Let us go back to problem (9.1). This problem is called *Neumann's problem* underlining the fact that the unknown T is not imposed anywhere on the boundary, contrary to Dirichlet's problem. Since T is not imposed on the boundary and since only ∇T is present in the functional, we deduce that, if T is a solution of the problem, then $T+c$, with c an arbitrary constant, is also a valid solution. We have an *indeterminacy* in T: T is only known up to a constant.

Together with and dual to this indeterminacy, there exists a *compatibility condition* on the source terms that needs to be verified. To exhibit this compatibility condition, we write the minimization condition (9.2) for a particular variation U, constant (but not zero) on Ω. This gives:

$$\int_{\Omega} s \,\mathrm{d}\Omega + \int_{\delta\Omega} q \,\mathrm{d}\delta\Omega = 0 \tag{9.11}$$

This equation is a *global conservation* statement: the amount of heat leaving the domain through the boundary $\delta\Omega$ (outgoing flux) is equal to the amount of heat produced in the domain Ω.

In the case of the discrete problem, the compatibility condition still holds because the constant functions on Ω are always part of the finite element space used to approximate T_h. Let us prove it: the sum of all the basis functions has value 1 on every node P_j due to the nodal basis property (section 3.4).

$$\left(\sum_i N_i(\boldsymbol{x})\right)\Bigg|_{\boldsymbol{x}=P_j} = 1 \quad \forall j \tag{9.12}$$

As the basis functions N_i are polynomials on every element, this gives:

$$\left(\sum_i N_i(\boldsymbol{x})\right) = 1 \quad \forall \boldsymbol{x} \in \Omega \tag{9.13}$$

Summing up on i the minimization conditions of the discrete problem (9.7), we get exactly the compatibility condition (9.11):

$$\begin{aligned}
&\sum_i \left(\int_{\Omega} \alpha \nabla T_h \cdot \nabla N_i - s N_i \,\mathrm{d}\Omega - \int_{\delta\Omega} q N_i \,\mathrm{d}\delta\Omega\right) \\
&= \int_{\Omega} \alpha \nabla T_h \cdot \nabla \left(\sum_i N_i\right) - s\left(\sum_i N_i\right) \mathrm{d}\Omega - \int_{\delta\Omega} q \left(\sum_i N_i\right) \mathrm{d}\delta\Omega \\
&= -\left(\int_{\Omega} s \,\mathrm{d}\Omega + \int_{\delta\Omega} q \,\mathrm{d}\delta\Omega\right) \\
&= 0
\end{aligned} \tag{9.14}$$

In matrix form, we have:

$$\sum_i \boldsymbol{s}_i + \sum_i \bar{\boldsymbol{q}}_i = 0 \tag{9.15}$$

Remark 3 *An important consequence is that the computed discrete solution T_h satisfies the same* global heat conservation *property as in the continuous problem. This is an important feature of the finite element method.*

Remark 4 *On the contrary, the discrete solution T_h does not satisfy a* local heat conservation *property on an element Ω_k. The reason for this is that the indicator function of an element $\mathbf{1}_k$ (with value 1 on Ω_k and 0 elsewhere) is not part of the functional space of T_h,* $\mathtt{H}^1(\Omega)$. *However, the i^{th} equation of the discrete system ($\sum_j T_j \mathsf{R}_{ji} - b_i = 0$ from equation (9.9)) can be interpreted as a local heat conservation statement <u>around</u> node i.*

From the point of view of the linear system (9.9), on the one hand, the indeterminacy of the unknown is characterized by the fact that the rigidity matrix $\mathsf{R}_{ij} = \int_\Omega \nabla N_i \cdot \nabla N_j \,\mathrm{d}\Omega$ has a zero eigenvalue associated with the eigenvector $\mathbf{1}$ of all ones: $\mathsf{R}\mathbf{1} = 0 \cdot \mathbf{1}$. This also means that the sum of all the terms in a given row of the matrix R is zero. On the other hand, the compatibility condition requires that the right-hand side b be orthogonal to $\mathbf{1}$ i.e.: $\sum_i b_i = 0$ (9.15). Otherwise, the linear system has no solution.

Eventually, once the compatibility condition has been checked, we can get rid of the indeterminacy of the unknown, by prescribing the value of T_h at a randomly-chosen node for instance, so that the linear system admits a unique solution.

9.2 A general diffusion problem

9.2.1 Varying the boundary conditions

In general, we want to solve diffusion problems with conditions of Dirichlet type on one part of the boundary and of Neumann type on the other part of the boundary. Thus, we build a partition of $\delta\Omega$: $\delta\Omega = \delta\Omega_\mathrm{D} \cup \delta\Omega_\mathrm{N}$ with $\delta\Omega_\mathrm{D} \cap \delta\Omega_\mathrm{N} = \emptyset$.

At the discrete level, the problem writes:

$$\begin{aligned} \delta_{N_i} I(T_h) = \int_\Omega \alpha \nabla T_h \cdot \nabla N_i - s N_i \,\mathrm{d}\Omega - \int_{\delta\Omega_\mathrm{N}} q N_i \,\mathrm{d}\delta\Omega_\mathrm{N} = 0 \quad \forall N_i \in \mathtt{H}^1_\mathrm{D}(\Omega) \\ \text{i.e.} \quad \forall i \in (\Omega - \delta\Omega_\mathrm{D}) \\ T_i = T_0(P_i) \quad \forall i \in \delta\Omega_\mathrm{D} \end{aligned} \tag{9.16}$$

In the Cast3M code, the `'CLIM'` keyword of the `'EQEX'` operator defines the Dirichlet boundary conditions. They will be used by the linear system solver called by the `EXEC` procedure. The surface (resp. volume) source terms $\int_{\delta\Omega_\mathrm{N}} q N_i \,\mathrm{d}\delta\Omega_\mathrm{N}$ (resp. $\int_\Omega s N_i \,\mathrm{d}\Omega$) are computed by the `'FIMP'` operator (Flux IMPosé in french, i.e. prescribed flux).

Now, what happens to the global conservation property of the finite-element method when the boundary conditions are not all Neumann? As soon as we introduce Dirichlet boundary conditions, we do not have $\sum_i N_i = 1$ on the domain anymore, because we require that the variation be zero on the Dirichlet part of the boundary: $\left(\sum_i N_i\right)(P_j) = 0 \quad \forall i \in \delta\Omega_\mathrm{D}$. Happily, we can still find a way to express a global conservation property. The most elegant way is to release the constraint on the variation, we do not require it to be zero on the Dirichlet boundary anymore, and to introduce *Lagrange multipliers* for the prescription of the Dirichlet boundary conditions.

We are already familiar with the concept of Lagrange multiplier. In the chapter dedicated to Stokes' problem (chapter 8), the multiplier p was used to prescribe the constraint $\nabla \cdot \boldsymbol{u} = 0$ and it had the physical meaning of a pressure.

Here, the Lagrange multipliers λ will be used to prescribe the Dirichlet constraints $T_i = T_0(P_i)$ and will have the physical meaning of a *flux*. This *duality* between essential conditions (here, prescribed values) and natural conditions (here, flux) is a very important concept. To illustrate this concept, we examine in some detail a simple and concrete example. Then, we will generalize in a more abstract way the Lagrange multiplier method for the prescription of Dirichlet boundary conditions.

9.2.2 A simple example

In this subsection, we investigate a simple 1D steady diffusion problem with a constant source term, a Dirichlet boundary condition on the left-hand side and a Neumann boundary condition on the right-hand side. The exact analytical solution to this problem is known. We discretize this problem with the finite element method on a regular mesh composed of two elements and discuss various ways to prescribe the Dirichlet boundary condition.

The continuous problem

The mathematical expression of the problem at hand is:

$$\begin{cases} -\alpha \Delta T = s \\ T|_{x=0} = T_{\text{imp}} \\ -\alpha \left.\dfrac{\partial T}{\partial x}\right|_{x=1} = f_{\text{imp}} \end{cases} \tag{9.17}$$

where the source term s and the coefficient α are constant in space.

Exact solution The problem (9.17) has a parabolic solution:

$$\begin{cases} T(x) = ax^2 + bx + c \\ a = -\dfrac{s}{2\alpha} \\ b = \dfrac{s - f_{\text{imp}}}{\alpha} \\ c = T_{\text{imp}} \end{cases} \tag{9.18}$$

The discrete problem

Discretizing problem (9.17) on a regular two-element mesh leads to:

$$\begin{cases} \displaystyle\int_\Omega \alpha \nabla T_h \nabla N_i \,\mathrm{d}\Omega = \int_{\delta\Omega} -f_{\text{imp}} N_i \,\mathrm{d}\delta\Omega + \int_\Omega s N_i \,\mathrm{d}\Omega \quad \forall i \\ T_h|_{x=0} = T_{\text{imp}} \end{cases} \tag{9.19}$$

where the unknown T_h is thought as:

$$T_h(x) = T_0 N_0(x) + T_1 N_1(x) + T_2 N_2(x) \tag{9.20}$$

Now, we are seeking the value of the unknown vector $\begin{pmatrix} T_0 \\ T_1 \\ T_2 \end{pmatrix}$.

Computing the integrals leads to the matrix formulation of problem (9.19):

$$\frac{\alpha}{\Delta x} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} T_0 \\ T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -f_{\text{imp}} \end{pmatrix} + s\Delta x \begin{pmatrix} \frac{1}{2} \\ 1 \\ \frac{1}{2} \end{pmatrix} \tag{9.21}$$

Here, the Dirichlet condition $T_0 = T_{\text{imp}}$ is not prescribed yet. We can check that the matrix without the Dirichlet condition has a zero eigenvalue, since for every row the sum of the terms is zero.

Looking at the last row of the system:

$$f_{\text{imp}} = -\alpha \frac{T_2 - T_1}{\Delta x} + s\frac{\Delta x}{2} \tag{9.22}$$

Also, we have that:

$$-\alpha \frac{\partial T_h}{\partial x}\bigg|_{x=1} = -\alpha \frac{T_2 - T_1}{\Delta x} \tag{9.23}$$

In agreement with the discussion of section 9.1.2, we see that, in the discrete setting:

$$-\alpha \frac{\partial T_h}{\partial x}\bigg|_{x=1} \neq f_{\text{imp}} \tag{9.24}$$

This is due to the fact that the Neumann condition is prescribed weakly in the finite-element method. However, we notice that if $\Delta x \to 0$, then $-\alpha \frac{\partial T_h}{\partial x}\big|_{x=1} \to f_{\text{imp}}$.

Prescription of the boundary condition

Let us discuss now some methods for prescribing the Dirichlet condition $T_0 = T_{\text{imp}}$.

Incomplete computation of the integrals In fact, the first method is the one that we have implicitly used up to now: we do not compute the integrals involving the test functions N_i associated to a prescribed T_i in the formulation (9.19) because the variation is constrained to be zero at these nodes. The corresponding matrix system is:

$$\frac{\alpha}{\Delta x}\begin{pmatrix} -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} T_0 \\ T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -f_{\text{imp}} \end{pmatrix} + s\Delta x \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} \tag{9.25}$$

This matrix is rectangular. Using the Dirichlet condition $T_0 = T_{\text{imp}}$, we obtain the following square system on the unprescribed unknowns:

$$\frac{\alpha}{\Delta x}\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \frac{\alpha}{\Delta x}\begin{pmatrix} T_{\text{imp}} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -f_{\text{imp}} \end{pmatrix} + s\Delta x \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} \tag{9.26}$$

The solution to this system, with $\Delta x = 0.5$, is:

$$\begin{cases} T_1 = T_{\text{imp}} - \dfrac{f_{\text{imp}}}{2\alpha} + \dfrac{3s}{8\alpha} \\ T_2 = T_{\text{imp}} - \dfrac{f_{\text{imp}}}{\alpha} + \dfrac{s}{2\alpha} \end{cases} \tag{9.27}$$

A remarkable property is that, in the particular case of problem (9.17), *the discrete solution coincides with the exact solution at the mesh nodes*. Indeed: $T_1 = T|_{x=0.5}$ and $T_2 = T|_{x=1}$.

The method of incomplete computation of the integrals, however, is not very convenient to use in a computer code because every discretization operator must be aware of the Dirichlet boundary conditions.

The elimination method The second method is very similar to the first method but its implementation is different: we compute all the integrals, including the ones involving the test functions N_i associated with a prescribed T_i. Once all the contributions have been computed, we have the initial system given by equation (9.21). Then, we modify the matrix rows and the right-hand side corresponding to the Dirichlet boundary conditions:

$$\frac{\alpha}{\Delta x}\begin{pmatrix} 1 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} T_0 \\ T_1 \\ T_2 \end{pmatrix} = \frac{\alpha}{\Delta x}\begin{pmatrix} T_{\text{imp}} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -f_{\text{imp}} \end{pmatrix} + s\Delta x\begin{pmatrix} 0 \\ 1 \\ \frac{1}{2} \end{pmatrix} \tag{9.28}$$

Next, we symmetrize the matrix using the known values of the Dirichlet conditions:

$$\frac{\alpha}{\Delta x}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} T_0 \\ T_1 \\ T_2 \end{pmatrix} = \frac{\alpha}{\Delta x}\begin{pmatrix} T_{\text{imp}} \\ T_{\text{imp}} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -f_{\text{imp}} \end{pmatrix} + s\Delta x\begin{pmatrix} 0 \\ 1 \\ \frac{1}{2} \end{pmatrix} \tag{9.29}$$

This method is almost similar to the first one and gives exactly the same solution. However, it is somewhat a pity to compute all the terms in the first row and then discard them. In fact, they can provide us with some useful piece of information.

The method of Lagrange multipliers In this third method the Dirichlet condition $T_0 = T_{\text{imp}}$ is prescribed without modifying the first row of the system, by introducing a supplementary unknown μ, called a *Lagrange multiplier*. On the row corresponding to μ, we write the equation $T_0 = T_{\text{imp}}$. The column acting on μ is taken as the transpose of the row[1] so that we get the following augmented symmetric system with four unknowns $\begin{pmatrix} T_0 \\ T_1 \\ T_2 \\ \mu \end{pmatrix}$:

$$\begin{pmatrix} 1\frac{\alpha}{\Delta x} & -1\frac{\alpha}{\Delta x} & 0 & 1 \\ -1\frac{\alpha}{\Delta x} & 2\frac{\alpha}{\Delta x} & -1\frac{\alpha}{\Delta x} & 0 \\ 0 & -1\frac{\alpha}{\Delta x} & 1\frac{\alpha}{\Delta x} & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} T_0 \\ T_1 \\ T_2 \\ \mu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ T_{\text{imp}} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -f_{\text{imp}} \\ 0 \end{pmatrix} + s\Delta x\begin{pmatrix} \frac{1}{2} \\ 1 \\ \frac{1}{2} \\ 0 \end{pmatrix} \tag{9.30}$$

This method gives *exactly* the same solution as the two previous methods for the unknowns $\begin{pmatrix} T_0 \\ T_1 \\ T_2 \end{pmatrix}$. In fact, it is algebraically equivalent: the fourth row of the system (9.30) gives $T_0 = T_{\text{imp}}$, then T_1, T_2 are computed with row 2 and 3 which have not been modified from the two previous methods. Finally, μ is computed from the first row, knowing the T_i:

$$\mu = -\alpha\frac{T_0 - T_1}{\Delta x} + s\frac{\Delta x}{2} \tag{9.31}$$

We remark that this equation is very similar to the equation (9.22) involving the weakly prescribed flux f_{imp} at the right boundary. This leads to the physical interpretation of the Lagrange multiplier μ: *μ is the flux which, if prescribed in order to replace the Dirichlet condition $T_0 = T_{imp}$, would lead to the same exact solution T_h.*[2]

[1] The reason for taking the transpose of the line is due to the structure of the underlying constrained optimization problem. The situation is similar to the one presented in section 8.2.

[2] More precisely, here T_h would be identical up to a constant because the boundary conditions would be Neumann everywhere at the boundary.

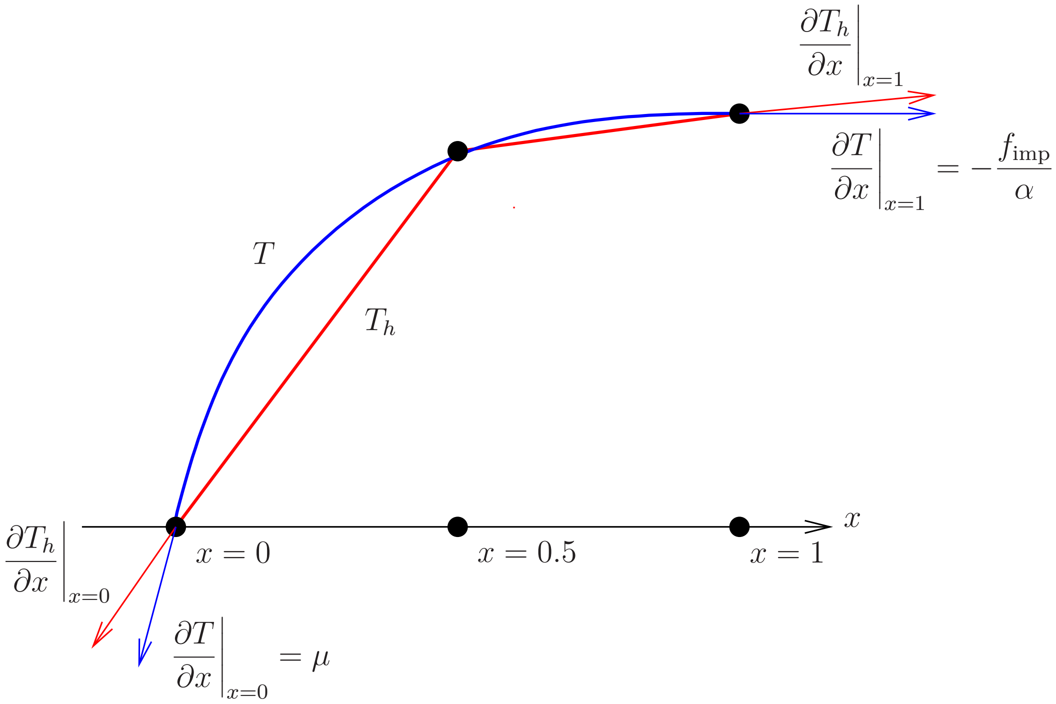

$\left.\frac{\partial T_h}{\partial x}\right|_{x=1}$
$\left.\frac{\partial T}{\partial x}\right|_{x=1} = -\frac{f_{\text{imp}}}{\alpha}$
T
T_h
$\left.\frac{\partial T_h}{\partial x}\right|_{x=0}$
$x = 0$
$x = 0.5$
$x = 1$
x
$\left.\frac{\partial T}{\partial x}\right|_{x=0} = \mu$

Figure 9.1: 1D diffusion problem on a two-element mesh (9.17).

The μ flux is also a weakly-prescribed flux because $\mu \neq -\alpha \left.\frac{\partial T_h}{\partial x}\right|_{x=0}$. In our particular case, another remarkable property is that μ is equal to the exact solution flux! Indeed: $\mu = -\alpha \left.\frac{\partial T}{\partial x}\right|_{x=0}$.

Summing up the first three rows of system (9.30), we recover a *global heat conservation* statement:

$$\int_\Omega s \,\mathrm{d}\Omega = s = f_{\text{imp}} + \mu \tag{9.32}$$

Now we understand the importance of the Lagrange multiplier method. Algebraically, it is equivalent to the other methods and gives the same solution for T_h. However, it gives us a supplementary piece of information: the weak flux value μ on the part of the boundary where the temperature is prescribed. This weak flux μ is very important because it allows us to recover an expression for the global conservation property of the finite-element method when Dirichlet boundary conditions are prescribed.

Some features of the discrete solution Some features of the discrete solution of problem 9.1 are shown on figure 9.1. The discrete solution obtained with the finite element method is nodally exact. The weak fluxes are also exact, contrary to the strong fluxes.

Exercise 6 *Check that the exactness properties of the finite-element method applied to problem* (9.17) *still holds when the mesh has more than two elements and when the mesh is not regular.*

Exercise 7 *Plot the value of the functional* $\int_\Omega \frac{\alpha}{2}\|\nabla T_h\|^2 - sT_h \,\mathrm{d}\Omega$ *as a function of the* position *of the middle node for a two-element mesh. What is the optimal mesh, which minimizes this value, in our case?*

The properties obtained with the finite-element mehthod in the particular case of problem (9.17) are quite remarkable and more difficult to obtain simultaneously with other discretization methods:

- in the *Finite-Differences* method, the Neumann condition is harder to prescribe. Generally, it is prescribed strongly (that is with the formula: $-\alpha\frac{T_2-T_1}{\Delta x} = f_{\mathrm{imp}}$) but the drawback is that the convergence order near the boundary is lowered. Indeed, this finite difference formula is of order 1 whereas the order is generally 2 for the finite-difference Laplacian in the domain interior;

- in the *Finite-Volume* method centered on the elements, we have a local heat conservation property on the elements. However, prescribing Dirichlet boundary conditions is harder because the unknowns are not naturally defined at the boundary.

Also, the Lagrange multiplier method and the global conservation property can be generalized to higher dimensional problems and to other conservation equations, as will be shown next.

9.2.3 The method of Lagrange multipliers for Dirichlet conditions

The continuous Lagrange multiplier method

Now, we reformulate the diffusion problem (9.16) with source terms, Dirichlet conditions on $\delta\Omega_{\mathrm{D}}$ and Neumann conditions on $\delta\Omega_{\mathrm{N}}$ as a *constrained-minimization* (or *saddle-point*) problem for the functional:

$$I(T,\lambda) = \int_\Omega \frac{\alpha}{2}\|\nabla T\|^2 - sT\,\mathrm{d}\Omega + \int_{\delta\Omega_{\mathrm{N}}} f_{\mathrm{imp}} T\,\mathrm{d}\delta\Omega_{\mathrm{N}} + \int_{\delta\Omega_{\mathrm{D}}} \lambda(T-T_0)\,\mathrm{d}\delta\Omega_{\mathrm{D}} \tag{9.33}$$

We have introduced the Lagrange multiplier λ, defined on $\delta\Omega_{\mathrm{D}}$, in order to prescribe the Dirichlet boundary conditions $T = T_0$.

The condition for (T,λ) to be a saddle-point of I writes:

$$\begin{aligned}\int_\Omega \alpha\nabla T\cdot\nabla U - sU\,\mathrm{d}\Omega + \int_{\delta\Omega_{\mathrm{N}}} f_{\mathrm{imp}} U\,\mathrm{d}\delta\Omega_{\mathrm{N}}&\\ + \int_{\delta\Omega_{\mathrm{D}}} \lambda U\,\mathrm{d}\delta\Omega_{\mathrm{D}} + \int_{\delta\Omega_{\mathrm{D}}} \mu(T-T_0)\,\mathrm{d}\delta\Omega_{\mathrm{D}} = 0 \quad \forall(U,\mu)&\end{aligned} \tag{9.34}$$

Under some regularity conditions on T, we can use an integration by parts formula which leads to:

$$\begin{aligned}\int_\Omega (-\alpha\Delta T - s)\,U\,\mathrm{d}\Omega + \int_{\delta\Omega} \alpha\nabla T\cdot\boldsymbol{n}U\,\mathrm{d}\delta\Omega + \int_{\delta\Omega_{\mathrm{N}}} f_{\mathrm{imp}} U\,\mathrm{d}\delta\Omega_{\mathrm{N}}&\\ + \int_{\delta\Omega_{\mathrm{D}}} \lambda U\,\mathrm{d}\delta\Omega_{\mathrm{D}} + \int_{\delta\Omega_{\mathrm{D}}} \mu(T-T_0)\,\mathrm{d}\delta\Omega_{\mathrm{D}} = 0 \quad \forall(U,\mu)&\end{aligned} \tag{9.35}$$

From which we infer the following PDE and boundary conditions:

$$\begin{cases} -\alpha\Delta T = s \\ -\alpha\nabla T\cdot\boldsymbol{n}|_{\delta\Omega_{\mathrm{N}}} = f_{\mathrm{imp}} \\ T|_{\delta\Omega_{\mathrm{D}}} = T_0 \end{cases} \tag{9.36}$$

We get as well the physical meaning of the Lagrange multiplier λ:

$$\lambda = -\alpha\nabla T\cdot\boldsymbol{n}|_{\delta\Omega_{\mathrm{D}}} \tag{9.37}$$

It has the meaning of an *outgoing flux* on the part of the domain boundary $\delta\Omega_{\mathrm{D}}$ where Dirichlet conditions are prescribed. This is an expression of the *duality* between the two types of boundary conditions: where T is imposed, we compute the flux λ; where f_{imp} is imposed, T is computed.

The *global conservation property* is obtained by substitution of the particular variation $(U, \mu) = (1, 0)$ into the equation (9.34):

$$\int_\Omega s \,\mathrm{d}\Omega = \int_{\delta\Omega_{\mathrm{N}}} f_{\mathrm{imp}} \,\mathrm{d}\delta\Omega_{\mathrm{N}} + \int_{\delta\Omega_{\mathrm{D}}} \lambda \,\mathrm{d}\delta\Omega_{\mathrm{D}}$$

However, the continuous Lagrange multiplier method is not without drawbacks:

- the choice of a functional space for λ is not trivial. Looking at the functional, we can choose a functional space with lesser regularity than T (there are no spatial derivatives of λ) and defined on $\delta\Omega_{\mathrm{D}}$ only;
- this choice of a functional space will have to make the continuous problem well-posed;
- we will need a finite-element subspace of this functional space and prove that the resulting discrete problem is stable.

In fact, defining spaces that verify all these conditions is difficult, especially when the boundary $\delta\Omega$ is not smooth. That is why we will proceed in a different way.

The discrete Lagrange multiplier method

In the 1D simple diffusion problem of section (9.17), the Lagrange multiplier method was applied *after the spatial discretization* of the problem. The Lagrange multiplier method was shown to be *algebraically* equivalent to the classical elimination method for prescribing the Dirichlet boundary conditions.

We do the same for the general diffusion problem (9.16) that we formulate as a *discrete saddle-point* problem for the functional:

$$I(T_h, \lambda_h) = \int_\Omega \frac{\alpha}{2}\|\nabla T_h\|^2 - sT_h \,\mathrm{d}\Omega + \int_{\delta\Omega_{\mathrm{N}}} f_{\mathrm{imp}} T \,\mathrm{d}\delta\Omega_{\mathrm{N}} + \sum_{i\in\delta\Omega_{\mathrm{D}}} \lambda_i (T_i - T_{0i}) \tag{9.38}$$

where λ_h are the discrete Lagrange multipliers used for imposing the Dirichlet conditions $T|_{P_i} = T_0|_{P_i}$ on the nodes P_i located on the boundary $\delta\Omega_{\mathrm{D}}$.

The saddle-point condition writes:

$$\begin{aligned}\int_\Omega \alpha\nabla T_h \cdot \nabla N_i - sN_i \,\mathrm{d}\Omega &+ \int_{\delta\Omega_{\mathrm{N}}} f_{\mathrm{imp}} N_i \,\mathrm{d}\delta\Omega_{\mathrm{N}} \\ &+ [\lambda_i + \mu_i (T_i - T_{0i})]|_{i\in\delta\Omega_{\mathrm{D}}} = 0 \quad \forall (i, \mu_i)\end{aligned} \tag{9.39}$$

We also get the following *global conservation statement* by summing equations (9.39) on index i, using $\sum_i N_i = 1$ and choosing $\mu_i = 0$:

$$\int_{\delta\Omega} f_{\mathrm{imp}} \,\mathrm{d}\delta\Omega + \sum_{i\in\delta\Omega_{\mathrm{D}}} \lambda_i = \int_\Omega s \,\mathrm{d}\Omega$$

Notice that the physical meaning of the Lagrange multiplier λ_i has slightly changed with respect to the continuous method: it is now an *integrated flux*.

This is due to the fact (cf. section 3.2) that we use the *weighted residual* method to solve the PDE on $\delta\Omega$ and to prescribe the fluxes on $\delta\Omega_{\mathrm{N}}$, but we apply the *collocation* method to prescribe the Dirichlet condition $T_h - T_0 = 0$ on the corresponding discretization nodes.

It can be shown that, similarly to the simple example of section 9.2.2, the discrete Lagrange multiplier method is algebraically equivalent to the elimination method for prescribing the Dirichlet conditions: the discrete solution T_h is exactly the same in the two methods. In particular, the *stability* of the discrete problem regarding the unknown T_h is also the same in the two methods[3].

Going back to Neumann's problem

Let us go back to Neumann's problem in variational form (9.2):

$$\int_{\Omega} \alpha \nabla T \cdot \nabla U - sU \,\mathrm{d}\Omega - \int_{\delta\Omega} qU \,\mathrm{d}\delta\Omega = 0 \quad \forall U \in \mathcal{H}^1(\Omega)$$

For this problem, we have already mentioned that T was only known up to a constant and that the source terms should satisfy the compatibility condition (9.11):

$$\int_{\Omega} s \,\mathrm{d}\Omega + \int_{\delta\Omega} q \,\mathrm{d}\delta\Omega = 0$$

What if the given functions s and q do not satisfy the compatibility condition (9.11)? The Lagrange multiplier method can help us to understand what will happen in this case. For instance, we can get rid of the indeterminacy of T by prescribing a freely-chosen value T_0 at a chosen node P_0 with a Lagrange multiplier λ. Problem (9.2) then writes:

$$\int_{\Omega} \alpha \nabla T \cdot \nabla U - sU \,\mathrm{d}\Omega - \int_{\delta\Omega} qU \,\mathrm{d}\delta\Omega + \lambda U|_{P_0} + \mu \left(T|_{P_0} - T_0 \right) = 0 \quad \forall (U, \mu) \in \mathcal{H}^1(\Omega) \times \mathbb{R}$$

A global heat conservation statement is obtained by choosing the particular variation $(U, \mu) = (1, 0)$ in the previous equation. This leads to:

$$\int_{\Omega} s \,\mathrm{d}\Omega + \int_{\delta\Omega} q \,\mathrm{d}\delta\Omega = \lambda$$

λ has the physical meaning of an integrated source term applied at the node P_0. If s and q satisfy the compatibility condition (9.11), then we have $\lambda = 0$. Otherwise, $\lambda = (\int_{\Omega} s \,\mathrm{d}\Omega + \int_{\delta\Omega} q \,\mathrm{d}\delta\Omega)$ which counterbalances the gap to the compatibility condition. Thus, numerically, in all cases, we will find a solution T but it will not behave as expected around node P_0 if the localized source term λ is non-zero.

As a side note, we mention that prescribing the temperature at a node is not the only way to get rid of the indeterminacy of T. For example, we could alternatively prescribe the mean value of T on Ω. The interested reader can refer to the review article [BL05] which discusses the best ways to solve the Neumann problem from a numerical point of view.

Simultaneous application of essential and natural conditions

Let us now discuss what happens if one does something wrong with the boundary conditions: for instance, if one prescribes simultaneously Dirichlet and Neumann boundary conditions on $\delta\Omega$.

[3] However, this says nothing about the convergence of the λ_i unknowns...

Applying the continuous Lagrange multiplier method (for the sake of simplicity in writing the formulae), we are led to the functional:

$$I(T,\lambda) = \int_\Omega \frac{\alpha}{2}\|\nabla T\|^2 - sT\,\mathrm{d}\Omega + \int_{\delta\Omega} f_{\text{imp}} T\,\mathrm{d}\delta\Omega + \int_{\delta\Omega} \lambda(T - T_0)\,\mathrm{d}\delta\Omega \tag{9.40}$$

The condition for (T,λ) to be a saddle-point of I writes:

$$\begin{aligned}\int_\Omega \alpha\nabla T\cdot\nabla U - sU\,\mathrm{d}\Omega + \int_{\delta\Omega} f_{\text{imp}} U\,\mathrm{d}\delta\Omega \\ + \int_{\delta\Omega} \lambda U\,\mathrm{d}\delta\Omega + \int_{\delta\Omega} \mu(T - T_0)\,\mathrm{d}\delta\Omega = 0 \quad \forall(U,\mu)\end{aligned} \tag{9.41}$$

From the above we can easily see:

- the (strong) Dirichlet boundary condition $T = T_0$ will be satisfied on $\delta\Omega$;
- the (weak) Neumann boundary condition will not be satisfied with f_{imp}. In fact, the Lagrange multiplier λ will absorb entirely the flux f_{imp} that we have added on the boundary: $\lambda = \lambda' - f_{\text{imp}}$.

Here, λ' is the value that the Lagrange multiplier would achieve if we did not add the flux f_{imp} in the problem statement.

Imposing the boundary conditions in the CastɜM code

Currently, in fluid mechanics, the elimination method is generally used to prescribe Dirichlet boundary conditions via the use of the `'CLIM'` keyword of the `'EQEX'` operator. Therefore, flux balances are not so easy to check because the weak fluxes λ_i are not readily computed.

However, in the other domains of application of CastɜM (solid mechanics, heat conduction…), the discrete Lagrange multiplier method is used to prescribe Dirichlet boundary conditions via the use of the `'BLOQ'` operator (BLOQue is the french word for constrain) to constrain a degree of freedom and the `'DEPI'` operator (DEPI is a short-hand for DEPlacement Imposé which means prescribed displacement) to prescribe the value of the constraint. These operators can also be used in fluid mechanics.

We mention that the Lagrange multiplier method is a general method that can be used to prescribe any kind of constraint, not only Dirichlet boundary conditions. We have already studied the case of the $\nabla\cdot\boldsymbol{u} = 0$ constraint in the chapter dedicated to Stokes' problem. The CastɜM operator `'RELA'` (RELAtion) can be used to impose other kind of constraints, for instance: periodicity constraint, imposition of the value of a vector unknown in a given direction…

Generalization to other problems

In the remaining of the chapter, we will see that it is possible to use the method of multipliers to prescribe essential boundary conditions in other problems, not just diffusion problems. The physical meaning of the multiplier will be closely related to the natural boundary condition of the given problem (in the case of the discrete method of multiplier, the natural boundary condition integrated on the surface is considered). This duality is well-known in solid mechanics: when a displacement is prescribed, the corresponding

Lagrange multiplier has the meaning of a reaction force. This duality is as important in fluid mechanics although it is less often mentioned and used. In particular, we already saw that it allows to express the global conservation property of the finite element method in a simple and elegant way.

9.2.4 Mixed boundary conditions

Other than Dirichlet and Neumann boundary conditions, there exists a third type of boundary condition which is frequently used in the field of thermal transfer: the exchange boundary condition of *Robin* type:

$$-\alpha\nabla T\cdot\boldsymbol{n}=h\,(T-T_\infty) \tag{9.42}$$

where h is an exchange coefficient and T_∞ a representative temperature of the exterior (outside the domain being modeled). The Robin condition is called a *mixed* boundary condition because it involves both the unknown T and its boundary flux $-\alpha\nabla T\cdot\boldsymbol{n}$.

A Robin problem without source term can be expressed as a variational problem by modifying the Dirichlet problem (9.1) in the following way:

$$\min_{T\in\mathcal{H}^1(\Omega)} I(T)=\min_{T\in\mathcal{H}^1(\Omega)}\int_\Omega\frac{\alpha}{2}\|\nabla T\|^2\,\mathrm{d}\Omega+\int_{\delta\Omega}h\frac{T^2}{2}-hTT_\infty\,\mathrm{d}\delta\Omega \tag{9.43}$$

The minimization condition then reads:

$$\delta_U I(T)=\int_\Omega\alpha\nabla T\cdot\nabla U\,\mathrm{d}\Omega+\int_{\delta\Omega}h\,(T-T_\infty)\,U\,\mathrm{d}\delta\Omega=0\quad\forall U\in\mathcal{H}^1(\Omega) \tag{9.44}$$

After discretization:

$$\begin{aligned}&\int_\Omega\alpha\nabla T_h\cdot\nabla N_i\,\mathrm{d}\Omega+\int_{\delta\Omega}h\,(T_h-T_\infty)\,N_i\,\mathrm{d}\delta\Omega\\ &=\sum_{j\in\Omega}T_j(\int_\Omega\nabla N_j\cdot\nabla N_i\,\mathrm{d}\Omega)+\sum_{k\in\delta\Omega}T_k(\int_{\delta\Omega}hN_kN_i\,\mathrm{d}\delta\Omega)-b_i(T_\infty)=0\quad\forall i\end{aligned} \tag{9.45}$$

In matrix form, one gets:

$$\left(\mathsf{R}+h\bar{\mathsf{M}}\right)\boldsymbol{T}=h\bar{\mathsf{M}}\bar{\boldsymbol{T}}_\infty \tag{9.46}$$

We notice that the Robin condition involves a boundary mass matrix: $h\bar{\mathsf{M}}=\int_{\delta\Omega}hN_kN_i\,\mathrm{d}\delta\Omega$.

In Cast3M, the boundary mass matrix $\int_{\delta\Omega}hN_kN_i\,\mathrm{d}\delta\Omega$ and the boundary source term $\int_{\delta\Omega}hT_\infty N_i\,\mathrm{d}\delta\Omega$ can be computed with the `'ECHI'` operator (ECHange Imposé is the french word for prescribed exchange)[4].

Formally, the Robin boundary condition is closely related to the Neumann boundary condition because it is weakly prescribed. However, in the Robin problem, there is no indeterminacy of the unknown (and thus there is no compatibility condition) because T appears in the formulation of the problem, unlike in Neumann's problem where only ∇T appeared. From the matrix viewpoint, the linear system for Robin's problem which involves the boundary mass matrix added to the rigidity matrix is invertible (no zero eigenvalue).

The approximate solution T_h and the exact solution T both satisfy the following heat conservation property which writes, for the Robin problem at hand:

$$\int_{\delta\Omega}h\,(T_h-T_\infty)\,\mathrm{d}\delta\Omega=0 \tag{9.47}$$

[4] The `'CONV'` operator (CONVective exchange) can also be used.

9.2.5 An unsteady diffusion problem

We write the unsteady diffusion equation with Neumann boundary conditions semi-discretized in time by a finite difference method (implicit Euler):

$$\frac{T-\hat{T}}{\Delta t} - \alpha \Delta T = 0 \tag{9.48}$$

The boundary conditions are:

$$-\alpha \nabla T \cdot \boldsymbol{n} = -q \quad \text{sur} \quad \delta\Omega \tag{9.49}$$

T is the unknown temperature at the current timestep, $\hat{T}$ is the (known) temperature at the previous timestep and Δt is the timestep value.

This problem can be expressed in variational form as the solution of the following unsteady Neumann problem:

$$\min_{T\in\mathcal{H}^1(\Omega)} I(T) = \min_{T\in\mathcal{H}^1(\Omega)} \int_\Omega \frac{T^2}{2\Delta t} - \frac{\hat{T}T}{\Delta t} + \frac{\alpha}{2}\|\nabla T\|^2 \,\mathrm{d}\Omega - \int_{\delta\Omega} qT \,\mathrm{d}\delta\Omega \tag{9.50}$$

The minimization condition reads:

$$\delta_U I(T) = \int_\Omega \frac{T-\hat{T}}{\Delta t} U + \alpha \nabla T \cdot \nabla U \,\mathrm{d}\Omega - \int_{\delta\Omega} qU \,\mathrm{d}\delta\Omega = 0 \quad \forall U \in \mathcal{H}^1(\Omega) \tag{9.51}$$

After discretization:

$$\begin{aligned} &\int_\Omega \frac{T_h - \hat{T}}{\Delta t} N_i + \alpha \nabla T_h \cdot \nabla N_i \,\mathrm{d}\Omega - \int_{\delta\Omega} qN_i \,\mathrm{d}\delta\Omega \\ = &\sum_{j\in\Omega} T_j \Big(\int_\Omega \frac{1}{\Delta t} N_j N_i + \nabla N_j \cdot \nabla N_i \,\mathrm{d}\Omega\Big) - b_i(\hat{T}) = 0 \quad \forall i \end{aligned} \tag{9.52}$$

In matrix form, one gets:

$$\left(\frac{\mathsf{M}}{\Delta t} + \mathsf{R}\right)\boldsymbol{T} = \frac{\mathsf{M}}{\Delta t}\hat{\boldsymbol{T}} + \bar{\boldsymbol{q}} \tag{9.53}$$

We notice that the implicit Euler time discretization involves the mass matrix on the domain: $\frac{\mathsf{M}}{\Delta t} = \int_\Omega \frac{1}{\Delta t} N_j N_i \,\mathrm{d}\Omega$.

In Cast3M, the mass matrix $\int_\Omega \frac{1}{\Delta t} N_j N_i \,\mathrm{d}\Omega$ and the source term $\int_\Omega -\frac{\hat{T}N_i}{\Delta t}\,\mathrm{d}\Omega$ can be computed with the `'DFDT'` (Derivative of a Function with respect to Time)[5].

For the unsteady Neumann problem, there is no indeterminacy of the unknown (and thus no compatibility condition) because T appears in the formulation of the problem, unlike in the steady Neumann problem where only ∇T appeared. From the matrix viewpoint, the linear system for the unsteady Neumann problem which involves the mass matrix added to the rigidity matrix is invertible (no zero eigenvalue).

The approximate solution T_h and the exact solution T both satisfy the following global heat conservation property which writes, for the unsteady Neumann's problem at hand:

$$\int_\Omega T_h - \hat{T} \,\mathrm{d}\Omega = \Delta t \int_{\delta\Omega} q \,\mathrm{d}\delta\Omega \tag{9.54}$$

[5] The `'CAPA'` operator (CAPAcity) can also be used.

This reads: the heat increase in the domain from time t to time $t + \Delta t$ is equal to the integral of the incoming heat flux through the boundary of the domain.

We can also notice that, if we want the problem to have a steady solution T_h (such that $T_h = \hat{T}$), we are led back to a compatibility condition which looks like (9.11):

$$\int_{\delta\Omega} q \, \mathrm{d}\delta\Omega = 0 \tag{9.55}$$

9.3 A convection-diffusion problem

In this section, we show that there are several possible choices for discretizing the convective term (it is not derived in a natural way from a minimization principle). Each of these possible choices leads to *different* natural boundary conditions. This freedom of choice in the boundary conditions is a strong point of the finite-element method.

9.3.1 Different problem formulations

Non-conservative formulation

Let us write the steady homogeneous convection-diffusion equation in a *non-conservative* form without Dirichlet boundary conditions:

$$\boldsymbol{u} \cdot \nabla T - \operatorname{div} \alpha \nabla T = 0 \tag{9.56}$$

Applying the weighted residual method and integrating by parts, the following continuous formulation is obtained:

$$\int_{\Omega} \boldsymbol{u} \cdot \nabla T U + \alpha \nabla T \cdot \nabla U \, \mathrm{d}\Omega - \int_{\delta\Omega} \alpha \nabla T \cdot \boldsymbol{n} U \, \mathrm{d}\delta\Omega = 0 \quad \forall U \tag{9.57}$$

The natural boundary condition under the boundary integral is *identical* to the one obtained for the purely diffusive Neumann problem. It is a boundary condition on the *diffusive* flux:

$$-\alpha \nabla T \cdot \boldsymbol{n} = 0 \tag{9.58}$$

This diffusive flux is prescribed to be zero since we have no surface source term in (9.57).

Conservative formulation

Let us now write the steady homogeneous convection-diffusion equation in a *non-conservative* form without Dirichlet boundary conditions:

$$\operatorname{div} \boldsymbol{u} T - \operatorname{div} \alpha \nabla T = 0 \tag{9.59}$$

or:

$$\operatorname{div} (\boldsymbol{u} T - \alpha \nabla T) = 0 \tag{9.60}$$

When $\operatorname{div} \boldsymbol{u} = 0$, this PDE is equivalent to the non-conservative one. Applying the weighted residual method and integrating by parts, the following continuous formulation is obtained:

$$\int_{\Omega} (-\boldsymbol{u} T + \alpha \nabla T) \cdot \nabla U \, \mathrm{d}\Omega + \int_{\delta\Omega} (T \boldsymbol{u} \cdot \boldsymbol{n} - \alpha \nabla T \cdot \boldsymbol{n}) \, U \, \mathrm{d}\delta\Omega = 0 \quad \forall U \tag{9.61}$$

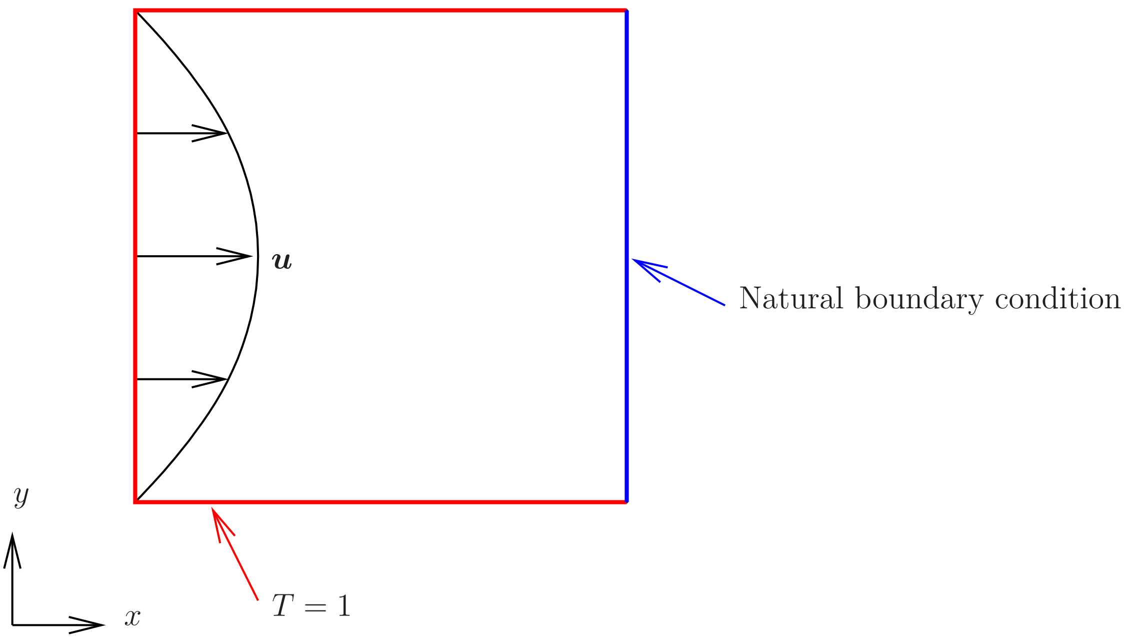


Figure 9.2: 2D steady convection-diffusion problem (9.63).

Notice that the convective term was also integrated by parts. The resulting natural boundary condition is *different* from the previous one. It is a homogeneous boundary condition on the *total flux*:

$$T\boldsymbol{u} \cdot \boldsymbol{n} - \alpha \nabla T \cdot \boldsymbol{n} = 0 \tag{9.62}$$

$T\boldsymbol{u}$ is the convective flux and $-\alpha\nabla T$ is the diffusive flux.

We can see that whether we integrate or do not integrate by parts the convective term leads to different types of natural boundary conditions.

Numerical example

We consider the following model problem:

$$\left\{ \begin{array}{l} \boldsymbol{u} \cdot \nabla T - \operatorname{div} \alpha \nabla T = 0 \\ T|_{\delta\Omega\setminus(x=1)} = 1 \\ \boldsymbol{u} = \begin{pmatrix} 4y(1-y) \\ 0 \end{pmatrix} \end{array} \right. \tag{9.63}$$

The problem is displayed in figure 9.2. Notice that: $\operatorname{div} \boldsymbol{u} = 0$.

In order to discretize the convective term, we use the Cast3M operator `'KONV'` with option `'NOCONS'` (by default) or `'CONS'`. These options correspond respectively to the non-conservative (9.57) and conservative (9.61) formulation of the convective term. Thus, the natural boundary conditions are:

1. `'NOCONS'` option: zero diffusive flux boundary condition

$$-\alpha \nabla T \cdot \boldsymbol{n}|_{x=1} = 0 \tag{9.64}$$

2. `'CONS'` option: zero total flux boundary condition

$$T\boldsymbol{u} \cdot \boldsymbol{n} - \alpha \nabla T \cdot \boldsymbol{n}|_{x=1} = 0 \tag{9.65}$$

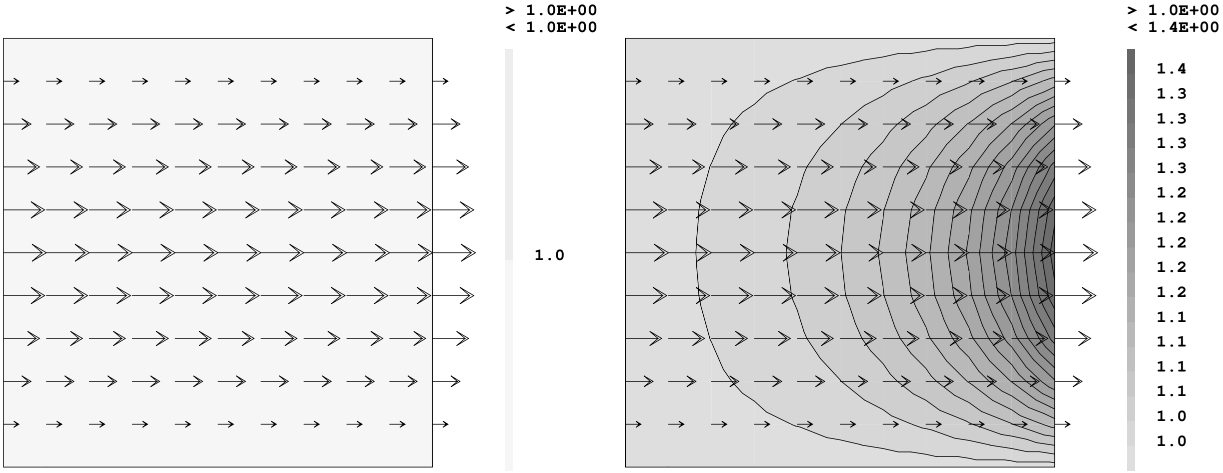


Figure 9.3: Numerical solution T_h (linear `LINE` finite element method) of the 2D steady convection-diffusion problem (9.63) with $\alpha = 1$ and natural exit boundary condition. Left: zero diffusive flux natural condition (9.64). Right: zero total flux natural condition (9.65).

The numerical results are shown in figure 9.3. In the case of the zero diffusive flux condition, the numerical solution is constant in space and is equal to 1. In the case of the zero total flux condition on the exit boundary, we notice that an incoming diffusive flux arises which exactly counterbalances the outgoing convective flux.

9.3.2 Indeterminacy of T and compatibility condition

In this subsection we focus on the case where only Neumann boundary conditions are prescribed on the boundary of the domain. Let us integrate the convection-diffusion equation (9.59) in conservative form on the domain Ω:

$$\int_{\Omega} \operatorname{div}\left(\boldsymbol{u}T - \alpha\nabla T\right)\,\mathrm{d}\Omega = \int_{\delta\Omega} T\boldsymbol{u}\cdot\boldsymbol{n} - \alpha\nabla T\cdot\boldsymbol{n}\,\mathrm{d}\delta\Omega = 0 \tag{9.66}$$

This expresses the *global conservation* of T: over the boundary $\delta\Omega$, the outgoing total flux is equal to the incoming total flux. This property does not depend on the discretization, it is intrinsic to the conservation equation at hand.

At the continuous or discrete level, the setting for the conservative formulation is similar to the one for Neumann's problem: on the one hand, T is indeterminate due to the fact that only its derivative appears in the formulation. On the other hand, choosing a particular variation U equal to the constant function 1 on Ω in equation (9.61), one obtains *exactly* the expression (9.66). This means that, if one wants to prescribe some non-zero total flux on the whole boundary $\delta\Omega$ (with the `'FIMP'` operator in the Cast3M code):

$$T\boldsymbol{u}\cdot\boldsymbol{n} - \alpha\nabla T\cdot\boldsymbol{n} = q \tag{9.67}$$

The following should hold:

$$\int_{\delta\Omega} q\,\mathrm{d}\delta\Omega = 0 \tag{9.68}$$

for the problem to admit a solution.

The situation is slightly different in the case of the non-conservative formulation: on the one hand, T is still indeterminate. On the other hand, as we only prescribe the diffusive flux, there is no compatibility condition that needs to hold on this flux. Nonetheless, the solution still satisfies a global conservation equation. As always, this equation is obtained by setting the variation U to unity in the conservation equation (9.57). If the prescribed diffusive flux is:

$$-\alpha\nabla T\cdot\boldsymbol{n}=q \tag{9.69}$$

Then the following holds:

$$\int_{\Omega}\boldsymbol{u}\cdot\nabla T\,\mathrm{d}\Omega-\int_{\delta\Omega}q\,\mathrm{d}\delta\Omega=0 \tag{9.70}$$

Integrating by parts[6], one gets:

$$\int_{\Omega}T\nabla\cdot\boldsymbol{u}+\nabla\cdot(T\boldsymbol{u})\,\mathrm{d}\Omega+\int_{\delta\Omega}q\,\mathrm{d}\delta\Omega=0 \tag{9.71}$$

When $\int_{\Omega}T\nabla\cdot\boldsymbol{u}\,\mathrm{d}\Omega=0$ (for instance, if $\nabla\cdot\boldsymbol{u}=0$ holds), an expression similar to (9.66) is obtained after applying the divergence theorem on the volume integral.

9.3.3 Upwinding

The use of the SUPG upwinding method (section 4.4.3) can modify the natural boundary condition. Indeed, this method adds a numerical diffusion term to the equation that we want to solve. More precisely, if we consider the convection-diffusion equation (9.56) in non-conservative form and applying the SUPG upwinding method, one gets:

$$\begin{cases}\boldsymbol{u}\cdot\nabla T-\operatorname{div}\mathsf{A}'\nabla T=0\\ \mathsf{A}'=\alpha\mathsf{I}+\frac{h_{\boldsymbol{u}}|\boldsymbol{u}|}{2}J(Pe_{h_{\boldsymbol{u}}})\frac{\boldsymbol{u}\otimes\boldsymbol{u}}{|\boldsymbol{u}|^2}\end{cases} \tag{9.72}$$

The natural boundary condition that arises by applying the weighted residual method is a Neumann condition similar to the one obtained in the pure diffusion case but with a modified and tensorial diffusion coefficient:

$$(-\mathsf{A}'\cdot\nabla T)\;\cdot\boldsymbol{n}=0 \tag{9.73}$$

This condition reduces to the usual Neumann condition:

$$-\alpha\nabla T\cdot\boldsymbol{n}=0 \tag{9.74}$$

in two cases:

1. whenever $J(Pe_{h_{\boldsymbol{u}}})=0$ i.e. the upwinding is inactive;

2. whenever $\boldsymbol{u}\cdot\boldsymbol{n}=0$ on the boundary. Indeed, in section 4.4.3, it is shown that upwinding adds some diffusion in the direction of the velocity $\boldsymbol{u}$ but adds no diffusion in the directions orthogonal to $\boldsymbol{u}$.

Exercise 8 *Show that* (9.74) *holds for case 2* ($\boldsymbol{u}\cdot\boldsymbol{n}=0$).

[6] Here, contrary to the Laplacian case, it is possible to integrate by parts also in the discrete formulation (if we replace T by T_h) because the basis function N_i are sufficiently regular.

9.3.4 Summary

The natural boundary condition for the convection-diffusion equation depends on the particular form chosen for the convective term: for instance, the non-conservative form $\boldsymbol{u} \cdot \nabla T$ or the conservative form div $(\boldsymbol{u}T)$. There is some freedom in the choice of the form because the convective term does not come from a variational formulation. The use of upwinding can also modify the natural boundary condition.

9.4 Stokes' problem

9.4.1 Essential and natural boundary conditions

In the previous section, we studied in some detail the boundary conditions for the convection-diffusion equation. In this subsection, we deal with the boundary conditions for the Stokes and Navier-Stokes' problem in lesser detail, mostly emphasizing the differences with the previous case. Let us rewrite Stokes' problem:

$$\left\{ \begin{array}{rcl} -\mu \boldsymbol{\Delta u} + \nabla p & = & 0 \\ \nabla \cdot \boldsymbol{u} & = & 0 \end{array} \right. \tag{9.75}$$

and its weak formulation:

$$\begin{aligned} \int_\Omega -\mu \boldsymbol{\Delta u} \cdot \boldsymbol{v} \, \mathrm{d}\Omega + \int_{\delta\Omega} (\mu \boldsymbol{\nabla u} \cdot \boldsymbol{n}) \cdot \boldsymbol{v} \, \mathrm{d}\delta\Omega + \int_\Omega \nabla p \cdot \boldsymbol{v} \, \mathrm{d}\Omega - \int_{\delta\Omega} p\boldsymbol{v} \cdot \boldsymbol{n} \, \mathrm{d}\delta\Omega \\ - \int_\Omega q \nabla \cdot \boldsymbol{u} \, \mathrm{d}\Omega = 0 \end{aligned} \tag{9.76}$$

The boundary terms, which vanish in the weak formulation when Dirichlet boundary conditions on $\boldsymbol{u}$ are prescribed on the whole boundary $\delta\Omega$, are:

$$\int_{\delta\Omega} (\mu \boldsymbol{\nabla u} \cdot \boldsymbol{n}) \cdot \boldsymbol{v} \, \mathrm{d}\delta\Omega - \int_{\delta\Omega} p\boldsymbol{v} \cdot \boldsymbol{n} \, \mathrm{d}\delta\Omega \tag{9.77}$$

or:

$$\int_{\delta\Omega} [(\mu \boldsymbol{\nabla u} - p\mathsf{I}) \cdot \boldsymbol{n}] \cdot \boldsymbol{v} \, \mathrm{d}\delta\Omega \tag{9.78}$$

The term $(\mu \boldsymbol{\nabla u} - p\mathsf{I}) \cdot \boldsymbol{n}$ is called *traction.* We shall denote it by $\boldsymbol{f}$. It has the physical dimension of a force per unit surface and is composed of a viscous part and a pressure part. It is the weakly prescribed natural boundary condition. The function $\boldsymbol{v}$ vanishes when Dirichlet boundary conditions are applied on $\boldsymbol{u}$: $\boldsymbol{u} = \boldsymbol{u_0}$. Thus, the two kind of boundary conditions that we can prescribe are:

Dirichlet	$\boldsymbol{v} = \boldsymbol{0}$	$\boldsymbol{u} = \boldsymbol{u_0}$	`'CLIM'`	essential
Neumann	$\boldsymbol{v}$ varies	$\boldsymbol{f} = \boldsymbol{f_0}$	`'TOIM'`	natural

Practically, in the context of fluid mechanics problem solved with the CastɜM code, on the one hand, Dirichlet boundary conditions are prescribed via the use of the `'CLIM'` keyword of the `'EQEX'` operator. On the other hand, the (surface) boundary source terms that corresponds to the weak natural boundary conditions are computed by the `'TOIM'` operator (τ IMposé in french, i.e. prescribed τ).

Formally, the boundary conditions are quite similar to the one encountered in the field of linear elasticity where we can prescribe either the displacement (instead of the velocity),

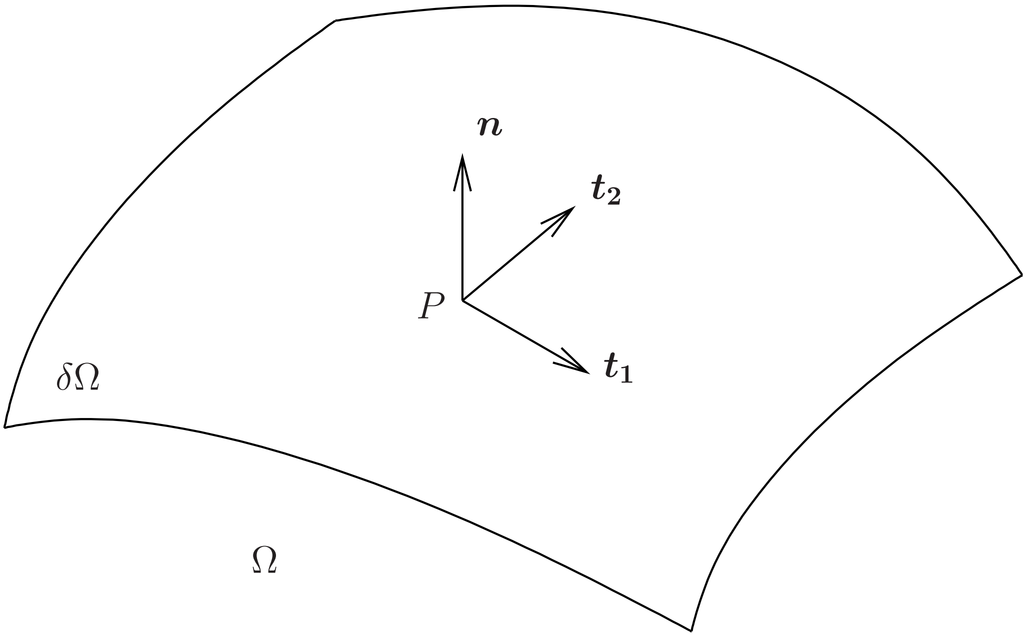


Figure 9.4: Local frame at point P on $\delta\Omega$.

or a surface force. Note however, that the traction $\boldsymbol{f}$ is not equivalent to the surface force, which is represented by the normal component of the stress tensor $\boldsymbol{\sigma} \cdot \boldsymbol{n}$. This is due to the fact that we have simplified the partial derivative equations of Stokes' problem by making use of $\nabla \cdot \boldsymbol{u} = 0$ to obtain $\mu \boldsymbol{\Delta u}$ from $\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}$. If we use the $\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}$ formulation of Stokes' problem, the natural boundary condition will be *different*: *force* instead of traction (more on this will be given in section 9.4.5).

9.4.2 Boundary conditions by direction

In fact, for Stokes' problem, we are prescribing boundary conditions on vectorial quantities, instead of scalar quantities as was the case for the convection-diffusion problem. Thus, it is possible to prescribe Dirichlet conditions along one or more directions and Neumann conditions in the remaining directions, orthogonal to Dirichlet's. Let us clarify this by introducing a local frame on the domain's boundary, as shown in figure 9.4.

$\boldsymbol{u}$ and $\boldsymbol{f}$ can be expressed in this local frame as:

$$\left\{ \begin{array}{l} \boldsymbol{u} = u_{\boldsymbol{n}} \boldsymbol{n} + u_{\boldsymbol{t_1}} \boldsymbol{t_1} + u_{\boldsymbol{t_2}} \boldsymbol{t_2} \\ \boldsymbol{f} = f_{\boldsymbol{n}} \boldsymbol{n} + f_{\boldsymbol{t_1}} \boldsymbol{t_1} + f_{\boldsymbol{t_2}} \boldsymbol{t_2} \end{array} \right. \tag{9.79}$$

with:

$$\left\{ \begin{array}{l} f_{\boldsymbol{n}} = \mu \left(\boldsymbol{\nabla u} \cdot \boldsymbol{n} \right) \cdot \boldsymbol{n} - p \\ f_{\boldsymbol{t_1}} = \mu \left(\boldsymbol{\nabla u} \cdot \boldsymbol{n} \right) \cdot \boldsymbol{t_1} \\ f_{\boldsymbol{t_2}} = \mu \left(\boldsymbol{\nabla u} \cdot \boldsymbol{n} \right) \cdot \boldsymbol{t_2} \end{array} \right. \tag{9.80}$$

Now, let us consider two frequently happening situations:

1. Boundary conditions at a pipe's exit end in *established regime*; one can prescribe for instance:

Dirichlet	$u_{\boldsymbol{t_1}} = 0$	'CLIM'
Dirichlet	$u_{\boldsymbol{t_2}} = 0$	'CLIM'
Neumann	$f_{\boldsymbol{n}} = \mu \left(\boldsymbol{\nabla u} \cdot \boldsymbol{n}\right) \cdot \boldsymbol{n} - p = 0$	'TOIM'

If we assume that: $\mu \left(\boldsymbol{\nabla u} \cdot \boldsymbol{n}\right) \cdot \boldsymbol{n}$ is negligible compared to p i.e., either μ is small, or $\boldsymbol{\nabla u} \cdot \boldsymbol{n}$ is small in the normal direction to the pipe exit surface (this is what we call established regime) then the Neumann boundary condition is close to a *prescribed exit pressure* condition.

2. Boundary condition of *prescribed traction* type on a wall, one can prescribe for instance:

Dirichlet	$u_{\boldsymbol{n}} = 0$	'CLIM'
Neumann	$f_{\boldsymbol{t_1}} = f_{0\boldsymbol{t_1}}$	'TOIM'
Neumann	$f_{\boldsymbol{t_2}} = f_{0\boldsymbol{t_2}}$	'TOIM'

9.4.3 Mixed boundary conditions

Similarly to the convection-diffusion case (section 9.2.4), a third type of boundary condition is frequently used, called mixed or Robin boundary condition. It is often presented in the form:

$$\boldsymbol{f} = -k \left(\boldsymbol{u} - \boldsymbol{u}_\infty\right) \tag{9.81}$$

This type of boundary condition can model *pressure head loss* Traduction de : pertes de charge due to friction at a wall. In CastƎM, the 'FROT' (FROTtement is french for friction) is used to discretize this boundary condition. As an example, similarly to the case of prescribed traction on a wall, one can prescribe:

Dirichlet	$u_{\boldsymbol{n}} = 0$	'CLIM'
Neumann	$f_{\boldsymbol{t_1}} = -k \left(u_{\boldsymbol{t_1}} - u_{\boldsymbol{t_1}\infty}\right)$	'FROT'
Neumann	$f_{\boldsymbol{t_2}} = -k \left(u_{\boldsymbol{t_2}} - u_{\boldsymbol{t_2}\infty}\right)$	'FROT'

9.4.4 Compatibility conditions

For Stokes' and Navier-Stokes' problems, there are potentially two compatibility conditions that have to be satisfied, corresponding to the two conservation equations for momentum and mass.

Compatibility condition for the momentum equation

If only Neumman boundary conditions: $\boldsymbol{f} = \boldsymbol{f_0}$ are prescribed on $\delta\Omega$ (no Dirichlet condition on the velocity) for the steady homogeneous (no source term) Stokes' problem then the following compatibility condition should hold:

$$\int_{\delta\Omega} \boldsymbol{f_0} \, \mathrm{d}\delta\Omega = \boldsymbol{0} \tag{9.82}$$

for the problem to admit a solution. Dual to this condition, there is an indeterminacy on the velocity unknown $\boldsymbol{u}$, which is only known up to a constant vector. Indeed, only the velocity derivatives appear in the weak formulation (9.76). This is similar to the well-known result in single-body mechanics *Traduction de : mécanique du point*: a system subject to zero resulting forces moves in a straight line at constant speed: $\boldsymbol{u} = \mathrm{C}^{\mathrm{ste}}$.

Compatibility condition for the mass equation

This condition is encountered in practice much more frequently than the previous one. Indeed, global conservation of mass states that:

$$\int_{\Omega} \nabla \cdot \boldsymbol{u} \, \mathrm{d}\Omega = \int_{\delta\Omega} \boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}\delta\Omega = 0 \tag{9.83}$$

should hold. In particular, if one prescribes the value of $u_{\boldsymbol{n}}$ everywhere on $\delta\Omega$, the previous condition must be verified for the problem to admit a solution. This condition is met if we study the flow in a *closed cavity* as: $u_{\boldsymbol{n}} = 0$ on $\delta\Omega$.

Note that this condition is slightly different in nature from the other compatibility conditions that we have met up to this point. Indeed, it is a condition on the unknowns subject to Dirichlet conditions and not on the unknown variables fluxes.

Dual to this condition, there is an *indeterminacy* in the pressure variable p, which is only known up to a constant. Indeed, only the pressure gradient appears in the weak formulation (9.76). In order to get rid of this indeterminacy and select a unique solution, one can, for instance, prescribe the value of p at a randomly-chosen node.

9.4.5 Other formulations of the Stokes' problem

We want to mention that there exist several ways to write the viscous term in the (Navier-)Stokes equations. Up to now, we used the formulation with a vector Laplacian on the velocity: $\mu\boldsymbol{\Delta u}$. This formulation is frequently used for practical reasons: in particular, when using Cartesian coordinates, the vector Laplacian reduces to a scalar Laplacian in each of the velocity's components.

Rotational form However, it is also possible to use other formulations. For example, with the help of the vector identity:

$$\boldsymbol{\Delta u} = \mathrm{rot}\,\mathrm{rot}\,\boldsymbol{u} - \nabla\,\mathrm{div}\,\boldsymbol{u} \tag{9.84}$$

and setting the second term to zero due to the fluid's incompressibility ($\mathrm{div}\,\boldsymbol{u} = 0$), one gets the rotational form of the problem. An important remark is that, on the one hand, from the PDE viewpoint, nothing is really changed. On the other hand, from the variational viewpoint, the natural boundary conditions are *different.* Indeed, after integration by parts, the boundary integral now writes:

$$\int_{\delta\Omega} (\boldsymbol{n} \wedge \mathrm{rot}\,\boldsymbol{u}) \cdot \boldsymbol{v} \, \mathrm{d}\delta\Omega \tag{9.85}$$

Viscous stress tensor form Another more useful form is based on the complete expression of the viscous stress tensor for a Newtonian fluid:

$$\boldsymbol{\tau} = \mu\left(\boldsymbol{\nabla u} + \boldsymbol{\nabla}^t\boldsymbol{u}\right) - \frac{2}{3}\mu\,(\mathrm{div}\,\boldsymbol{u})\,\mathsf{I} \tag{9.86}$$

Whenever μ is constant and $\nabla \cdot \boldsymbol{u} = 0$, the divergence of the viscous stress tensor is algebraically equivalent to the vector Laplacian because of the following identities:

$$\begin{cases} \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}^t\boldsymbol{u} = \nabla\,\mathrm{div}\,\boldsymbol{u} \\ \boldsymbol{\nabla} \cdot \,((\mathrm{div}\,\boldsymbol{u})\,\mathsf{I}) = \nabla\,\mathrm{div}\,\boldsymbol{u} \end{cases} \tag{9.87}$$

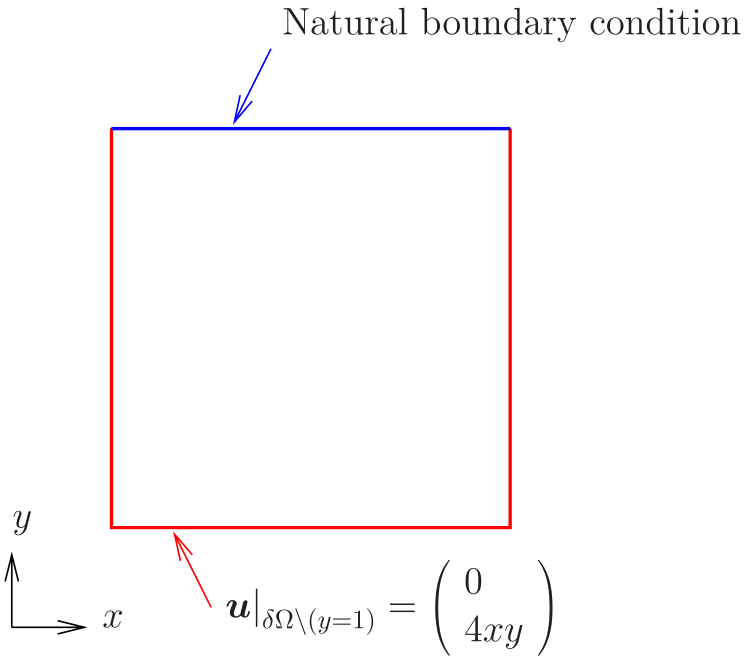

Natural boundary condition
y
x
$\boldsymbol{u}|_{\delta\Omega\backslash(y=1)} = \begin{pmatrix} 0 \\ 4xy \end{pmatrix}$

Figure 9.5: Steady 2D Navier-Stokes problem (9.89).

However, the natural boundary conditions for the two formulations are different. Integrating by parts the divergence of the stress tensor and neglecting the div $\boldsymbol{u}$ term leads to:

$$\begin{aligned} -\int_\Omega \boldsymbol{\nabla}\cdot\boldsymbol{\tau}\boldsymbol{v}\,\mathrm{d}\Omega &= \int_\Omega \mu\left(\boldsymbol{\nabla}\boldsymbol{u}:\boldsymbol{\nabla}\boldsymbol{v} + \boldsymbol{\nabla}^t\boldsymbol{u}:\boldsymbol{\nabla}\boldsymbol{v}\right)\,\mathrm{d}\Omega \\ &\quad -\int_{\delta\Omega}\mu\left(\left(\boldsymbol{\nabla}\boldsymbol{u}+\boldsymbol{\nabla}^t\boldsymbol{u}\right)\cdot\boldsymbol{n}\right)\cdot\boldsymbol{v}\,\mathrm{d}\delta\Omega \end{aligned} \tag{9.88}$$

The $\mu\left(\boldsymbol{\nabla}\boldsymbol{u}+\boldsymbol{\nabla}^t\boldsymbol{u}\right)\cdot\boldsymbol{n} = \boldsymbol{\tau}\cdot\boldsymbol{n}$ term is the viscous force acting on $\delta\Omega$. Adding the pressure force term $-p\boldsymbol{n}$, one obtains the total force acting on $\delta\Omega$, which is the natural boundary condition for this formulation.

9.5 Navier-Stokes' problem

In this section, we shall contempt ourselves with displaying the numerical results for a steady Navier-Stokes problem where a conservative or a non-conservative formulation has been used for the convective term, and a vector Laplacian or a divergence of the stress tensor form has been used for the viscous term. This corresponds to choosing the `'NOCONS'` (default) or the `'CONS'` option of the `'KONV'` operator and the `'MUCONS'` (default) or the `'FTAU'` option of the `'LAPN'` operator. The problem is defined as follows:

$$\left\{\begin{array}{l} (\boldsymbol{\nabla}\boldsymbol{u})\cdot\boldsymbol{u} = -\nabla p^* + \frac{1}{Re}\boldsymbol{\Delta}\boldsymbol{u} \\ \nabla\cdot\boldsymbol{u} = 0 \\ \boldsymbol{u}|_{\delta\Omega\backslash(y=1)} = \begin{pmatrix} 0 \\ 4xy \end{pmatrix} \end{array}\right. \tag{9.89}$$

Figure 9.5 illustrates this problem.

The determination of the natural boundary conditions for every different formulation of the problem is left to the reader, as is the interpretation of the numerical results of figure 9.6, in particular the velocity profile at $y = 1$.

9.6 Summary

This chapter was focused on the various boundary conditions that one can prescribe for a finite element discretization of the Navier-Stokes' and other related equations. Conservation properties of the method and compatibility conditions were also described.

We emphasize the following important practical points:

1. The Lagrange multiplier method is useful for prescribing Dirichlet boundary conditions. Indeed, the multiplier has the physical meaning of a flux, bringing forward the duality between Dirichlet and Neumann boundary conditions.

2. The flux obtained by the Lagrange multiplier method is also the most natural one when it comes to writing the global conservation of a quantity, in the context of solving conservation equations discretized by the finite element method.

3. The discrete Lagrange multiplier method is algebraically equivalent to the elimination method for prescribing Dirichlet boundary conditions. That is, the solution is exactly the same for the two methods and these methods satisfy the same conservation properties.

4. If natural boundary conditions are prescribed on the whole boundary of the domain for a steady conservation equation written in conservative form, then there a compatibility condition that has to be satisfied. The most frequently encountered compatibility condition is the one associated with the mass conservation equation $\nabla \cdot \boldsymbol{u} = 0$: if $\boldsymbol{u} \cdot \boldsymbol{n}$ is prescribed on the whole boundary $\delta\Omega$, then one must have $\int_{\delta\Omega} \boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}\delta\Omega = 0$.

5. Dual to this compatibility condition, we have the indeterminacy of an unknown (in this particular case, the pressure p). In order to get rid of this indeterminacy, one can prescribe the value of the unknown at one node.

6. There are several ways to write the convection terms $\boldsymbol{u} \cdot \nabla T$, $(\boldsymbol{\nabla u}) \cdot \boldsymbol{u}$ and the viscous term $\mu \boldsymbol{\Delta u}$ that are equivalent from the PDE (strong formulation) viewpoint, but which lead to different natural boundary conditions for the weak formulation and thus, for the discretized problem.

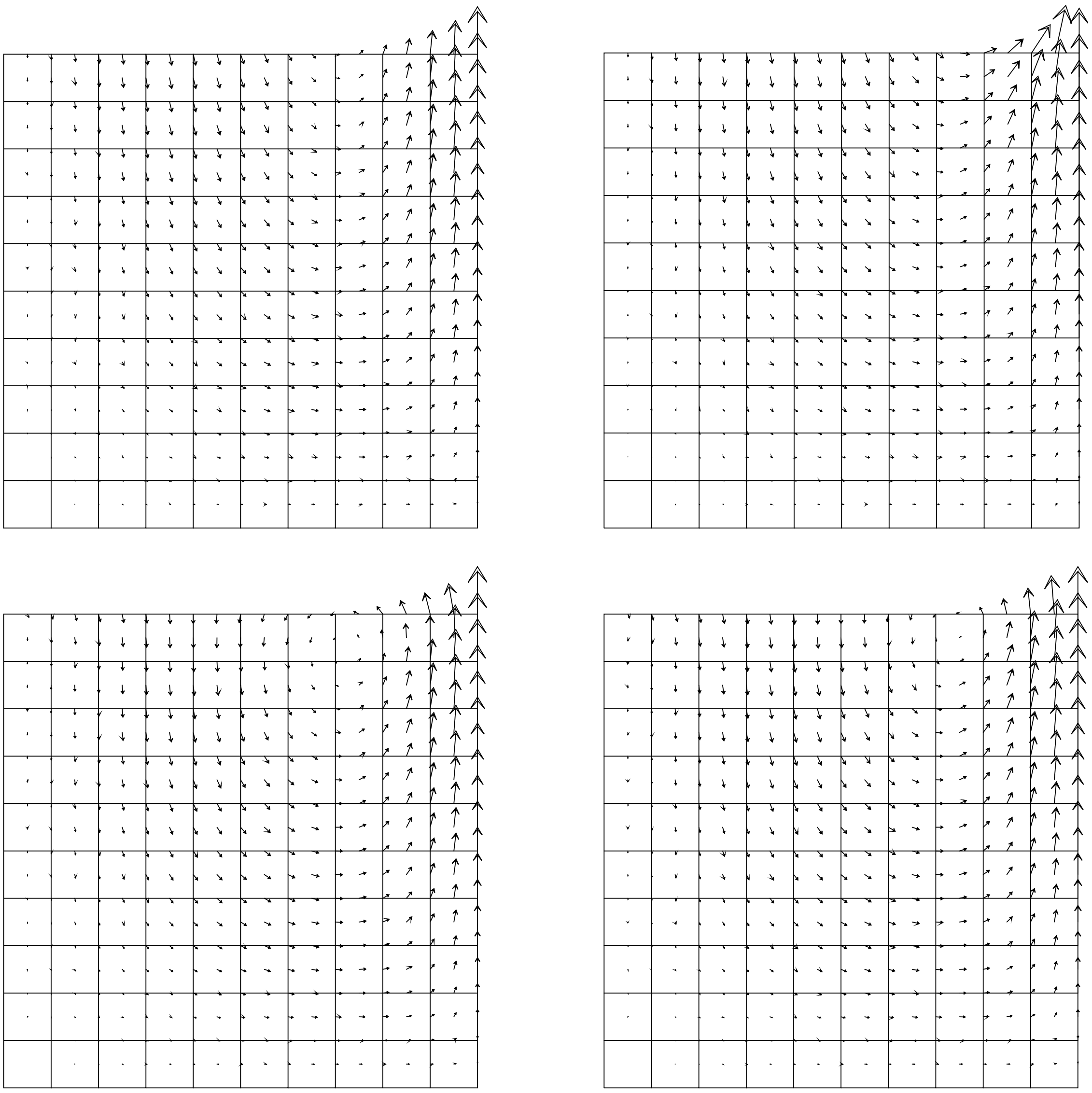


Figure 9.6: Numerical solution $\boldsymbol{u}_h$ (`QUAF/CENTREP1` finite elements) of the steady 2D Navier-Stokes problem (9.89) with $Re = 15$. Up: vector Laplacian form of the viscous term (`'MUCONS'`). Down: stress tensor form of the viscous term (`'FTAU'`). Left: non-conservative form of the convective term (`'NOCONS'`). Right: conservative form of the convective term (`'CONS'`).

Chapter 10

Practical solution method for an unsteady non-linear problem

$$\begin{aligned}
\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\cdot \boldsymbol{u} &= -\nabla p^* + \nu \boldsymbol{\Delta u} + \boldsymbol{s_u} \\
\nabla \cdot \boldsymbol{u} &= 0 \\
\frac{\partial T}{\partial t} + \boldsymbol{u}\cdot\nabla T &= \alpha \Delta T + s_T
\end{aligned}$$

In the previous chapters, we gave some excerpts of CastɜM data files. In this chapter, we describe in more detail how the discretization operators work in CastɜM. These operators are used to compute the matrices and right-hand sides (RHS) which are the result of the finite element spatial discretization process.

First, we *describe* the problem at hand which consists in the PDEs that we want to solve together with their boundary and initial conditions. This description is mainly done with the `'EQEX'` operator, described in section 10.1.

Second, we *solve* the problem at hand by calling the `EXEC` procedure. This procedure implements a relaxed fixed-point algorithm which allows to solve non-linear unsteady problems[1]. The algorithm is described in section 10.2.

10.1 Gibiane-language description of a problem

10.1.1 Spatial discretization operator syntax

All the fluid-mechanics discretization operators in CastɜM (a list is given in table 10.2) have a unified syntax:

```
  chpo matk = OPER RVX ;
```

where `OPER` is the operator's name, `RVX` is a `TABLE`-type object containing all the necessary inputs for the operator, `matk` is a `MATRIK`-type object and `chpo` is a `CHPOINT`-type object which are the matrix and the right-hand side corresponding to the discretization operator.

[1] Linear and/or steady problems are simplified cases of the former and can also be solved by the algorithm.

One of these two objects can be void if the operator is expected to create only a matrix or a right-hand side.

10.1.2 Creation of the table of inputs

The structure of the table of inputs RVX for each discretization operator is slightly complicated. This is the reason why an auxiliary operator, called 'EQEX', is generally used to generate the RVX tables and some other pieces of information. In fact, the 'EQEX' operator allows one to describe all the parameters of the fluid mechanics problem at hand. The generic syntax of such a description is:

```
RV = 'EQEX' MOT1 val1 MOT2 val2 ...
         'OPTI' motopt11 motopt12 ....
         'ZONE' mod1 'OPER' oper1 arg11 arg12 ... 'INCO' nominco1
         'OPTI' motopt21 motopt22 ....
         'ZONE' mod2 'OPER' oper2 arg21 arg22 ... 'INCO' nominco2
          ...
         'CLIM' nominco1 typclim1  mail1 chclim1
         'CLIM' nominco2 typclim2  mail2 chclim2
          ...
          ;
RV . 'INCO' = 'TABLE' 'INCO' ;
RV . 'INCO' . nominco1 = 'KCHT' mod1 typinco1 disinco1 valinco1 ;
RV . 'INCO' . nominco2 = 'KCHT' mod2 typinco2 disinco2 valinco2 ;
...
```

RV is TABLE-type object containing all the given data, structured in a particular way. The given data are:

1. global *optional* data given as keyword-value pairs (MOT-clé valeur in french). Table 10.1 lists the main global optional data.

2. *discretization operators* oper1, oper2... The operator oper1 acts on the domain mod1, deals with the unknown whose name is nominco1 and needs the information given by arguments arg1*i*. motopt1*i* are keywords specifying the discretization options for the oper1 operator. Table 10.2 lists the main discretization operators.

3. *prescribed values* for the unknowns (Dirichlet boundary conditions) using the 'CLIM' keyword;

4. *initial values* for the unknowns. These are not initialized by the 'EQEX' operator and must be put "manually" in the RV . 'INCO' table. This is mandatory even if a steady problem is to be solved. Indeed, no initial conditions are necessary in this case but the data in the table is also used to retrieve some information about the discretization spaces of the unknowns.

Remark 5 *It is recommended to specify the following options for each discretization operator* oper*i*: 'EF' *(Finite Element or Éléments Finis in french),* 'IMPL' *(IMPLicit) and* 'CENTREE' *(CENTERED spatial discretization, CENTRÉE in french). This allows*

Keyword	Value	Description
ITMA	$n_{\texttt{ITMA}}$	Number of time steps
NITER	$n_{\texttt{NITER}}$	Number of non-linear iterations
OMEGA	ω	Relaxation factor for non-linear iterations

Table 10.1: Main global options for fluid-mechanics computations.

to override any other default options. An exception is the convection operator 'KONV'*, for which the upwind* 'SUPG' *or the discontinuity-capturing* 'SUPGDC' *options can be specified instead of* 'CENTREE' *(see section 4.7 for details).*

Remark 6 *The online information for every Cast3M operator is accessible by typing* 'INFO' oper ; *at the Cast3M prompt or via the website* ***http://www-cast3m.cea.fr/***[2]*. In particular, the available discretization options and the necessary arguments for the discretization operators can be found here.*

For each discretization operator oper*i* (following the 'OPER' keyword), 'EQEX' will create a corresponding RVX table stored at the RV . *i*oper*i* index.

This table has the following structure:

- RVX . 'EQEX' points to the RV table;
- RVX . 'DOMZ' points to the domain mod*i*;
- RVX . 'ARG*j*' refers to the *j*th argument of oper*i*;
- RVX . 'LISTINCO' contains the unknown's name nominco*i*.

10.2 Non-linear unsteady problem solver

We recall here, by increasing order of complexity, some model problems we have dealt with before describing the complete solution algorithm.

10.2.1 Steady linear problem

In this type of problem, the time variable is not present (no partial derivative in time operator $\frac{\partial}{\partial t}$). Moreover, all the operators of the given PDE are linear with respect to the unknowns. An example of such a problem is the steady convection-diffusion equation (see chapter 4):

$$\boldsymbol{u} \cdot \nabla T - \alpha \Delta T = 0 \tag{10.1}$$

Here, $\boldsymbol{u}$ is a given vector field and T is the unknown variable. Spatially discretizing equation (10.1) by the finite element method leads to:

$$\mathsf{L}\boldsymbol{x} = \boldsymbol{b} \tag{10.2}$$

[2] The website may contain manual pages that are more recent than the one given by 'INFO'. The reference information is the one given by 'INFO' because it corresponds to the Cast3M version in use.

Operator	Name	Options	Arguments	Matrix	RHS
Volume terms					
$\left\{\begin{array}{l} -\alpha\Delta T \\ \alpha\nabla T\cdot\boldsymbol{n}\vert_{\delta\Omega_{\mathrm{N}}}\end{array}\right.$	LAPN	–	α	$\int_\Omega \alpha\nabla N_i\nabla N_j$	–
$\left\{\begin{array}{l} -\nu\boldsymbol{\Delta u} \\ \nu\boldsymbol{\nabla u}\cdot\boldsymbol{n}\vert_{\delta\Omega_{\mathrm{N}}}\end{array}\right.$	LAPN	$\underline{\texttt{MUCONS}}$	ν	$\int_\Omega \nu\boldsymbol{\nabla N}_i:\boldsymbol{\nabla N}_j$	–
$\left\{\begin{array}{l} -\boldsymbol{\nabla}\cdot\boldsymbol{\tau} \\ \boldsymbol{\tau}\cdot\boldsymbol{n}\vert_{\delta\Omega_{\mathrm{N}}}\end{array}\right.$	LAPN	FTAU	$\mu,\boldsymbol{u}$	$\int_\Omega \mu\boldsymbol{\nabla N}_i:\boldsymbol{\nabla N}_j$	$\int_\Omega -\mu\boldsymbol{\nabla N}_i:\boldsymbol{\nabla}^t\boldsymbol{u}$
$\rho\boldsymbol{u}\cdot\nabla T$	KONV	CENTREE	$\rho,\boldsymbol{u}$	$\int_\Omega \rho\boldsymbol{u}\cdot\nabla N_j N_i$	–
$\left\{\begin{array}{l} \rho\boldsymbol{u}\cdot\nabla T \\ \mathsf{B}\nabla T\cdot\boldsymbol{n}\vert_{\delta\Omega_{\mathrm{N}}}\end{array}\right.$	KONV	SUPG	$\rho,\boldsymbol{u},\alpha$	$+\int_\Omega \nabla N_j\mathsf{B}(\rho,\boldsymbol{u},\alpha)\nabla N_i$	–
$\left\{\begin{array}{l} \rho\boldsymbol{u}\cdot\nabla T \\ (\mathsf{B}+\mathsf{C})\nabla T\cdot\boldsymbol{n}\vert_{\delta\Omega_{\mathrm{N}}}\end{array}\right.$	KONV	$\underline{\texttt{SUPGDC}}$	$\rho,\boldsymbol{u},\alpha,T$	$+\int_\Omega \nabla N_j\mathsf{C}(\rho,\boldsymbol{u},\alpha,T)\nabla N_i$	–
$\rho\boldsymbol{u}\cdot\nabla T$	KONV	$\underline{\texttt{NOCONS}}$	$\rho,\boldsymbol{u}$	$\int_\Omega \rho\boldsymbol{u}\cdot\nabla N_j N_i$	–
$\left\{\begin{array}{l} \nabla\cdot\rho\boldsymbol{u}T \\ -T\boldsymbol{u}\cdot\boldsymbol{n}\vert_{\delta\Omega_{\mathrm{N}}}\end{array}\right.$	KONV	CONS	$\rho,\boldsymbol{u}$	$\int_\Omega -\rho\boldsymbol{u}N_j\nabla N_i$	–
$\rho\left(\boldsymbol{\nabla u}\right)\cdot\boldsymbol{u}$	KONV	$\underline{\texttt{NOCONS}}$	$\rho,\boldsymbol{u}$	$\int_\Omega \rho\left(\boldsymbol{\nabla N}_j\right)\cdot\boldsymbol{uN}_i$	–
$\left\{\begin{array}{l} \boldsymbol{\nabla}\cdot\rho\boldsymbol{u}\otimes\boldsymbol{u} \\ -\rho(\boldsymbol{u}\otimes\boldsymbol{u})\cdot\boldsymbol{n}\vert_{\delta\Omega_{\mathrm{N}}}\end{array}\right.$	KONV	CONS	$\rho,\boldsymbol{u}$	$\int_\Omega -\rho(\boldsymbol{u}\otimes\boldsymbol{N}_j)\boldsymbol{\nabla N}_i$	–
$-\nabla\cdot\boldsymbol{u}$	KMAB	–	–	$\int_\Omega -\nabla\cdot\boldsymbol{N}_j N_i$	–
$\left\{\begin{array}{l} \nabla p \\ -p\boldsymbol{n}\vert_{\delta\Omega_{\mathrm{N}}}\end{array}\right.$	KMBT	–	–	$\int_\Omega -N_j\nabla\cdot\boldsymbol{N}_i$	–
$\begin{pmatrix} 0 & \nabla p \\ -\nabla\cdot\boldsymbol{u} & 0\end{pmatrix}$	KBBT	–	–	$\int_\Omega -\nabla\cdot\boldsymbol{N}_j N_i - N_j\nabla\cdot\boldsymbol{N}_i$	–
$\rho\dfrac{T^{k+1}-T^k}{\Delta t}$	DFDT	CENTREE	$\rho,T^k,\Delta t$	$\int_\Omega \frac{\rho}{\Delta t}N_iN_j$	$\int_\Omega \frac{\rho}{\Delta t}T^kN_i$
$\rho\dfrac{\boldsymbol{u}^{k+1}-\boldsymbol{u}^k}{\Delta t}$	DFDT	CENTREE	$\rho,\boldsymbol{u}^k,\Delta t$	$\int_\Omega \frac{\rho}{\Delta t}\boldsymbol{N}_i\boldsymbol{N}_j$	$\int_\Omega \frac{\rho}{\Delta t}\boldsymbol{u}^k\boldsymbol{N}_i$
$-s_T$	FIMP	–	s_T	–	$\int_\Omega s_TN_i$
$\begin{array}{l}\rho c\boldsymbol{u}\cdot\nabla T \\ \quad -\lambda\Delta T - s_T\end{array}$	TSCA	–	$\rho c,\boldsymbol{u},\lambda,s_T$	$\int_\Omega \rho c\boldsymbol{u}\cdot\nabla N_jN_i+\lambda\nabla N_i\nabla N_j$	$\int_\Omega s_TN_i$
$\begin{array}{l}\rho(\boldsymbol{\nabla u})\cdot\boldsymbol{u} \\ \quad -\mu\boldsymbol{\Delta u}-\boldsymbol{s_u}\end{array}$	NS	–	$\rho,\boldsymbol{u},\mu,\boldsymbol{s_u}$	$\int_\Omega \rho(\boldsymbol{\nabla N}_j)\cdot\boldsymbol{uN}_i+\mu\boldsymbol{\nabla N}_i\boldsymbol{\nabla N}_j$	$\int_\Omega \boldsymbol{s_u}N_i$
Surface terms					
$-q$	FIMP	–	q	–	$\int_{\delta\Omega} qN_i$
$h\left(T-T_\infty\right)$	ECHI	–	h,T_∞	$\int_{\delta\Omega} hN_jN_i$	$\int_{\delta\Omega} hT_\infty N_i$
$-\boldsymbol{f_0}$	TOIM	–	$\boldsymbol{f_0}$	–	$\int_{\delta\Omega} \boldsymbol{f_0N}_i$
$k\left(\boldsymbol{u}-\boldsymbol{u}_\infty\right)$	FROT	–	$k,\boldsymbol{u}_\infty$	$\int_{\delta\Omega} k\boldsymbol{N}_j\boldsymbol{N}_i$	$\int_{\delta\Omega} k\boldsymbol{u}_\infty\boldsymbol{N}_i$

Table 10.2: Main CastƐM discretization operators for fluid-mechanics problems. Underlined options are the default options.

$\boldsymbol{x}$ is an unknown vector carrying all the nodal values T_i at the mesh vertices $i \in [1, n]$. L is a $n \times n$ square matrix, not depending on the unknown $\boldsymbol{x}$, coming from the spatial discretization of the two operators $\boldsymbol{u} \cdot \nabla$ and $-\alpha\Delta$. $\boldsymbol{b}$ is a right-hand side vector carrying all the contributions of known terms, typically source terms (equal to 0 in our example equation (10.1)) and boundary conditions.

Solving problem (10.1) after discretization amounts to solving one linear system (10.2) in the unknowns $\boldsymbol{x}$ involving the square matrix L. In Cast3M, operator `'KRES'` or `'RESO'` are used for this purpose.

10.2.2 Unsteady linear problem

In this type of problem, a partial derivative in time operator $\frac{\partial}{\partial t}$ is present and all the operators of the given PDE are linear with respect to the unknowns. An example of such a problem is the heat equation of section 5.1:

$$\frac{\partial T}{\partial t} - \alpha \Delta T = 0 \tag{10.3}$$

A common practice is to first discretize in time by a finite-difference method. The time interval $[0, t]$ upon which we want to solve the problem is discretized into a finite number of points: $\{0, \Delta t, \dots, k\Delta t, (k+1)\Delta t, \dots, n_{\mathtt{ITMA}}\Delta t\}$[3]. Using an implicit Euler method time discretization, equation (10.3) writes:

$$\frac{T^{k+1} - T^k}{\Delta t} - \alpha \Delta T^{k+1} = 0 \quad \forall k \in [1, n_{\mathtt{ITMA}}] \tag{10.4}$$

where T^k denotes the unknown's value at time $k\Delta t$.

Thus we are led to solving $n_{\mathtt{ITMA}}$ linear steady-like problems sequentially. T^0, the initial condition is a data that allows to solve for T^1. Then, by induction, T^k being known, we can solve for T^{k+1}. After spatial discretization by the finite element method, problem (10.4) takes the following matrix form:

$$(\frac{\mathsf{M}}{\Delta t} + \mathsf{L})\boldsymbol{x}^{k+1} = \frac{\mathsf{M}\boldsymbol{x}^k}{\Delta t} \quad \forall k \in [1, n_{\mathtt{ITMA}}] \tag{10.5}$$

Solving problem (10.3) after discretization amounts to solving $n_{\mathtt{ITMA}}$ linear systems (10.5) sequentially in the unknowns $\boldsymbol{x}^{k+1}$ involving the square matrix $(\frac{\mathsf{M}}{\Delta t} + \mathsf{L})$, which does not depend on $\boldsymbol{x}$.

10.2.3 Steady non-linear problem

In this type of problem the time variable is not present (no partial derivative in time operator $\frac{\partial}{\partial t}$). However, at least one of the operator of the given PDE is non-linear with respect to the unknowns. An example of such a problem is the steady Burgers' equation (the unsteady version was studied in chapter 7):

$$(\boldsymbol{\nabla u}) \cdot \boldsymbol{u} = 0 \tag{10.6}$$

[3] For reason of simplicity, we have chosen here to discretize $[0, t]$ using $(n_{\mathtt{ITMA}} + 1)$ equally distributed points.

Fixed point method

A way of solving this type of problem is to bring it in a form similar to the steady linear problem by linearization. This point was thoroughly discussed in chapter 6. Here we use a fixed point (also called Picard) method. This requires to define an initial state $\boldsymbol{u}_{[0]}$ around which we linearize the PDE at hand. In our particular case we take the convective velocity as $\boldsymbol{u}_{[0]}$ and the convected velocity unknown as $\boldsymbol{u}_{[1]}$:

$$\left(\boldsymbol{\nabla} \boldsymbol{u}_{[1]}\right) \cdot \boldsymbol{u}_{[0]} = 0 \tag{10.7}$$

The operator is now linear in the $\boldsymbol{u}_{[1]}$ unknown. Then, we can discretize it spatially and solve the resulting linear system. Once $\boldsymbol{u}_{[1]}$ is known, we can linearize around $\boldsymbol{u}_{[1]}$ and iterate the process $n_{\texttt{NITER}}$ times:

$$\left(\boldsymbol{\nabla} \boldsymbol{u}_{[i+1]}\right) \cdot \boldsymbol{u}_{[i]} = 0 \quad \forall i \in [1, n_{\texttt{NITER}}] \tag{10.8}$$

Convergence of the process is assured when the iterations tend to a *fixed point*: $\boldsymbol{u}_{[i]} \overset{i\to\infty}{\longrightarrow} \boldsymbol{u}_{[\infty]}$. In practice, a finite number of non-linear iterations is undertook and we have to check that $\|\boldsymbol{u}_{[n_{\texttt{NITER}}]} - \boldsymbol{u}_{[n_{\texttt{NITER}}-1]}\|$ is “small”.

Thus we are led to solving $n_{\texttt{NITER}}$ linear steady-like problems sequentially. This kind of non-linear equations solver is called *fixed-point* or *Picard* iteration.

After spatial discretization by the finite element method, the i^{th} problem (10.8) takes the following matrix form:

$$\left[\mathsf{N}(\boldsymbol{x}_{[i]})\right] \boldsymbol{x}_{[i+1]} = \boldsymbol{b} \quad \forall i \in [1, n_{\texttt{NITER}}] \tag{10.9}$$

Solving problem (10.6) after discretization amounts to solving $n_{\texttt{NITER}}$ linear systems (10.9) sequentially in the unknowns $\boldsymbol{x}_{[i+1]}$ involving the square matrix N, which varies at each non-linear iteration i, because it depends on $\boldsymbol{x}_{[i]}$.

Relaxation of the fixed point method

Notice that, generally, there is no guaranty that the iterative process defined by (10.9) converges: this will depend on the spectral properties of the matrix N. Also, this matrix depends on the unknown, and the initial guess $\boldsymbol{x}_{[0]}$ plays an important role: it should not be chosen too far from the thought solution.

In order to enlarge the convergence radius of the fixed point iteration (10.9), one often introduces a *relaxation parameter* $0 < \omega \leq 1$ which leads to the following relaxed update for the unknown:

$$\mathsf{N}(\boldsymbol{x}_{[i]})\tilde{\boldsymbol{x}}_{[i+1]} = \boldsymbol{b} \tag{10.10}$$

$$\boldsymbol{x}_{[i+1]} = \omega \tilde{\boldsymbol{x}}_{[i+1]} + (1-\omega)\boldsymbol{x}_{[i]} \qquad \forall i \in [1, n_{\texttt{NITER}}] \tag{10.11}$$

A drawback of the relaxation method is that it can slow down the convergence towards the solution $\boldsymbol{x}_{[\infty]}$ (when convergence occurs).

```
Initial condition: $\boldsymbol{x}^0$
for $k = 0, n_{\texttt{ITMA}} - 1$ do {Time-stepping loop}
  $\boldsymbol{x}^{k+1}_{[0]} \leftarrow \boldsymbol{x}^k$
  for $i = 0, n_{\texttt{NITER}} - 1$ do {Fixed-point (non-linear) loop}
    $\mathsf{A} \leftarrow 0 \quad \boldsymbol{b} \leftarrow 0$
    for $j = 1, n_{\texttt{OPER}}$ do {Loop on the operators}
      $\mathsf{A} \leftarrow \mathsf{A} + \mathsf{A}_j\left(\boldsymbol{x}^{k+1}_{[i]}, \boldsymbol{x}^k\right)$
      $\boldsymbol{b} \leftarrow \boldsymbol{b} + \boldsymbol{b}_j\left(\boldsymbol{x}^{k+1}_{[i]}, \boldsymbol{x}^k\right)$
    end for
    $\tilde{\boldsymbol{x}}^{k+1}_{[i+1]} = \mathsf{A}^{-1}\boldsymbol{b}$ {Linear system resolution}
    $\boldsymbol{x}^{k+1}_{[i+1]} = \boldsymbol{x}^{k+1}_{[i]} + \omega\left(\tilde{\boldsymbol{x}}^{k+1}_{[i+1]} - \boldsymbol{x}^{k+1}_{[i]}\right)$ {Relaxed update}
  end for
  $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^{k+1}_{[n_{\texttt{NITER}}-1]}$
  $t^{k+1} \leftarrow t^k + \Delta t$
end for
```

Table 10.3: Relaxed Picard algorithm for solving unsteady non-linear problems in fluid dynamics.

10.2.4 Non-linear unsteady problem

Considering together the methods of the previous subsections, we eventually write the relaxed Picard algorithm for the unsteady problem. One iteration of the algorithm for the space- and time-discretized problem writes:

$$\left(\frac{\mathsf{M}}{\Delta t} + \mathsf{L} + \mathsf{N}(\boldsymbol{x}^{k+1}_{[i]})\right)\tilde{\boldsymbol{x}}^{k+1}_{[i+1]} = \frac{\mathsf{M}\boldsymbol{x}^k}{\Delta t} + \boldsymbol{b} \tag{10.12}$$

$$\boldsymbol{x}^{k+1}_{[i+1]} = \omega\tilde{\boldsymbol{x}}^{k+1}_{[i+1]} + (1-\omega)\boldsymbol{x}^{k+1}_{[i]} \qquad \forall (k,i) \in [1, n_{\texttt{ITMA}}] \times [1, n_{\texttt{NITER}}] \tag{10.13}$$

M is a (mass) matrix related to the operator $\frac{\partial}{\partial t}$, L is a matrix related to the linear operators and N is a matrix related to the linearization of the non-linear operators. At the end, it will be necessary to solve $n_{\texttt{ITMA}} \times n_{\texttt{NITER}}$ linear systems. The complete algorithm is given in table 10.3.

In the Cast3M code, algorithm 10.3 is translated straightforwardly using the Gibiane language. The implementation is done in the `EXEC` procedure whose most significant part is given in listing 10.1.

10.2.5 Adequate choice of the important parameters

In practice, the `EXEC` procedure allows one to solve all types of problems discussed in this section: linear or non-linear, steady or unsteady. Table 10.4 summarizes the possible choices for the four important parameters of the algorithm: time-step value Δt, number of time steps $n_{\texttt{ITMA}}$, number of non-linear iterations $n_{\texttt{NITER}}$ and relaxation factor ω.

The most common and complex case is the unsteady non-linear problem case for which the four parameters have to be chosen adequately. In practice, it is mandatory to check that at every time step the relaxed Picard iteration converges, i.e. that the problem non-

```
'DEBPROC'  EXEC ;
'ARGUMENT' rv*'TABLE   ' ;
omeg = rv . 'OMEGA' ;
itma = rv . 'ITMA' ;
*
* Itérations en temps
*
'REPETER' bloc1 itma ;
*
* Itérations internes (non-linéarités)
*
   'REPETER' bloci (rv . 'NITER') ;
      sf mau  = 'KOPS' 'MATRIK' ;
*
* Construction de la matrice et du second membre au pas de temps &bloc1
* a l'itération interne &bloci
*
      'REPETER' bloc2 ('DIME' (rv . 'LISTOPER')) ;
         nomper = 'EXTRAIRE' &bloc2 (rv . 'LISTOPER') ;
         notable= 'CHAINE' &bloc2 nomper ;
         msi mai= ('TEXTE' nomper) (rv . notable) ;
         mau = mau 'ET' mai ;
         sf  = sf  'ET' msi ;
      'FIN' bloc2 ;
      s1 = rv . 'CLIM' ;
*
* Résolution du système lineaire
*
      res = 'KRES'  ma1 'TYPI' (rv . 'METHINV')
                      'CLIM' s1 'SMBR' s2 ;
*
* Calcul de l'erreur
*
      eps = 'TCRR' res omeg (rv . 'INCO') ;
   'FIN' bloci ;
*
* Mise a jour des inconnues
*
    irt = 'TCNM' rv ;
    'SI' ('EGA' irt 1) ;
       'MESSAGE' ' Temps final atteint : ' (rv . 'PASDETPS' . 'TPS') ;
       'QUITTER' bloc1 ;
    'FINSI' ;
'FIN' bloc1 ;
***********************    E X E C ************************************
'FINPROC' ;
```

Listing 10.1: Most significant part of the procedure `exec.procedur` corresponding to algorithm 10.3.

Problem type	Δt	$n_{\mathtt{ITMA}}$	$n_{\mathtt{NITER}}$	ω
Steady Linear	–	1	1	1.0
Unsteady Linear	∘	∘	1	1.0
Steady Non-Linear	–	1	∘	$0.1 < \circ \leq 1.$
Unsteady Non-Linear	∘	∘	∘	$0.4 < \circ \leq 1.$

Table 10.4: Choice of the important parameters of the EXEC procedure for solving fluid dynamics problems. –: not prescribed. ∘: to be chosen adequately.

linearity is correctly resolved. If this is not the case (for example, when $\|\boldsymbol{u}_{[i+1]} - \boldsymbol{u}_{[i]}\|$ does not decrease), two remedies are possible:

1. decrease ω (as seen in section 10.2.3), but then, it is necessary to increase $n_{\mathtt{NITER}}$ in inverse proportion. Practically, we do not recommend $\omega < 0.4$. And for $\omega = 0.4$, a typical value for $n_{\mathtt{NITER}}$ would be 5.

2. decrease Δt. In this case, $n_{\mathtt{ITMA}}$ will increase in inverse proportion if one is to reach the same final time value t_∞.

In general, it is more efficient to decrease the time step Δt than to decrease ω. Indeed, decreasing Δt is likely to enhance the convergence of the non-linear iteration (better spectral properties of the linear system's matrix and initial guess $\boldsymbol{x}_{[0]}^{k+1}$ closer to the sought solution $\boldsymbol{x}_{[\infty]}^{k+1}$) but also the temporal precision of the approximate solution is enhanced.

The adequate choice of a time step is still part of the art of the engineer. Some automatic method for choosing the time step are available but they are not always satisfactory. The adequate choice of a time step frequently involves a preliminary physical analysis of the problem (dimensional analysis…) or an a posteriori analysis of the approximate numerical solution (Is there a large variation in a time step ? Is the convergence in the non-linear loop slow or fast ?).

If it is still difficult to converge in the non-linear iterations loop, one might have to modify some parameters outside the resolution algorithm which also play an important role:

1. the upwinding options for the convective terms (see chapter 4 and the 'KONV' operator in table 10.2);

2. the mesh: it should be chosen so that the sought solutions can be correctly approximated in space.

10.3 Summary

In this chapter we have briefly described how to solve a fluid dynamics problem with the CastƐM code. This is generally done in four steps:

1. creation of the mesh (part of the spatial discretization process);

2. description of the problem to be solved with the 'EQEX' operator;

3. resolution of the problem with the EXEC procedure;

4. post-treatment and analysis of the results.

Only steps 2 and 3 were dealt with in this chapter, the other two will be seen during the tutorial sessions. One can refer to the complete data files used in these lecture notes as examples. They are available on the Cast3M Web site [Cas].

The solution method implemented in the `EXEC` procedure is a relaxed fixed-point algorithm. The parameters of the algorithm, to be adequately chosen by the user are: the time-step value Δt, the number of time steps $n_{\mathtt{ITMA}}$, the number of non-linear iterations $n_{\mathtt{NITER}}$ and the relaxation factor ω. In particular, the correct choice of the time-step is of paramount importance.

Bibliography

[BAH87] Robert Byron Bird, Robert C. Armstrong, and Ole Hassager. *Dynamics of Polymeric Liquids*, volume 1. Fluid Mechanics. Wiley Interscience, 2nd edition, 1987.

[BL05] Pavel Bochev and R. B. Lehoucq. On the Finite Element Solution of the Pure Neumann Problem. *SIAM Review*, 47(1):50–66, 2005.

[Can01] Sébastien Candel. *Mécanique des fluides*. Dunod, 2nd edition, 2001.

[Cas] Cast3M. Web site. `http://www-cast3m.cea.fr/`.

[DLT12] Gouri Dhatt, Emmanuel Lefrançois, and Gilbert Touzot. *Finite Element Method*. Wiley-ISTE, October 2012.

[DP00] Frédéric Dabbène and Henri Paillère. Initiation à la simulation numérique en mécanique des fluides : éléments d'analyse numérique. Technical Report SEMT/LTMF/RT/00–015/A, CEA, 2000. Cours B2-1 ENSTA, `http://www-cast3m.cea.fr/`.

[EG02] Alexandre Ern and Jean-Luc Guermond. *Éléments finis: théorie, applications, mise en œuvre*. Springer, 2002.

[EG04] Alexandre Ern and Jean-Luc Guermond. *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematical Series*. Springer, New York, 2004.

[ESW14] Howard Elman, David Silvester, and Andy Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, second edition, 2014.

[Gur81] Morton E. Gurtin. *An Introduction to Continuum Mechanics*, volume 158 of *Mathematics in Science and Engineering*. Academic Press, 1981.

[Hug87] T.J.R Hughes. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations. *IJNMF*, 7(11):1261–1275, 1987.

[Lev02] Randall J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, first edition, 2002. see also `http://amath.washington.edu/~rjl`.

[SF88] Gilbert Strang and George J. Fix. *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, 2nd edition, 1988. `http://math.mit.edu/~gs`.

[Str07] Gilbert Strang. *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007. See also `http://math.mit.edu/cse`.