Prof. Santiago Badia ©

Lecture Notes on

Numerical Methods for Partial Differential Equations

v0.0.2, July 13 2023

Preface

I have created these notes for the unit MTH3340 *Numerical Methods for Partial Differential Equations*. In these lecture notes, I provide the mathematical background of finite element methods. Finite element methods rely on weak forms of partial differential equations grounded on functional analysis. Likely, many students will not have much (or any) background in functional analysis. One option would be to skip these concepts and go straight to the discrete problem without discussing well-posedness at the continuous level. However, functional spaces cannot be skipped if we want to understand the convergence properties of these methods. So, I have decided to start this presentation with a *very* short and simple introduction to variational methods and functional analysis. This chapter should be accessible to students with some mathematical background. Next, I introduce the Galerkin method, analyse the error of this approach, and combine it with spectral and finite element methods in one dimension. The last section extends the finite element method to multiple dimensions, showing the machinery needed in practical implementations. I have also included some *advanced* material that is not part of the unit assessment. I have marked this material with a section title in red. Interested students who want to explore the mathematical theory of the method in more detail can read these sections. Besides, some proofs of theorems have also been flagged as *(advanced)*, and students do not need to understand them.

We will combine the theory with computational tutorials. They can be found in this `Github` repository. I provide instructions for a local installation of the tools used in the tutorial. The tutorials use the `Gridap` software library developed by co-workers and me. This library provides tools for the numerical approximation of partial differential equations using mesh-based techniques (finite element methods in general). The tutorials and the library are written in `Julia`, a programming language that combines the

expressiveness of dynamic languages like `Python` and the performance of static languages like `C++` or `FORTRAN`. `Julia` is an exciting fresh language for numerical implementations in computational and data science.

Computational mathematics is a fantastic research field at the core of computational science. It is the third pillar of science, together with experiments and theory. Finite element methods (in a broad sense) are in the core of most state-of-the-art research on numerical partial differential equations. They have applications in almost any scientific discipline since many mathematical models are multi-dimensional differential equations on complex geometries. This unit provides a broad picture of the field. Please contact me if you want more information about these techniques or research topics in this field.

It has been a formidable task to select suitable topics, how to present them, typesetting the equations in latex, and create illustrations, together with tutorials, exercises, and computational tutorials. I am positive I have made some mistakes in the process. Please, inform me of typos you find and any other feedback related to this material. It will help me to improve these notes for future students. In any case, I will continuously go through this material, fix and improve things, and update them at Moodle accordingly. Try to stick to the latest version of the lecture notes indicated on the front page.

Santiago Badia

Melbourne, July 15, 2022

Contents

Chapter 1

Mathematical models

1.1 Introduction

In this chapter, we will start with the statement of some mathematical models in physics that can be defined as the minimisation of a potential (energy). Then, using the calculus of variations, we will end up with its corresponding variational form. Next, the existence and uniqueness of solutions, together with the equivalence between the minimisation and variational statements, will be provided. Finally, under extra regularity assumptions, we will relate these formulations with the standard *strong* partial differential equation form. We will also discuss how we can go in the opposite direction, starting from a partial differential equation in strong form and ending up with a *weak* or variational equation or minimisation problem.

The existence issue in finite-dimensional problems is straightforward. However, for infinite-dimensional function spaces, it involves advanced mathematics; more specifically, it involves *functional analysis*. However, we will try to reduce the exposition to the minimum while keeping a rigorous presentation. In this sense, we will provide some intuitive examples showing how important it is to choose suitable functional spaces to get well-posed minimisation and variational problems. We will introduce the concepts of Banach and Hilbert spaces, Cauchy sequences, completeness, and *maximal function space* concerning a given norm. With this idea, we can naturally define some useful Lebesgue and Sobolev spaces and provide some critical properties of these spaces (without proof).

As we will see in the next chapter, finite element and (some) spectral

methods rely on weak formulations of partial differential equations. Presenting these numerical methods without a thorough introduction to these problems and their relation with partial differential equations in the classical sense is not much satisfactory for maths-oriented students. This is a big difference compared to finite difference (and other collocation) methods that instead work with the strong form.

1.1.1 Minimisation problem

Let us consider a mathematical model representing a straight bar under traction (i.e., a bar in which external forces can only be in the longitudinal directions). In this case, the physical domain that represents the bar is an interval $[a, b] \subset \mathbb{R}$ and we define the longitudinal (or tangential) displacement of the bar particles in the original configuration with $u(x)$. Let us consider that the displacements of the bar particles are very small compared to the bar length $|b-a|$, i..e, the so-called *small displacement* assumption. Under these circumstances, the elastic energy stored by the bar is expressed by Hooke's law as

$$E(u) = \frac{1}{2}\int_a^b \kappa(x)u'(x)^2 \mathrm{d}x,$$

where $\kappa(x)$ is the Young's modulus of the bar material, which is assumed to be in the *elastic* regime. u' represents the derivative of u with respect to x. Let us assume the following *boundary conditions*, i.e., a prescribed longitudinal displacement at the end-points:

$$u(a) = u_a, \qquad u(b) = u_b, \tag{1.1}$$

where $u_a, u_b \in \mathbb{R}$, and a *forcing* term $f : [a, b] \to \mathbb{R}$. u_a, u_b and f are *data*. Due to the *minimum energy principle*, the displacement of the bar is the function that minimises the following functional:

$$J(u) \doteq \int_a^b (\frac{1}{2}\kappa(x)u'(x)^2 - f(x)u(x))\mathrm{d}x. \tag{1.2}$$

An elastic cord with a perpendicular load $f(x)$ under the assumption of the small displacements satisfies the same problem; in this case $u(x)$ represents the displacement in the perpendicular direction. An analogous minimisation problem is obtained when modelling the heat conduction on a bar. In this

case, $u(x)$ is the temperature, $\kappa(x)$ is the heat conductivity, and $f(x)$ is the *source* term.

In any case, we have not stated yet the minimisation problem for the functional (1.2), since we have not defined yet in which set of functions we want to minimise it. Since we need to evaluate first derivatives of the function in (1.2), it is natural to consider that $u(x) \in C^1([a,b])$. Thus, the solution $u(x)$ reads as:

$$u \doteq \underset{v \in C^1([a,b]) \text{ satisfying (1.1)}}{\operatorname{argmin}} J(v). \tag{1.3}$$

Thus, using physical principles (energy minimisation), we can state mathematically physical problems as minimisation problems in *infinite dimensional* space of functions.

1.1.2 Variational formulation

In the previous subsection, we have ended up with a minimisation problem on an infinite-dimensional space of functions. In particular, we are interested on *vector spaces* of functions under addition and multiplication by a real number.

Definition 1.1.1: Vector space

A vector space V of real-valued functions in a domain $\Omega \subset \mathbb{R}^d$ (where d denotes the space dimension) is such that, for any $u, v \in V$ and $\alpha \in \mathbb{R}$, it holds

$$(u+v)(x) \doteq u(x) + v(x), \quad (\alpha \cdot u)(x) \doteq \alpha u(x), \quad \forall x \in \Omega.$$

Using *calculus of variations*, we can state the minimisation problem (1.3) in a *variational* form. Assuming the existence of a global minimum $u(x)$ for (1.3), we can now consider the variation of the functional J at u with respect to a variation $v \in C_0^1([a,b])$ times $\alpha \in \mathbb{R}$. $C_0^1([a,b])$ is the subspace of functions $C^1([a,b])$ that vanish at the end-points, i.e.,

$$C_0^1([a,b]) \doteq \{v \in C^1([a,b]) \ : \ v(a) = v(b) = 0\}.$$

Without the zero boundary conditions for the perturbation, the perturbed function $u + \alpha v$ would not satisfy the boundary conditions (1.1). Since u is

a global minimiser, it naturally holds

$$J(u) \leq \Phi_v(\alpha) \doteq J(u + \alpha v) \quad \forall v \in C_0^1([a,b]), \ \forall \alpha \in \mathbb{R}.$$

Thus, $\Phi_v(\alpha)$ has a global minimum at 0. If Φ_v is differentiable, $\Phi_v'(0) = 0$. Since u is the minimum energy configuration, any perturbation of u cannot produce a decrease of energy.

The derivative of Φ reads

$$\Phi_v'(0) = \lim_{\alpha \to 0} \frac{J(u + \alpha v) - J(u)}{\alpha}.$$

Such derivative is the so-called *directional derivative* of J at u in the direction v. Enforcing $\Phi_v'(0)$ to be zero for any perturbation $v \in C_0^1([a,b])$, we get, under simple algebraic manipulations and eliminating high order terms, that

$$\int_a^b (\kappa(x)u'(x)v'(x) - f(x)v(x))\mathrm{d}x = 0 \quad \forall v \in C_0^1([a,b]). \tag{1.4}$$

This expression is the *weak or variational form* of the physical problem at hand. In the field of statics, this statement of the elastic problems above is called the *principle of virtual work*, which means that any perturbation of the equilibrium configuration requires energy supply to the system.

1.1.3 Differential equation

We would like to relate the variational and minimisation formulations above with standard differential equations. In this process, we will observe that, to reach this form, we will need to make some additional regularity assumptions over the solution.

Lemma 1.1.2: Fundamental lemma of calculus of variations

If a function $f \in C^0([a,b])$ is such that

$$\int_a^b f(x)v(x)\mathrm{d}x = 0, \ \forall v \in C_0^0([a,b]),$$

then $f \equiv 0$.

Let us assume that the solution u of the variational formulation (1.4) is in $C^2([a,b])$, $\kappa \in C^1([a,b])$, and $f \in C^0([a,b])$. Using integration by parts, we get for any $v \in C_0^0([a,b])$

$$\int_a^b \kappa(x)u'(x)v'(x)\mathrm{d}x - \int_a^b f(x)v(x)\mathrm{d}x = -\int_a^b ((ku')' - f(x))v(x)\mathrm{d}x.$$

Lemma 1.1.2 leads to

$$-(\kappa u')' = f \quad \text{in } [a,b], \quad u(a) = u_a,\ u(b) = u_b. \tag{1.5}$$

As a result, if the solution of (1.4) with the boundary conditions in (1.1) is in $C^2([a,b])$, it satisfies the two-point boundary value problem (1.5).

1.2 Abstract setting

This section considers an abstract setting for minimisation and variational problems, including the examples above. Under regularity assumptions, we show how to find their corresponding boundary value problem. We consider minimisation problems that are related to an energy functional J on a space of functions $\Omega \subset \mathbb{R}^d \to \mathbb{R}$, where d is the space dimension, e.g., $d = 2, 3$. Ω is assumed to be bounded, i.e.,

$$\mathrm{diam}(\Omega) = \sup_{x,y\in\Omega} \|x - y\| < \infty,$$

and its boundary $\partial\Omega$ is piecewise smooth, i.e. it can be expressed as a smooth diffeomorphism from a reference polyhedron boundary.

Definition 1.2.1: (Bi)linear forms

Given a vector space V in the field of scalars $\mathbb{R}$, a form $\ell : V \to \mathbb{R}$ is linear if it satisfies:

$$\ell(u+v) = \ell(u) + \ell(v),\ u,\ v \in V, \quad \ell(\alpha u) = \alpha\ell(u)\ \forall u \in V,\ \alpha \in \mathbb{R}.$$

A form $a : V \times V \to \mathbb{R}$ is a bilinear form if it is linear with respect to each of its two arguments separately.

1.2.1 Abstract minimisation

We are interested in an important class of minimisation problems that covers the abovementioned examples.

Definition 1.2.2: Quadratic minimisation problem

Let us consider a functional $J : V \to \mathbb{R}$ on a vector space V that can be expressed as

$$J(v) \doteq \frac{1}{2}a(v,v) - \ell(v) + c$$

for a symmetric bilinear form $a : V \times V \to \mathbb{R}$, a linear form $\ell : V \to \mathbb{R}$ and $c \in \mathbb{R}$. The problem

$$u = \operatorname*{argmin}_{v \in V} J(v)$$

is called a quadratic minimisation problem *on V.*

We note that the space of functions with non-homogeneous boundary conditions is not a vector space but an affine space. The minimisation in an affine space can easily be transformed into a vector space using the *offset function* method. We consider a vector space of functions V_0 and an affine space V that can be expressed as $u_0 + V_0$, where u_0 is the so-called *offset function*. In the examples in Section 1.1, u_0 is a function in $C^1([a,b])$ that satisfies the boundary conditions (1.1) (while keeping a desired level of smoothness), whereas V_0 is a space of functions with homogeneous (zero) boundary conditions, i.e., $V_0 = C_0^1([a,b])$. Given J, V_0 and u_0, the minimisation problem in V can be expressed as follows. We have

$$\begin{aligned} J(v+u_0) &= \frac{1}{2}a(v+u_0, v+u_0) - \ell(v+u_0) + c \\ &= \frac{1}{2}a(v,v) + a(v,u_0) - \ell(v) + \frac{1}{2}a(u_0,u_0) - \ell(u_0) + c \doteq \tilde{J}(v) \end{aligned}$$

where $\tilde{J}$ is a quadratic functional too. Thus,

$$\operatorname*{argmin}_{u \in u_0 + V_0} J(u) = u_0 + \operatorname*{argmin}_{v \in V_0} \tilde{J}(v).$$

As a result, we can restrict to quadratic minimisation problems on vector spaces without loss of generality. Now, let us consider the well-posedness of the quadratic minimisation problem.

Definition 1.2.3: Positive definiteness

A symmetric bilinear form $a : V_0 \times V_0 \to \mathbb{R}$ *on a real vector space* V_0 *is* semi-positive definite *if*

$$a(u,u) \geq 0 \quad \forall u \in V_0.$$

Moreover, it is positive definite *if*

$$a(u,u) > 0 \quad \forall u \in V_0 \setminus \{0\}.$$

Lemma 1.2.4: Necessary condition for existence of a global minimum

If the quadratic minimisation problem in Definition 1.2.2 has a solution, then its bilinear form $a : V_0 \times V_0 \to \mathbb{R}$ *must be semi-positive definite.*

Proof. If the bilinear form is not semi-positive definite, we can pick a $u \in V_0$ such that $a(u,u) < 0$. Thus, $J(\alpha u) = 1/2\alpha^2 a(u,u) - \alpha\ell(u) + c$ and $\lim_{\alpha\to\infty} J(\alpha u) = -\infty$. □

Lemma 1.2.5: Necessary condition for uniqueness of the global minimum

If the quadratic minimisation problem in Definition 1.2.2 has a solution and its bilinear form a *is positive definite, then the solution is unique.*

Proof. Let us assume that there exist two solutions $u,\ v \in V_0$ such that $u \neq v$. $\Phi(\alpha) \doteq J(\alpha u + (1-\alpha)v)$ has two distinct global minima at $\alpha = 0$ and $\alpha = 1$. Besides, $\Phi(\alpha) = \alpha^2/2a(u-v,u-v) +$ lower order terms and $a(u-v,u-v) > 0$. Thus, Φ is a non-degenerate parabola that opens up and can only have a minimum at its vertex. It proves the result by contradiction. □

If a is semi-positive definite, there are infinite solutions that can only differ in an element of the kernel of a, i.e., two solutions $u,\ v \in V_0$ must

satisfy $a(u-v, u-v) = 0$.

Let us introduce some additional concepts that will allow us to prove more necessary conditions for existence.

Definition 1.2.6: Norm on a vector space

A norm $\|\cdot\|$ *on a vector space* V *is a map* $\|\cdot\| : V \to \mathbb{R}_+$ *such that*

- *Positive definiteness:* $\|v\| = 0 \iff v = 0 \; \forall \in V$,
- *Absolutely homogeneous:* $\|\alpha v\| = |\alpha|\|v\|, \; \forall \alpha \in \mathbb{R}, \; \forall v \in V$,
- *Triangle inequality:* $\|v + w\| \leq \|v\| + \|w\| \forall v, \; w \in V$.

Definition 1.2.7: Positive definite bilinear form

A symmetric positive definite bilinear form $a : V \times V \to \mathbb{R}$ *induces the* energy norm

$$\|u\|_a \doteq a(u,u)^{1/2}.$$

Definition 1.2.8: Continuity of (bi)linear forms

Given a vector space V *with norm* $\|\cdot\|$, *a linear form* $\ell : V \to \mathbb{R}$ *is* continuous *or* bounded *if*

$$\exists C > 0 \; : \; |\ell(v)| \leq C\|v\| \; \forall v \in V.$$

A bilinear form $a : V \times V \to \mathbb{R}$ *is* continuous *if*

$$\exists K > 0 \; : \; |a(u,v)| \leq K\|u\|\|v\|, \; \forall u, \; v \in V.$$

We note that continuity of the inner product that provides the energy norm can readily be checked for $K = 1$ by using the Cauchy-Schwarz inequality.

It is essential to prove that the potential J is bounded from below to have a well-posed minimisation problem.

Lemma 1.2.9: Boundedness from below

The quadratic functional J in Definition 1.2.2 with a positive definite bilinear form a is bounded from below *in V_0 if and only if the linear form ℓ is continuous in V_0 with respect to the energy norm $\|\cdot\|_a$.*

Proof. If a is positive definite and ℓ is continuous, using the generalised Young's inequality

$$ab \leq \frac{a^2}{2\epsilon} + \frac{\epsilon b^2}{2}, \ a, \ b \in \mathbb{R}, \ \epsilon > 0,$$

we readily get

$$J(u) = 1/2a(u,u) - \ell(u) \geq 1/2\|u\|_a^2 - C\|u\|_a \geq -1/2C^2.$$

In the other direction, let us assume that J is bounded below and the conditions in the lemma do not hold. For every $n \in \mathbb{N}$, we can pick $u_n \in V_0$ such that

$$\ell(u_n) \geq n\|u_n\|_a.$$

By re-scaling $u_n \leftarrow \frac{u_n}{\|u_n\|_a}$ we can assume that $\|u_n\|_a = 1$, thus

$$J(u_n) \leq 1/2 - n \to -\infty, \ \text{as} \ \ n \to \infty.$$

Thus, J cannot be bounded below, leading to a contradiction. It proves the result. □

So far, we have found necessary conditions for existence and uniqueness of solutions. Assuming that the vector space V_0 is finite dimensional, existence can readily be proven; the quadratic minimisation functional in finite dimension is a non-degenerate parabola opening up, for which there is a unique global minimum at its vertex.

Theorem 1.2.10: Existence and uniqueness of a minimiser in finite dimension

If the vector space V_0 in the quadratic minimisation problem in Definition 1.2.2 involves a positive definite symmetric bilinear form a and

a continuous functional ℓ, and V_0 has finite dimension, there exists a unique solution for this problem.

Proof. If the vector space V_0 has a finite dimension, we can define an ordered basis for it, and thus V_0 is isomorphic to $\mathbb{R}^N$. The variational form of the quadratic minimisation problem can be recast as a *square* linear system of equations with a positive-definite system matrix (thus non-singular) and right-hand side with bounded entries. Therefore, the unique solution equals the inverse of that matrix times the right-hand side vector. □

More details on the concepts in the previous proof can be found later, when we discuss finite dimensional discretisations of infinite dimensional problems. Unfortunately, sufficient conditions for existence in infinite dimensions are more elusive and are strongly related to the right choice of the vector space V_0. It must be *large enough* for existence but *small enough* for uniqueness.

1.2.2 Abstract variational form

In this section, we provide an abstract definition of variational forms. In the most general case, a linear variational problem is stated as follows.

Definition 1.2.11: Variational problem

A variational problem reads as:

$$u \in V \ : \ a(u,v) = \ell(v), \quad \forall v \in V_0$$

where V_0 is a vector space of functions, V is an affine space of functions, $a : V \times V_0 \to \mathbb{R}$ is a bilinear form and $\ell : V_0 \to \mathbb{R}$ is a linear form.

In Definition 1.2.11, the space V in which we seek the solution is the *trial space* (affine space) and the space V_0 of admissible test functions is the *test space* (vector space). Variational problems can also be stated in terms of vector spaces by using the fact that $V = u_0 + V_0$ for an arbitrary *offset function* $u_0 \in V$. The abstract variational problem in Definition 1.2.11 can

be written as:

$$\tilde{u} \in V_0 : \ a(\tilde{u}, v) = \ell(v) - a(u_0, v) \ \ \forall v \in V_0, \quad u = \tilde{u} + u_0.$$

Thus, we can consider variational problems for trial and test spaces without loss of generality.

Let us consider now the relationship between minimisation and variational problems. Let us restrict ourselves to quadratic minimisation problems. Let us *assume the existence* of a global minimiser u. Using the calculus of variations (as described above), we can compute the directional derivative of the functional as

$$\begin{aligned} \lim_{\alpha \to 0} \frac{J(u + \alpha v) - J(u)}{\alpha} &= \lim_{\alpha \to 0} \frac{\alpha a(u, v) + \frac{\alpha^2}{2} a(v, v) - \alpha \ell(v)}{\alpha} \\ &= a(u, v) - \ell(v), \quad \forall v \in V_0. \end{aligned}$$

Enforcing that such derivative is equal to zero because u is a minimum of the functional, we end up with a *linear* variational problem. As a result, *the global minimiser u of a quadratic function is the solution of a linear variational problem.*

1.2.3 Well-posedness

In the previous sections, we have proved some necessary conditions for existence of solutions of a quadratic minimisation problem and, in turn, its corresponding linear variational problem. Whereas existence is not complicated in finite dimensional vector spaces, the situation is far more subtle in infinite dimensions. Let us start with a necessary condition for existence. If a global minimiser u exists for the quadratic minimisation problem, its energy norm $\|u\|_a$ must be bounded. It leads to the following necessary condition.

Corollary 1.2.12: Necessary continuity of the linear form

If there is a global minimiser for the quadratic minimisation problem in Definition 1.2.2 with a symmetric positive definite bilinear form a, then the linear form ℓ must be continuous.

Proof. Any minimiser $u \in V_0$ of the quadratic minimisation problem satisfies the variational problem in Definition 1.2.11. If the linear form in the quadratic minimisation problem is continuous and the bilinear form is symmetric positive definite, it holds:

$$|\ell(v)| = |a(u,v)| \leq \|u\|_a \|v\|_a \leq C\|v\|_a,$$

for $C \doteq \|u\|_a < \infty$, where we have used the Cauchy-Schwarz inequality for the energy norm. □

The trial space should be large enough to find a solution but if the space is *too small*, existence of solution will not hold. The following example shows the issue.

Example (advanced) 1.2.13: Non-existence of positive definite quadratic minimisation problems

Let us consider the positive definite quadratic minimisation problem

$$J \doteq \int_0^1 \frac{1}{2}u^2(x) - u(x)\mathrm{d}x = \frac{1}{2}\int_0^1 (u(x)-1)^2 - 1\mathrm{d}x,$$

and seek the global minimiser in $C_0^0([0,1])$. It can be cast in the abstract linear variational form with

$$a(u,v) \doteq \int_0^1 u(x)v(x)\mathrm{d}x, \quad \ell(v) \doteq \int_0^1 v(x)\mathrm{d}x.$$

Let us assume that $u \in V_0$ is the global minimiser of J in V_0. Now, let us consider

$$\Phi_u(x) \doteq \min\{1, 2\max\{u(x), 0\}\}, \ x \in [0,1]$$

After some algebraic manipulations, taking into account that J penalises the distance between $u(x)$ and 1, one can check that $J(\Phi_u) < J(u)$ unless $u \equiv 1$, which is not possible since $1 \notin C_0^0([0,1])$. Thus, u cannot be the global minimiser and we cannot get a minimum in $C_0^0([0,1])$.

We can create a sequence $\{u_n\}_{n\in\mathbb{N}}$ with $u_0 \doteq u \in V_0$ and $u_{n+1} \doteq \Phi_{u_n}$. Thus, $J(u_{n+1}) < J(u_n)$ but the sequence has no limit in $C_0^0([0,1])$. The problem strives in the boundary conditions. If we would seek a solution in $C([0,1])$, we would just take $u(x) = 1$, but we cannot.

In the next section, we will show how to *complete* functional spaces so that one can prove the existence of solutions for a given quadratic minimisation problem.

1.3 Functional spaces

The motivation of this section is to define functional spaces for which the variational (or minimisation) problems above have solutions. We present here some functional spaces that solve the question of existence for some simple quadratic minimisation functionals. The broad idea is to define the largest space of functions for which the bilinear form *a has sense* and satisfies *suitable* boundary conditions.

As commented above, a minimiser $u : \Omega \to \mathbb{R}$ of the quadratic minimisation problem must have bounded energy, i.e., $a(u,u) < \infty$. Thus, we define the space V in terms of J as follows:

$$V \doteq \{v : \Omega \to \mathbb{R} \ : \ a(v,v) < \infty\}. \tag{1.6}$$

It is the so-called *maximal functional space* on which J is defined.

1.3.1 The $L^2(\Omega)$ space

Let us consider the potential

$$J_0(u) \doteq 1/2 \int_\Omega |u(x)|^2 \mathrm{d}x.$$

The *maximal functional space* with respect to J_0 leads to

$$V_0 \doteq \{v : \Omega \to \mathbb{R} : \int_\Omega |v(x)|^2 \mathrm{d}x < \infty\}. \tag{1.7}$$

Definition 1.3.1: $L^2(\Omega)$ space

The function space (1.7) *is the space of* square-integrable *functions on* Ω*, which is represented with* $L^2(\Omega)$*. It is a normed space for*

$$\|v\|_0 \doteq \|v\|_{L^2(\Omega)} \doteq \left(\int_\Omega |v(x)|^2 \mathrm{d}x\right)^{1/2}.$$

We note that boundary values of $L^2(\Omega)$ functions are ill-posed.

Example (advanced) 1.3.2: Boundary conditions cannot be imposed in $L^2(\Omega)$

Let us consider a function in $u \in C^0([0,1]) \subset L^2((0,1))$*. Now, let us consider a sequence of perturbations* $\{\tilde{u}_n\}_{n\in\mathbb{N}}$ *of this function for arbitrary values* u_0 *and* u_1*:*

$$\tilde{u}_n \doteq \begin{cases} u(x) + (1-nx)(u_0 - u(0)), & 0 \le x \le 1/n, \\ u(x), & 1/n < x < 1 - 1/n \\ u(x) - n(1-1/n-x)(u_1 - u(1)), & 1 - 1/n < x \le 1. \end{cases}$$

Clearly $\tilde{u}_n(0) = u_0$*,* $\tilde{u}_n(1) = u_1$ *for any* $n \in \mathbb{N}$*. On the other hand, we obtain after integration* $\|\tilde{u}_n - u\|^2_{L^2((0,1))} = \frac{1}{3n}(u_0 - u(0) + u_1 - u(1)) \to 0$ *as* $n \to \infty$*.*
Thus, we can find functions arbitrary close to u *in the* L^2*-norm that satisfy whatever boundary condition on the boundary. Thus, the* L^2*-norm is not strong enough to* feel *the boundary conditions. It means that the space* $V \doteq \{u \in L^2((0,1)) \ : \ u(0) = u(1) = 0\}$ *is not a* closed *subspace of* $L^2((0,1))$*, i.e., we can find functions that are not in* V *but can be arbitrarily well approximated by functions in* V *(as in the example above). Another way to express this situation is saying that the linear operator that takes a solution in* $L^2((0,1))$ *and returns its value at one of the end-points is not continuous.*

Definition 1.3.3: Cauchy sequence

Consider a normed vector space V *equipped with the norm* $\|\cdot\|$*. A*

sequence $\{v_n\}_{n\in\mathbb{N}}$ *of elements of* V *is called a* Cauchy sequence *if*

$$\forall \epsilon > 0 : \exists n = n(\epsilon) \in \mathbb{N} \ : \ \|v_k - v_m\| \leq \epsilon, \ \forall k, m \geq n.$$

It is obvious to check that every convergent sequence is a Cauchy sequence. However, the equivalence between Cauchy and convergent sequences is only true for a particular type of spaces of paramount importance for the statement of well-posed variational problems.

Definition 1.3.4: Banach space

A normed vector space is called complete *if every Cauchy sequence converges. A complete normed vector space is called a* Banach space.

Definition 1.3.5: Inner product

An inner product $(\cdot,\cdot) : V \times V \to \mathbb{R}$ *in a vector space* V *is a symmetric positive definite bilinear form in* V.

Definition 1.3.6: Hilbert space

If a Banach space is endowed with a norm that is an energy norm with respect to a symmetric positive definite bilinear form, it is called a Hilbert space.

Let us list some examples of Banach and Hilbert spaces. The set of real numbers $\mathbb{R}$ with the modulus norm is complete (this is in fact an axiom, the *axiom of completeness*, which states that the real line has no gaps). Finite dimensional spaces are also complete.

We note that the definition of completeness depends not only on the set but the distance (norm) being used, i.e., in the metric space. For instance, the space $C^0(\Omega)$ for $\Omega \subset \mathbb{R}$ bounded endowed with the supremum norm $\|\cdot\|_\infty$, where $\|u\|_\infty \doteq \sup_{x\in\Omega} u(x)$, is complete. But we have seen that $C^0(\Omega)$ with the L^2 norm is not complete.

Intuitively, completeness tells us there are no *missing* points in our set (in the interior or boundary).

The space L^2 is complete. As stated in the definition, this is a normed space (the L^2 norm), and completeness can be proved with respect to this

norm. When we talk, e.g., about $L^2(\Omega)$, you have to understand that we are not just defining the set of functions but also the norm for these functions, which in turn provides a distance $d(u,v) = \|u-v\|_{L^2(\Omega)}$.

Completeness is the essential ingredient that is needed to prove the existence of a minimiser for the quadratic minimisation problem in Definition 1.2.2 (and thus, a solution of the variational problem in Definition 1.2.11). We provide more details for the interested reader in the following (advanced) section. Intuitively, we want to design a sequence of functions that get closer and closer to the functional infimum and be sure that this limit (the solution) does exist and belongs to our space.

1.3.2 Well-posedness in Hilbert spaces

Theorem 1.3.7: Existence and uniqueness of solutions in Hilbert spaces

Let us consider a Hilbert space V_0 endowed with the inner product $a : V_0 \times V_0 \to \mathbb{R}$ and a continuous linear functional $\ell : V_0 \to \mathbb{R}$. The corresponding quadratic minimisation problem

$$u = \underset{v\in V_0}{\operatorname{argmin}} J(v), \;\; J(v) \doteq 1/2a(v,v) - \ell(v)$$

has a unique solution.

Proof. By Lemma 1.2.9, the quadratic functional J is bounded below. Thus, we can define a sequence $\{v_n\}_{n\in\mathbb{N}}$ such that

$$|J(v_n) - \mu| \leq 1/n, \quad \mu \doteq \inf_{v\in V_0} J(v).$$

On the other hand, due to the bilinearity of a, we obtain

$$\begin{aligned}
&\frac{1}{2}(J(v) + J(w)) - J(1/2(v+w)) \\
&= 1/4\,(a(v,v) + a(w,w) - 2a(1/2(v+w), 1/2(v+w)) \\
&= 1/8\|v-w\|_a^2.
\end{aligned}$$

Clearly, $J(1/2(v+w)) \geq \mu$, which combined with the previous results

leads to

$$1/8\|v_k - v_m\|_a^2 \leq 1/2(J(v_k) + J(v_m)) - \mu$$
$$\leq 1/2(1/k + 1/m) \leq \max\{1/k, 1/m\}.$$

Thus, $\{v_n\}_{n\in\mathbb{N}}$ is a Cauchy sequence and

$$u \doteq \lim_{n\to\infty} v_n \in V_0$$

due to completeness of V_0. Since J is a continuous functional in V_0, we have that

$$J(u) = J(\lim_{n\to\infty} v_n) = \lim_{n\to\infty} J(v_n) = \mu.$$

Thus, $u \in V_0$ is a global minimiser of the problem at hand. Uniqueness is proved using Lemma 1.2.5. □

We can observe that the problem in Advanced Example 1.3.2 was ill-posed in $C^0([a,b])$ but it is well-posed in $L^2((a,b))$. The space $C^0([a,b])$ was *too small*; it did not include the limits of Cauchy sequences, i.e., not complete. In fact, a quadratic minimisation problem can always be *fixed* by *completing* the vector space, which means to augment the space by including also all the potential limits of Cauchy sequences in it.

Definition 1.3.8: Dense space

A subset $W \subset V$ is dense *in a normed vector space V if V is the union of W and all the limits of Cauchy sequences in W.*

Theorem 1.3.9: Completion of a normed vector space

For every normed space V_0 there is a unique complete vector space $\tilde{V}_0$ (up to isomorphisms) that contains V_0 as a dense subspace.

These results provide a constructive way to create well-posed minimisation problems. We start with an admissible space V_0 of functions with bounded energy (see (1.6)). Next, we consider the completion $\tilde{V}$ of V with respect to the energy norm. $\tilde{V}$ is complete by definition, and thus, the

problem is well-posed in $\tilde{V}$ by Theorem 1.3.7. For instance, in the case of Advanced Example 1.3.2, we should consider the completion of $C^0([0,1])$ with respect to $a(u,v) = \int_0^1 u(x)v(x)\mathrm{d}x$. On the other hand, we know that the space in which this problem is well-posed in $L^2((a,b))$. The following result give sense to these two observations.

Theorem 1.3.10: $L^2(\Omega)$ as completion of $C^0(\Omega)$

Given $\Omega \subset \mathbb{R}^d$, the completion of $C^0(\Omega)$ with respect to $\|\cdot\|_{L^2(\Omega)}$ is the space $L^2(\Omega)$.

1.3.3 The $H^1(\Omega)$ space

Definition 1.3.11: Gradient of a function

Given a function $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$, we define its gradient as $\boldsymbol{\nabla} f(x) \doteq [\partial f/\partial x_1(x), \dots, \partial f/\partial x_d(x)]^T \in \mathbb{R}^d$ for $x \in \Omega$.

We can now proceed analogously for the semi-positive definite bilinear form

$$a(u,v) \doteq \int_\Omega \boldsymbol{\nabla} u(x) \cdot \boldsymbol{\nabla} v(x)\mathrm{d}x.$$

The corresponding maximal functional set for the semi-norm endowed by this inner product reads

$$V_0 \doteq \{v : \Omega \to \mathbb{R} \ : \ v = 0 \text{ on } \partial\Omega, \ \int_\Omega |\boldsymbol{\nabla} v(x)|^2 \mathrm{dx} < \infty\}. \tag{1.8}$$

Definition 1.3.12: Sobolev space

A Sobolev space is a vector space of function endowed with a norm that combines L^p norms of the function itself and its derivatives up to a given order. The $L^p(\Omega)$ space, $1 \le p < \infty$, is the space of functions such that $\left(\int_\Omega |v|^p \mathrm{dx}\right)^{\frac{1}{p}} < \infty$.

Definition 1.3.13: Sobolev space $H_0^1(\Omega)$

The function space (1.8) *is the space of* square integrable *functions with* square integrable *gradients on* Ω *that vanish on* $\partial\Omega$*, which is represented with* $H_0^1(\Omega)$*. It is a normed space for*

$$|v|_{H^1(\Omega)} \doteq \left(\int_\Omega |\boldsymbol{\nabla} v(x)|^2 \mathrm{d}x \right)^{1/2}.$$

We note that we have started considering the space $H_0^1(\Omega)$, in which the zero subscripts means zero value on $\partial\Omega$. The control provided by the semi-norm $|\cdot|_{H^1(\Omega)}$ is enough to make boundary conditions meaningful.[1] For instance, if we consider Advanced Example 1.3.2, we can observe that $|u - \tilde{u}|_{H^1((0,1))} = n(u_0 - u(0) + u_1 - u(1)) \to \infty$ as $n \to \infty$. Thus, the H^1-semi-norm *feels* the boundary value, since it cannot be changed without changing the energy of the function.

Now, we want to consider more general boundary values, not just zero on $\partial\Omega$. We eliminate the boundary condition and add an additional term (the L^2 norm) to the semi-norm $|\cdot|_{H^1(\Omega)}$ to get the norm $\|\cdot\|_0$.

Definition 1.3.14: The $H^1(\Omega)$ space

The Sobolev *space*

$$H^1(\Omega) \doteq \{v \in L^2(\Omega) \ : \int_\Omega |\boldsymbol{\nabla} v(x)|^2 \mathrm{d}x < \infty\}$$

is a normed space with

$$\|v\|_{H^1(\Omega)}^2 \doteq \|v\|_0^2 + |v|_{H^1(\Omega)}^2.$$

$H^1(\Omega)$ is the maximal function space with respect to the norm $\|v\|_{H^1(\Omega)}$. We note that $|\cdot|$ is a semi-norm in $H^1(\Omega)$ because $|v| = 0$ for v a non-zero constant, violating the definiteness condition. However, it is a norm, e.g., in the subspace of functions in $H^1(\Omega)$ with zero mean value.

[1] Let us remark that $|\cdot|_{H^1(\Omega)}$ is in fact a norm for the space $H_0^1(\Omega)$ thanks to the homogeneous boundary conditions.

Example 1.3.15: Piecewise continuous functions in $H^1(\Omega)$

We observe that the space $H^1(\Omega)$ includes functions that do not posses classical derivatives, *i.e., they are not differentiable at all points. For instance, let us consider a* piecewise *function in* $[0,1]$

$$u(x) = \begin{cases} 2x, & 0 < x < 1/2, \\ 2(1-x), & 1/2 \leq x < 1. \end{cases}$$

This function belongs to $H^1(\Omega)$ since

$$|u|_{H^1(\Omega)} = \int_0^1 |\nabla u(x)|^2 \mathrm{d}x = 4 \leq \infty.$$

In general, it can be checked that the space $C^1_{\mathrm{pw}}(\bar{\Omega})$ of functions with piecewise continuous first derivatives is a subset of $H^1(\Omega)$.

1.3.4 $H^1(\Omega)$ by completion

Analogously as $L^2(\Omega)$, the $H^1(\Omega)$ space can be defined by completion.

Theorem 1.3.16: $H^1(\Omega)$ and $H^1_0(\Omega)$ by completion

Given a piecewise smooth domain Ω, the space $H^1(\Omega)$ is the completion of $C^\infty(\Omega)$ with respect to the norm $\|\cdot\|_{H^1(\Omega)}$. For a bounded domain $\Omega \subset \mathbb{R}^d$, $H^1_0(\Omega)$ is the completion of $C^\infty_0(\Omega)$ with respect to the semi-norm $|\cdot|_{H^1(\Omega)}$.

Thus, the space of smooth functions $C^\infty(\Omega)$ and $C^\infty_0(\Omega)$ are dense in $H^1(\Omega)$ and $H^1_0(\Omega)$, respectively.

1.4 A multidimensional problem

At this point, we are in position to consider the multi-dimensional version of the model problem in (1.3), which models the normal displacement u of a membrane under a normal external pressure f and prescribed displacement on the boundary; we consider $\kappa = 1$ and zero boundary conditions for simplicity. This problem is represented by the quadratic minimisation problem

with the functional

$$J(u) = 1/2 \int_\Omega |\boldsymbol{\nabla} u(x)|^2 - f(x)u(x)\mathrm{d}x.$$

Thus, the energy norm of this problem is $\|\cdot\|_a^2 \doteq a(u,u)$ for

$$a(u,v) \doteq \int_\Omega \boldsymbol{\nabla} u(x) \cdot \boldsymbol{\nabla} v(x)\mathrm{d}x,$$

whereas the corresponding linear form reads

$$\ell(u) \doteq \int_\Omega f(x)u(x)\mathrm{d}x.$$

Thus, we consider the maximal functional space for this energy, i.e., the space of functions with bounded energy, which we now know is $H^1(\Omega)$. The subspace of functions in $H^1(\Omega)$ that satisfy zero boundary conditions is $H_0^1(\Omega)$; thus, this will be the test space.

As proved in Corollary 1.2.12, the continuity of the linear form is necessary for the existence of a global minimiser. Assuming that $f \in L^2(\Omega)$, using the Cauchy-Schwarz inequality we readily get:

$$\int_\Omega f(x)v(x)\mathrm{d}x \leq (\int_\Omega |f(x)|^2)^{1/2}(\int_\Omega |v(x)|^2)^{1/2} = \|f\|_0\|v\|_0, \ v \in H_0^1(\Omega),$$

where $\|f\|_0 < \infty$. We note that the continuity must be in terms of the energy (semi-)norm $|\cdot|_{H^1(\Omega)}$ but the above result is bounded with respect to the norm $\|\cdot\|_0$. The following classical inequality solves this issue.

Theorem 1.4.1: First Poincaré-Friedrichs inequality

Given a bounded domain $\Omega \subset \mathbb{R}^d$, it holds

$$\|u\|_0 \leq \mathrm{diam}(\Omega)\|\boldsymbol{\nabla} u\|_0, \quad \forall u \in H_0^1(\Omega).$$

Proof (advanced). We can prove this result by relying on the fact that smooth functions in $C_0^\infty(\bar{\Omega})$ are dense in $H_0^1(\Omega)$. If the result holds for these smooth functions, it readily holds for $H_0^1(\Omega)$, using the definition of density; such strategy is common in functional analysis and is called

a *density argument*. We show the result for $d = 1$ for the sake of simplicity. Using the fundamental theorem of calculus, we have:

$$u(x) = u(0) + \int_0^x u'(s)\mathrm{d}s,\ 0 \leq x \leq 1, \forall u \in C^0(\bar{\Omega}).$$

Using the fact that $u(0) = 0$ for $u \in C_0^\infty(\bar{\Omega})$, we get, using the Cauchy-Schwarz inequality

$$\begin{aligned}\|u\|_0^2 = \int_0^1 \left|\int_0^1 u'(s)\mathrm{d}s\right|^2 \mathrm{d}x \leq \int_0^1 \left(\int_0^x 1\mathrm{d}s \cdot \int_0^1 |u'(s)|^2\mathrm{d}s\right)^2 \mathrm{d}x \\ \leq \|u'\|_0^2.\end{aligned}$$

□

It leads to the following result.

Corollary 1.4.2: Admissible forcing term

The linear functional $\int_\Omega f(x)u(x)\mathrm{d}x$ *with* $f \in L^2(\Omega)$ *is continuous in* $H_0^1(\Omega)$.

Corollary 1.4.3: $|\cdot|_{H^1(\Omega)}$ is a norm in $H_0^1(\Omega)$

The semi-norm $|\cdot|_{H^1(\Omega)}$ *is a norm in* $H_0^1(\Omega)$.

Proof. Due to the Poincaré-Friedrichs inequality in 1.4, we have that

$$\|u\|_{H^1(\Omega)} \leq (\mathrm{diam}(\Omega) + 1)|u|_{H^1(\Omega)}.$$

□

Now, let us consider non-homogeneous boundary conditions

$$u(x) = g(\boldsymbol{x}),\ \forall x \in \partial\Omega,$$

where $g : \partial\Omega \to \mathbb{R}$ is the boundary value to be prescribed on the boundary. Let us define the trial space $H_g(\Omega) \doteq \{v \in H^1(\Omega)\ :\ v = g\ \text{ on } \partial\Omega\}$. The

way we understand the equality in the boundary conditions is also weak, and the regularity assumptions over g and why this boundary condition has sense are out of the scope of the book. The interested student can look for *trace theorems in Sobolev spaces* for more information.

Taking the directional derivatives of the quadratic functional and enforcing them to be zero, we end up with the following variational formulation:

$$u \in H^1_g(\Omega) \ : \ \int_\Omega \boldsymbol{\nabla} u(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x = \int_\Omega f(x) v(x), \ \forall v \in H^1_0(\Omega).$$

As commented above, this problem can also be stated using the *offset function* method. Pick a function $u_0 \in H_g(\Omega)$, and re-state the problem as:

$$\begin{aligned} &\delta u \in H^1_0(\Omega) \ : \ \int_\Omega \boldsymbol{\nabla} \delta u(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x \\ &= \int_\Omega f(x) v(x) - \int_\Omega \boldsymbol{\nabla} u_0(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x, \ \forall v \in H^1_0(\Omega), \end{aligned} \tag{1.9}$$

and return $u = u_0 + \delta u$. It is obvious to check that the additional right-hand side term due to boundary conditions is continuous, since $u_0 \in H^1(\Omega)$.

Clearly, this problem is a linear variational problem with $V_0 \doteq H^1_0(\Omega)$,

$$a(u, v) \doteq \int_\Omega \boldsymbol{\nabla} u(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x,$$

and

$$\ell(v) \doteq \int_\Omega f(x) v(x) \mathrm{d}x - a(u_0, v)$$

for $f(x) \in L^2(\Omega)$ and $u_0 \in H^1_g(\Omega)$. It is obvious to check that ℓ is continuous in $H^1_0(\Omega)$. As a result, the energy norm of the problem is $\| \cdot \|_a \doteq | \cdot |_{H^1(\Omega)}$, which is in fact a norm in $H^1_0(\Omega)$ due to the first Poincare-Friedrichs inequality.

We note that the regularity $f(x) \in L^2(\Omega)$ is sufficient for well-posedness but not necessary. Instead, we could just consider $f : H^1_0(\Omega) \to \mathbb{R}$ to be a linear and continuous functional, i.e., $f(v) < \infty$ for any $v \in H^1_0(\Omega)$; the application of f to v can still be symbolically represented using integration, i.e., $f(v) = \int_\Omega f(x) v(x)$. The vector space of these bounded functionals is

the *dual* space of $H_0^1(\Omega)$ and is usually represented with $H^{-1}(\Omega)$. It is a Banach space (normed and complete) with norm

$$\|f\|_{H^{-1}(\Omega)} \doteq \sup_{v \in H_0^1(\Omega)} \frac{f(v)}{\|v\|_{H^1(\Omega)}}$$

We have the following significant result showing that *primal* and dual spaces are *isometrically isomorphic*.

Theorem 1.4.4: Riesz representation theorem

For a linear continuous functional $\ell : V \to \mathbb{R}$ in a real Hilbert space V endowed with the inner product $a(\cdot,\cdot)$ and corresponding norm $\|\cdot\|_a$, there exists a unique $u \in V$ such that

$$a(u,v) = \ell(v), \qquad \|u\|_a = \sup_{v \in V} \frac{\ell(v)}{\|v\|_a} \doteq \|\ell\|_{a'}.$$

Proof. The existence and uniqueness of a solution for the linear variational problem has already been proved above. Concerning the second part, the one related to the equivalence between the primal norm of the solution and the dual norm of the linear form, we proceed as follows. First, due to the Cauchy-Schwarz inequality, we have $|\ell(v)| = |a(u,v)| \leq \|u\|_a \|v\|_a$ and thus $\sup_{v \in V} \frac{\ell(v)}{\|v\|_a} \leq \|u\|_a$ for any $v \in V$. The supremum is attained since $\ell(u) = \|u\|_a^2$. It proves the theorem. □

Corollary 1.4.5: Well-posedness of a 2nd order elliptic problem

The variational formulation (1.9) has a unique solution for $f \in H^{-1}(\Omega)$ and $u_0 \in H^1(\Omega)$.

Example 1.4.6: Load force

Let us consider the unit interval $[0,1]$. It is obvious to check that the Dirac delta $\delta_s(v) \doteq v(s)$ for $0 < s < 1$ is linear but does not belong to $L^2((0,1))$. On the other hand, it is possible to check that in 1D

the functions in $H_0^1(\Omega)$ are continuous and thus, $\delta(\cdot) \in H^{-1}((0,1))$. It has some physical implications, e.g., one can consider the elastic bar problem with a point load, whereas it does not have sense in the strong form of the problem.

1.5 Boundary value problem

In this section, we will go from the variational formulation to its corresponding boundary value problem and its corresponding partial differential equation. As we did in the introduction for a 1D problem, it will require some regularity assumptions. They key to transforming the variational form into a partial differential equation is *integration by parts*. We need the following standard results.

Lemma 1.5.1: Product rule

For all $\boldsymbol{\psi} \in C^1(\bar{\Omega})^d$, $v \in C^1(\bar{\Omega})$, it holds

$$\boldsymbol{\nabla} \cdot (\boldsymbol{\psi} v) = v \boldsymbol{\nabla} \cdot (\boldsymbol{\psi}) + \boldsymbol{\psi} \cdot \boldsymbol{\nabla} v.$$

Theorem 1.5.2: Gauss theorem

Let us represent with $\boldsymbol{n} : \partial\Omega \to \mathbb{R}^d$ the outwards normal vector field on $\partial\Omega$. It holds:

$$\int_\Omega \boldsymbol{\nabla} \cdot \boldsymbol{\psi}(x) \mathrm{d}x = \int_{\partial\Omega} \boldsymbol{\psi}(s) \cdot \boldsymbol{n}(s) \mathrm{d}s, \quad \forall \boldsymbol{\psi} \in C^1(\bar{\Omega})^d.$$

where $\mathrm{d}s$ denotes integration with respect to the surface measure.

Theorem 1.5.3: Green's first formula

For all $\boldsymbol{\psi} \in C^1(\bar{\Omega})^d$, $v \in C^1(\bar{\Omega})$, it holds:

$$\int_\Omega \boldsymbol{\psi}(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x = -\int_\Omega \boldsymbol{\nabla} \cdot (\boldsymbol{\psi}(x)) v(x) \mathrm{d}x + \int_{\partial\Omega} \boldsymbol{\psi}(s) \cdot \boldsymbol{n}(s) v(s) \mathrm{d}s.$$

At this point, if we assume that $u \in C^2(\bar{\Omega})$ and thus, $\boldsymbol{\nabla} u \in C^1(\bar{\Omega})$, we

can use Green's first formula to get:

$$\int_\Omega \boldsymbol{\nabla} u(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x$$
$$= -\int_\Omega \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} u(x)) v(x) \mathrm{d}x, \quad \forall v \in C_0^1(\bar{\Omega}),$$

using the fact that the boundary terms vanish because $v = 0$ on $\partial\Omega$. As a result, using the variational formulation (since $C_0^1(\bar{\Omega}) \subset H_0^1(\Omega)$) and assuming that $f \in C^0(\Omega)$, we get

$$\int_\Omega (-\boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} u(x)) - f(x)) v(x) \mathrm{d}x = 0, \ \forall v \in C_0^1(\Omega).$$

Lemma 1.5.4: Fundamental lemma of calculus of variations

If a function $f \in C^0(\Omega)$ is such that

$$\int_a^b f(x) v(x) \mathrm{d}x = 0, \ \forall v \in C_0^\infty(\bar{\Omega})$$

then $f(x) = 0$ for any $x \in \Omega$.

As a result, if the solution of the variational formulation is in $C^2(\Omega)$, it satisfies the following boundary value problem:

$$-\Delta u = f \quad \text{in} \quad \Omega, \quad u = g \quad \text{on} \quad \partial\Omega. \tag{1.10}$$

Δ defined as $\Delta v \doteq \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} v)$ is the so-called *Laplacian* operator.

Solutions of (1.10) are called *classical (or strong)* solutions. If they exist, they are also solutions of the variational formulation. However, the extra regularity assumption requires additional smoothness over the data (not only over the forcing term, which has been made explicit above, but also the domain Ω and the boundary value g). Summarising, the variational formulations have been proved to be well-posed, i.e., it has a unique solution in the maximal function space. The corresponding boundary value problem (1.10) is not well-posed in general because it requires additional smoothness assumptions over the solution that are not true in general.

The general strategy to obtain the boundary value problem out of a variational formulation is to use integration by parts to transfer all derivatives

over test functions to the trial functions, as above. After this process, we end up with the following re-statement of the variational equation in the following form

$$u \in V \ : \ \int T(u) v \mathrm{d}x = 0, \ \forall v \in V.$$

We note that the integral could involve bulk and surface integration. One can ask whether it has sense to do integration by parts for the space V. In fact, it is related to the fact that we consider derivatives *in a weak (or distributional) way*, but we do not want to go further into these concepts. When we want $T(u)$ to have a classical pointwise sense, we need to enforce the additional regularity over u. The equation $T(u) = 0$ is the so-called *Euler-Lagrange* equation of the underlying functional.

1.6 Boundary conditions

So far, we have always enforced the value of the unknown $u : \Omega \to \mathbb{R}$ on the whole domain boundary $\partial\Omega$: $u = g$ on $\partial\Omega$, where we can assume that $g \in C^0(\Omega)$. These are the so-called Dirichlet boundary conditions or *essential* boundary conditions. But we can consider other types of boundary conditions too. For instance, in the model for an elastic membrane above, we could instead consider that part of $\partial\Omega$ is not clamped, and instead, we provide the surface load on that boundary. Analogously, the model above also represents heat conduction. Instead of just imposing the temperature on the whole boundary, we could prescribe the heat flux in part of the boundary. Let us see how these physically relevant boundary conditions are described in our mathematical formulations above.

Let us consider $\Gamma_0 \neq \partial\Omega$, and enforce the Dirichlet boundary condition $u = g$ on Γ_0. On the other hand, let us assume that there exists an offset function $u_0 \in H^1(\Omega)$ that satisfies this boundary conditions. Using the offset function method, $u - u_0 \in V_0 \doteq \{v \in H^1(\Omega) \ : \ v = g \ \text{ on } \Gamma_0\}$; clearly, V_0 is a vector space. On the other hand, we can now define a function

$h : \partial\Omega \setminus \Gamma_0 \to \mathbb{R}$ and state the following linear variational problem:

$$\begin{aligned}\delta u \in V_0 \ : \ & \int_\Omega \boldsymbol{\nabla}\delta u(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x \\ & = -\int_\Omega \boldsymbol{\nabla} u_0(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x \\ & \quad + \int_\Omega f(x) v(x) \mathrm{d}x + \int_{\partial\Omega\setminus\Gamma_0} h(s) v(s) \mathrm{d}s, \ \forall v \in V_0.\end{aligned}$$

h is a prescribed flux on the Neumann boundary $\partial\Omega \setminus \Gamma_0$ (e.g., a heat flux in thermal problems or surface tension in fluid/solid mechanics), and it is data (as f).

Now, since the test functions do not vanish anymore on $\partial\Omega$, we can add boundary *loads*. In order to keep a linear continuous form, the only thing that we need is that $h(v) \doteq \int_{\partial\Gamma_0} h(s) v(s) \mathrm{d}s < \infty$ for any $v \in V_0$.

Now, we are going to use the procedure defined above to recover the boundary value problem associated to this linear variational problem. Let us assume that $u \in C^2(\bar{\Omega})$, $f \in C^0(\Omega)$, and $h \in C^0(\partial\Omega \setminus \Gamma_0)$. We also need to assume that $\partial\Omega$ is such that its outward normal vector field $\boldsymbol{n}$ is in $C^0(\partial\Omega)^d$. Using first Green's formula, now keeping (part of) the boundary terms, we obtain

$$\int_\Omega \left(-\Delta u(x) - f(x)\right) v(x) \mathrm{d}x + \int_{\partial\Omega\setminus\Gamma_0} (\boldsymbol{n} \cdot \boldsymbol{\nabla} u(x) - h(s)) v(s) \mathrm{d}s = 0, \ \forall v \in C^1_{\Gamma_0}(\Omega).$$

where $C^1_{\Gamma_0} \doteq \{v \in C^1(\Omega) \ : \ v = 0 \ \text{ on } \Gamma_0\}$. Using the fundamental lemma of calculus of variations, we get that u satisfies the following boundary value problem:

$$-\Delta u = f \ \text{ in } \Omega, \quad \boldsymbol{n} \cdot \boldsymbol{\nabla} u = h \ \text{ on } \partial\Omega \setminus \Gamma_0, \quad u = g \ \text{ on } \Gamma_0.$$

Thus, we have seen how we can define so-called *Neumann (or natural)* boundary conditions for variational formulations, i.e., via a boundary source or load term, and how it transforms into a flux boundary condition in the corresponding boundary value problem.

1.7 Further topics

In the previous presentation, I have not considered some crucial concepts that are out of the scope of this course. For instance, I have not discussed

Lebesgue integration, Lebesque measure, or a measurable function. I have also omitted the concept of weak derivative and I have not been rigorous about boundary conditions; I have not explained why one can define the trace of a function in $H^1(\Omega)$, in which sense that equality is understood, or the regularity required on the boundary data g for the problem to be well-posed. It would involve introducing trace theorems, which were also out of the scope of this course. In any case, the reader can do an Internet search for these keywords and find lots of material discussing these issues.

1.8 Tutorial

1. Let us consider the problem: find u such that $-u''(x) = \delta(0)$ (where δ is the Dirac delta) in $(-1, 1)$ and $u(-1) = u(1) = 0$. Do you think that this problem has sense in strong (classical) pointwise form? (Hint: Check the regularity of the forcing term (right-hand side of the PDE).)

   State the weak form of the problem. Can you find the solution to this problem? Hint: Use the fact that $H_0^1((-1, 1)) \subset C_0^0([-1, 1])$. Try to find possible solutions to this problem on $[-1, 0]$ and $[0, 1]$ separately. Enforce the continuity of the solution (since the function must belong to H^1) at $x = 0$ to end up with a unique solution.)

2. Show that pointwise evaluation of $L^2((0, 1))$ functions (i.e., the evaluation at a point in $(0, 1)$ of these functions) is not continuous/bounded in general. Hint: Find a counterexample by building a sequence of functions in $L^2((0, 1))$ (i.e., the integral of its suqare is bounded) whose value at a given point (e.g., at 0) is not bounded.

3. We want to solve the following problem: find u such that

$$-\boldsymbol{\nabla}\boldsymbol{\nabla}\cdot\boldsymbol{u} + \boldsymbol{u} = \boldsymbol{f} \text{ in } \Omega \subset \mathbb{R}^3,$$

   for the boundary conditions $\boldsymbol{u}\cdot\boldsymbol{n} = 0$ on $\partial\Omega$. Can you obtain the weak formulation of this problem? Which functional space would you consider as trial/test space? (Hint: Use the definition of maximal functional space)

   Which are the natural boundary conditions for this problem? (Hint: Leave free (no Dirichlet boundary conditions) part of the boundary, use integration by parts, and check the boundary term that arises.)

4. We want to solve the following minimisation problem: find u that minimises the functional

$$J(u) \doteq 1/2 \int_\Omega |\alpha(x)^{1/2} u(x)|^2 + 1/2 \int_\Omega |\beta(x)^{1/2} \boldsymbol{\nabla} u(x)|^2 \mathrm{d}x - \int_\Omega f(x) u(x) \mathrm{d}x,$$

with zero boundary conditions. Can you state the variational formulation? Can you provide *sufficient* conditions on α, β for the problem to be well-posed? (Hint: Consider the Poincaré inequality.)

Chapter 2

Discretisation

In the previous section, we have presented different ways to look at mathematical models that involve partial differential equations. We have observed that, for the elliptic problems, we can state the same problem as a minimisation (a.k.a. variational) or a boundary value problem. We have also observed that these formulations are equivalent only under particular circumstances, i.e., when we have enough regularity. Finally, we have shown the well-posedness of the variational (and minimisation) problems and that the choice of suitable functional space is essential.

We have learned that variational formulations allow one to understand the problem in a weaker (non-pointwise) sense, which allows one to solve more general problems. It has clear *practical* implications. For example, the solution of a clamped elastic structure in an L-shaped domain does not have a solution in the classical sense since the stresses in the inner corner are infinite.

Even though the variational formulations have excellent properties, it is generally impossible to find an analytical solution. These problems are infinite-dimensional and cannot be solved with computers; a computer can only perform floating point operations with finite precision. As a result, it has motivated whole fields of mathematics, namely numerical analysis and scientific computing. These fields are about the design and analysis of *approximations* of mathematical models, with applications in almost any scientific discipline. They are part of a broad scientific area called *computational science*, which has become the third pillar of science after traditional theory and experimentation.

This course focuses on the numerical approximation of partial differential

equations. The idea is simple, try to get the most accurate approximation of a partial differential solution with the minimum computational cost. The spectral and finite element methods that we will study in this course rely on the variational formulation of a mathematical model. It is a clear difference compared to finite difference methods that work on the strong form. It also motivates *why* we have started this course with an introduction to *variational forms and functional spaces.*

We start this section with an abstract discretisation framework (the Galerkin method) that uses finite-dimensional approximations of our variational problem. Next, we will consider two different ways to generate *accurate* finite-dimensional spaces, namely *spectral Galerkin* methods and *finite element* methods. For the time being, we will restrict these constructions to one-dimensional problems since its multi-dimensional generalisation requires some technicalities addressed in the next chapter.

We will perform a numerical analysis of the Galerkin method, showing that the error depends on the discrete space approximability properties. After proving some approximability results, we will obtain error estimates for the finite element method in one dimension.

2.1 Galerkin method

The idea of Galerkin methods is quite simple. Let us recall an abstract variational formulation:

$$u \in V_g \ : \quad a(u;v) = \ell(v), \quad \forall v \in V_0. \tag{2.1}$$

As commented above, this problem is infinite dimensional. In order to *solve* this problem, let us consider *finite dimensional vector subspaces* $V_{N,0} \subset V_0$ (the vector space with homogeneous boundary conditions) and $V_N \subset V$ (without any Dirichlet boundary conditions), where $N \doteq \dim(V_N)$. Let us also consider the affine space with the boundary conditions is $V_{N,g} \subset V_N$. Now, we can solve the following problem.

Definition 2.1.1: Galerkin approximation

Let us consider the variational formulation in (2.1) and subspaces $V_{N,0} \subset V_0$ and $V_N \subset V$. The Galerkin approximation of (2.1) in

$V_{N,0}$ reads:

$$u \in V_{N,g} \ : \quad a(u;v) = \ell(v), \quad \forall v \in V_{N,0}. \tag{2.2}$$

We can readily check that if the variational formulation is linear and well-posed (which now we can check using the results in the previous chapter), the Galerkin approximation is also well-posed (see Theorem 1.2.10). We note that the Galerkin problem in (2.2) is also called the *Galerkin projection* of the problem. The previous problem can be understood as solving (2.1) in the subspace V_N.

Analogously, we could consider the discrete version of the quadratic minimisation problem in Definition 1.2.2. Besides, for non-homogeneous boundary conditions, we can analogously use at the discrete level the offset function method commented above.

$V_{N,0}$ and V_N are real vector spaces of finite dimension, for which we can pick a *basis*.

Definition 2.1.2: Basis of a finite dimensional vector space

Given a finite dimensional real vector space V_M, the set $\{b_1, \ldots, b_M\} \subset V_M$, $M \in \mathbb{N}$ is a basis of V_M if $\forall v \in V_M$ there is a unique set of coefficients $\{\nu_i\}_{i=1}^M \subset \mathbb{R}$ such that $v = \nu_1 b_1 + \ldots + \nu_M b_M$. M is equal to the dimension of V_M.

A finite dimensional functional space V_N of dimension N is isomorphic with respect to $\mathbb{R}^N$. Such an isomorphism can be defined by considering an *ordered* basis and mapping functions in V_N with its unique vector of coefficients $\boldsymbol{\mu} \in \mathbb{R}^N$.

To consider the linear system related to the Galerkin problem, let us first use the offset function method to the Galerkin formulation Definition 2.1.1. Let us also assume that the bilinear form is linear.

Definition 2.1.3: Galerkin method with offset function

Let us consider the Galerkin problem in Definition 2.1.1 for a linear bilinear form. Let us pick an offset function $u_{N,0} \in V_{N,g} \subset V_N$. The

solution of the Galerkin problem reads as $u = u_{N,0} + \delta u_N$, *where*

$$\delta u_N \in V_{N,0} \;:\; a(\delta u_N, v_N) = \tilde{\ell}(v_N) \doteq \ell(v_N) - a(u_{N,0}, v_N), \quad \forall v_N \in V_{N,0}. \tag{2.3}$$

Lemma 2.1.4: Galerkin system as linear system

We consider a basis $\mathcal{B} \doteq \left\{b_N^1, \dots, b_N^{N_0}\right\}$ *of* $V_{N,0}$, *where clearly* $N_0 \doteq \dim(V_{N,0})$. *The Galerkin problem in (2.3) is equivalent to the linear system: find* $u = \sum_{i=1}^{N_0} \mu_i b_N^i$ *with*

$$\begin{aligned} \boldsymbol{\mu} \in \mathbb{R}^{N_0} \quad : \quad & \boldsymbol{A}\boldsymbol{\mu} = \boldsymbol{f}, \quad \text{where} \\ & \boldsymbol{A} \in \mathbb{R}^{N_0 \times N_0}, \quad \boldsymbol{A}_{ij} \doteq a(b_N^j, b_N^i), \quad i,\ j \in \{1, \dots, N_0\}, \\ & \boldsymbol{f} \in \mathbb{R}^{N_0}, \quad \boldsymbol{f}_i \doteq \tilde{\ell}(b_N^i), \quad i \in \{1, \dots, N_0\}. \end{aligned}$$

Proof. Let us consider the solution $u \in V_{N,0}$ of (2.2), and its unique expression $u = \sum_{i=1}^{N_0} \mu_i b_N^i$. Using (2.2) with this expression of u as a linear combination of elements in $\mathcal{B}$ and using as test function the elements of the basis, we get

$$a\left(\sum_{j=1}^{N_0} \mu_j b_N^j, b_N^i\right) = \boldsymbol{A}_{ij}\boldsymbol{\mu}_j = \tilde{\ell}(b_N^j) = \boldsymbol{f}_j, \quad \forall j \in \{1, \dots, N_0\}. \tag{2.4}$$

Thus, it solves the linear system. On the other hand, for any $v \in V_{N,0}$, we can write it as $v = \sum_{n=1}^{N_0} \nu_i b_N^i$. Multiplying (2.4) times ν_j and adding up for $j = 1, \dots, N_0$, we readily check that the variational equation (2.2) holds. □

We note that the choice of the basis $\mathcal{B}$ *does not affect* the solution u of (2.2) but it does affect the matrix $\boldsymbol{A}$, the right-hand side $\boldsymbol{f}$ and the solution vector $\boldsymbol{\mu}$. In short, it determines the isomorphism between $V_{N,0}$ and $\mathbb{R}^{N_0}$.

2.2 Spectral Galerkin methods

Spectral Galerkin methods use as approximation for the infinite-dimensional vector space V (and V_0) the complete set of polynomials up to a given order that vanish on the boundary. For simplicity, let us consider that $\Omega \subset \mathbb{R}$, even though the generalisation to d-cubes does not require new ingredients. We choose V_N to be the set of polynomials up to order p in $\mathbb{R}$, which is represented with $\mathcal{P}_p(\mathbb{R}) \doteq \text{span}\{1, x, \ldots, x^p\}$. In order to enforce boundary conditions, we consider $V_{N,0} \doteq \mathcal{P}_p(\mathbb{R}) \cap C_0^0(\overline{\Omega})$. We note that the space of polynomials $\mathcal{P}_p(\mathbb{R})$ is a vector space of dimension $p+1$. On the other hand, the imposition of the zero boundary conditions at the end-points of Ω reduces the dimension in two, since they involve two indepedent restrictions over such space. As a result $N \doteq p+1$ and $N_0 \doteq p-1$.

Example 2.2.1: Choice of the basis in finite dimension

As commented in the previous section, the choice of the basis for our finite dimensional space does not affect the solution. This assertion would certainly be true if computations were performed with exact arithmetics. However, this is not the case with computers, which can only perform floating point operations up to finite precision. Thus, in practise, the choice of the basis can dramatically affect the solution *due to rounding errors. One example is the basis for the polynomial spaces. For instance, one can easily check that*

$$V_{N,0} = \text{span}\left\{1 - x^2, x(1-x^2), \ldots, x^{p-2}(1-x^2)\right\}.$$

However, such monomial-like basis is highly ill-conditioned and useless for high values of p. Instead, one must consider other polynomial bases like Legendre polynomials. In any case, we are not going to explore this issue here.

In practical applications, the right-hand side will have a general expression. The matrix can also involve physical parameters, e.g., a non-constant heat conductivity or Young's modulus. So, we want a general procedure for numerical integration of functions in terms of floating point of operations that can easily be implemented in computers. Thus, we can replace integrals

by an m-point quadrature formula on $\Omega \doteq [a,b] \subset \mathbb{R}$, $m \in \mathbb{N}$,

$$\int_a^b f(x)\mathrm{d}x \approx Q_m^{[a,b]}(f) \doteq \sum_{j=1}^{m} \omega_j^m f(\xi_j^m),$$

where ω_j^m are the quadrature *weights* and ξ_j^m are the quadrature *points*. The optimal choice for these weights and quadratures in the domain $[-1,1]$ is the m-point Gauss quadrature, which is exact up to polynomials of degree $2m-1$. We note that we can write the integral over a general interval $[a,b]$ as

$$\int_a^b f(x)\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} f(\xi)\mathrm{d}\xi,$$

and apply the Gauss quadrature formula. This way, we can state the spectral Galerkin method in terms of floating point operations only and thus, solve it using computers.

Non-homogeneous boundary conditions in the spectral Galerkin method can be applied as follows. Given the boundary condition

$$u(a) = u_a, \quad u(b) = u_b$$

on the end-points of $[a,b]$, we need to find a polynomial that *goes through* these two points, i.e., that interpolates these values. For two points, we need at least a polynomial in $\mathcal{P}_1(\mathbb{R})$. So, let us consider the *unique* first order interpolation polynomial for the data (a,u_a), (b,u_b) (thus satisfying the boundary conditions), which has the following expression

$$u_0(x) = \frac{bu_a - au_b}{b-a} + \frac{u_b - u_a}{b-a}x.$$

In the next definition, we state the spectral Galerkin method.

Definition 2.2.2: Spectral Galerkin method with offset function

Let us consider the continuous problem in (2.1) where a is a bilinear form. Let us consider the domain $\Omega \doteq [a,b] \subset \mathbb{R}$ and the boundary conditions $u(a) = u_a$ and $u(b) = u_b$. We define the Spectral Galerkin approximation u_p of order $p \in \mathbb{N}^+$ as follows. First, we define the offset

function

$$u_{p,0}(x) = \frac{bu_a - au_b}{b-a} + \frac{u_b - u_a}{b-a}x, \qquad x \in [a,b].$$

Let us define $\mathcal{V}_p \doteq \mathcal{P}_p([a,b]) \cap \mathcal{C}_0^0([a,b])$ *. Next, we compute*

$$\delta u_p \in \mathcal{V}_p \ : \ a(\delta u_p, v_p) = \ell(v_p) - a(u_{p,0}, v_p), \qquad \forall v_p \in \mathcal{V}_p.$$

The Spectral Galerkin approximation finally reads $u_p \doteq u_{p,0} + \delta u_p$.

We note that we have not only defined a finite dimensional space but a *family* of spaces parameterised with the order p, i.e., $\{V_p\}_{p=1}^{\infty}$. In our simulations, we will certainly pick an order. Still, it will be interesting to consider how the solution improves as we increase that order, i.e., to perform a *convergence* analysis of the solution.

2.3 Finite element methods

Spectral methods make use of global polynomials in the domain Ω to build the finite dimensional spaces in the Galerkin formulation. Instead, finite element methods consider a partition of the domain into pieces (*elements or cells* in a *mesh*) and globally continuous functions that are polynomials inside the cells, i.e., *piecewise* polynomial continuous functions.

The first ingredient in a finite element method is the *mesh*, i.e., a partition of the domain Ω. As for the spectral Galerkin method, we consider a 1D problem with $\Omega \doteq [a,b]$. The first ingredient that we require is the concept of *mesh*.

Definition 2.3.1: Mesh in $\Omega \subset \mathbb{R}$

Given a domain $\Omega \doteq [a,b] \subset \mathbb{R}$, $M \in \mathbb{N}$, *and a set of* nodes

$$\Xi_M \doteq \{a \doteq x_0 < x_1 < \ldots < x_{M-1} < x_M \doteq b\},$$

we can define a mesh $\mathcal{M}_M$ *of* Ω *as*

$$\mathcal{M}_M \doteq \{(x_{j-1}, x_j), \ : \ 1 \leq j \leq M\}.$$

The open sub-intervals (x_{j-1}, x_j) *are the* cells *of* $\mathcal{M}_M$*, with cell* size $h_j \doteq |x_j - x_{j-1}|$*. The mesh* width *is* $h \doteq \max_{j\in\{1,\dots,M\}} h_j$*. In the particular case in which all cells have the same size, i.e.,* $h_j \doteq h$ *for any* $j \in \{1, \dots, M\}$*, the mesh is called* uniform.

On top of the 1D mesh, let us create a finite dimensional space $V_{N,0}$. To do that, we define the following function.

Definition 2.3.2: Hat function in 1D linear finite elements

Given a mesh $\mathcal{M}_{N+1}$*, for every interior node* x_j*,* $j = 1, \dots, N$*, we define its corresponding hat function* b_N^j *as*

$$b_N^j \doteq \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}}, & x_{j-1} \le x < x_j, \\ 1 - \frac{x-x_j}{x_{j+1}-x_j}, & x_j \le x < x_{j+1}, \\ 0 & \text{otherwise}, \end{cases}$$

At the end-points, i.e., $j \in \{0, N+1\}$*, we consider the hat functions as*

$$b_N^0(x) \doteq \begin{cases} 1 - \frac{x-a}{h_1} & a \le x \le x_1 \\ 0 & \text{otherwise}, \end{cases}$$

$$b_N^{N+1}(x) \doteq \begin{cases} 1 - \frac{b-x}{h_{N+1}} & x_N \le x \le b, \\ 0 & \text{otherwise}. \end{cases}$$

The linear finite element space concerning $\mathcal{M}_{N+1}$ *is defined as*

$$V_N \doteq \left\{b_N^0, b_N^1, \dots, b_N^{N+1}\right\}, \qquad V_{N,0} \doteq \left\{b_N^1, \dots, b_N^N\right\}.$$

It is obvious to check that $V_{N,0}$ vanishes at the end-points. We can also observe that the hat functions are continuous and piecewise linear polynomials. In fact, the hat function b_N^j associated to node x_j is equal to zero at all cells of the mesh $\mathcal{M}_{N+1}$ but the ones that *contain* x_j, i.e., $[x_{j-1}, x_j]$ and $[x_j, x_{j+1}]$. In these two cells, the function is a first order polynomial, reason why it is called a linear finite element space. We can also re-state the global

finite element spaces as

$$V_N \doteq \left\{ v \in C^0([a,b]) \ : \ v_{[x_{j-1},x_j]} \in \mathcal{P}_1([x_{j-1},x_j]),\ j = 1,\dots,N \right\}, \quad (2.5)$$
$$V_{N,0} \doteq V_N \cap C_0^0([a,b]).$$

On the other hand, we can also check that $b_N^j(x_i) = \delta_{ij}$, i.e., hat functions take the value one in their corresponding node and zero at all other nodes. In the finite element method, the basis $\mathcal{B}_N$ that satisfies this property is the basis of *shape functions*, which we will define below.

We also know from the first chapter that the hat functions are in $H^1(\Omega)$. Thus, $V_N \subset H^1((a,b))$ and $V_{N,0} \subset H_0^1((a,b))$. The derivative of the interior hat function b_N^j takes the following value

$$\frac{\mathrm{d}b_N^j}{\mathrm{d}x}(x) \doteq \begin{cases} \frac{1}{h_j}, & x_{j-1} \le x < x_j \\ -\frac{1}{h_{j+1}}, & x_j \le x \le x_{j+1} \\ 0, & \text{otherwise.} \end{cases} \quad (2.6)$$

We proceed similarly for the end-point shape functions. Let us remark that *finite element spaces are unsuitable if we require a pointwise well-defined (classical) derivative*. The fact that we will only need our space to be a subset of $H^1(\Omega)$ is what allow us to use finite element spaces (for partial differential equations that involve at most second-order derivatives in their strong form).

Definition 2.3.3: Finite element method with offset function

Let us consider the continuous problem in (2.1) where a is a bilinear form. Let $\Omega \doteq [a,b] \subset \mathbb{R}$ and the boundary conditions $u(a) = u_a$ and $u(b) = u_b$. Given a mesh $\mathcal{M}_{N+1}$, we build the finite element spaces V_N and $V_{N,0}$ in (2.5) . With these ingredients, we define the finite element approximation u_N as follows. First, we define the offset function

$$u_{N,0}(x) = u_a b_N^0 + u_b b_N^{N+1}.$$

Next, we compute

$$\delta u_N \in V_{N,0} \ : \ a(\delta u_N, v_N) = \ell(v_N) - a(u_{N,0}, v_N),\ \forall v_N \in V_{N,0}.$$

The finite element approximation finally reads $u_N \doteq u_{N,0} + \delta u_N$.

2.3.1 Computing the entries of the linear system

In this section, we will apply the finite element method to an elementary problem computing by-hand the matrix and right-hand side.

Example 2.3.4: 1D model problem

We consider our 1D model problem introduced in the previous chapter:

$$u \in H_0^1((a,b)) : \int_a^b \kappa(x)\frac{\mathrm{d}u}{\mathrm{d}x}(x)\frac{\mathrm{d}v}{\mathrm{d}x}(x)\mathrm{d}x = \int_a^b f(x)v(x)\mathrm{d}x,$$

$\forall v \in H_0^1((a,b))$. *We consider homogeneous boundary conditions,* $f(x) = 1$ *and* $\kappa(x) = 1$ *for simplicity. We want to compute the system matrix for a uniform mesh with* $N+1$ *cells.*

Using Lemma 2.1.4, the entries of the matrix and right-hand side using the Galerkin method applied with the 1D linear finite element space in a uniform mesh read

$$\boldsymbol{A} = \int_a^b \frac{\mathrm{d}b_N^j}{\mathrm{d}x}(x)\frac{\mathrm{d}b_N^i}{\mathrm{d}x}(x)\mathrm{d}x,$$

Now, using the expression of the derivatives of the hat functions in (2.6) , we get

$$\boldsymbol{A}_{ij} \doteq \begin{cases} -\frac{1}{h_{i+1}}, & \text{if } j = i+1 \\ -\frac{1}{h_i}, & \text{if } j = i-1 \\ \frac{1}{h_i} + \frac{1}{h_{i+1}}, & \text{if } j = i \\ 0 & \text{otherwise,} \end{cases}$$

for $i,\ j = 1,\dots,N$. An extremely important property of finite element methods is their *sparsity*; the entries $\boldsymbol{A}_{ij}$ of the matrix are equal to zero if $|i-j| \geq 2$. Thus, the system matrix will read:

$$\begin{bmatrix} \frac{1}{h_1}+\frac{1}{h_2} & -\frac{1}{h_2} & 0 & \dots & & 0 \\ -\frac{1}{h_2} & \frac{1}{h_2}+\frac{1}{h_3} & -\frac{1}{h_3} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 \\ \vdots & & & & & \vdots \\ 0 & \ddots & \ddots & 0 & -\frac{1}{h_{N-1}} & \frac{1}{h_{N-1}}+\frac{1}{h_N} \end{bmatrix}$$

As commented above, we need a numerical quadrature to integrate the right-hand side. For simplicity, let us consider the trapezoidal rule to perform the numerical integration at every cell of the mesh, i.e.,

$$\int_{x_{i-i}}^{x_i} j(x)\mathrm{d}x \approx \frac{1}{2}h_i(j(x_{i-1}) + j(x_i)).$$

Assuming that we have a subroutine that allows us to evaluate the forcing term at a point, we can compute the entries of the right-hand side for a unit force as follows:

$$\boldsymbol{f}_j = \int_a^b b_N^j(x)\mathrm{d}x = \frac{h_j + h_{j+1}}{2}, \qquad j = 1, \dots, N.$$

In the particular case in which the mesh is uniform with a mesh size h, the linear system reads:

$$\begin{bmatrix} 2 & -1 & 0 & \dots & & & 0 \\ -1 & 2 & -1 & 0 & \dots & & 0 \\ 0 & \ddots & \ddots & \ddots & & & 0 \\ \vdots & & & & & & \vdots \\ 0 & \ddots & \ddots & & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \\ \vdots \\ \\ \mu_N \end{bmatrix} = h \begin{bmatrix} 1 \\ \\ \vdots \\ \\ 1 \end{bmatrix}$$

The systems that arise from finite element and finite difference schemes are identical for a 1D uniform mesh and a trapezoidal rule for the right-hand side. However, this is not the case for general meshes or $d > 1$, as we will see.

Now, let us consider the non-homogeneous boundary conditions $u(a) = u_a$ and $u(b) = u_b$. Let us note that the end-point hat functions are not being used for solving the problem with homogeneous boundary conditions, they are essential to pick a suitable offset function. As commented above, we take $u_0 \doteq u_a b_N^0 + u_b b_N^{N+1}$. It is obvious to check that u_0 satisfies the right boundary conditions. The main reason why we use such technique for finite element methods is to keep *sparsity*. With this expression, the additional term to be added to the right hand side is

$$\begin{aligned} -\int_a^b \frac{\mathrm{d}b_N^j}{\mathrm{d}x}\frac{\mathrm{d}u_0}{\mathrm{d}x}\mathrm{d}x &= -\int_a^b \frac{\mathrm{d}b_N^j}{\mathrm{d}x}\frac{\mathrm{d}b_N^0}{\mathrm{d}x}\mathrm{d}x - \int_a^b \frac{\mathrm{d}b_N^j}{\mathrm{d}x}\frac{\mathrm{d}b_N^{N+1}}{\mathrm{d}x}\mathrm{d}x \\ &\doteq \begin{cases} \frac{u_a}{h_1} & j = 1 \\ \frac{u_b}{h_{N+1}} & j = N \\ 0 & j \in \{2, \dots, N-1\}. \end{cases} \end{aligned}$$

On the contrary, if we chose the offset function in the spectral Galerkin method, all the entries in the right-hand side would be different from zero. It would imply the computation of terms in all the cells of the mesh, which is a computational cost that we can avoid. On the other hand, as we will see, the finite element approach is a practical way to impose Dirichlet boundary conditions for general geometries in 2D and 3D. At the same time, it is impossible to do this with global polynomials.

2.4 Error analysis of the Galerkin method

Let us consider our linear variational problem

$$u \in V_0 \ : \ a(u,v) = \ell(v), \qquad \forall v \in V_0,$$

where we recall that V_0 is a real vector space of functions from a physical domain Ω to $\mathbb{R}$.

Assumption 2.4.1: Requirements for a well-posed linear variational problem

We assume $a : V_0 \times V_0 \to \mathbb{R}$ to be a bilinear symmetric and positive definite form and $\ell : V_0 \to \mathbb{R}$ a continuous linear form with respect to the energy norm $\|\cdot\|_a$. Furthermore, V_0 endowed with the inner product provided by a is a Hilbert space (it is complete).

Under these assumptions, we already know from Theorem 1.3.7 that there exists a unique solution $u_N \in V_0$ to Theorem 1.3.7. Now, let us consider the Galerkin discretisation of this problem, using as trial and test space the vector space $V_{N,0} \subset V_0$, $\dim V_{N,0} < \infty$. The Galerkin approximation reads:

$$u \in V_{N,0} \ : \ a(u,v) = \ell(v), \qquad \forall v \in V_{N,0}.$$

The existence and uniqueness of the discrete problem is a direct consequence of the continuous counterpart since the discrete problem inherits the requirements in Assumption 2.4.1 above (even though we already observed that the proof for the discrete case is much simpler). The remaining point in the Galerkin formulations is the definition of the discrete space $V_{N,0}$; we have learned two ways to define such space in 1D, using either the spectral Galerkin method or the finite element method.

Now, the essential questions that arise are: Which is the error we are committing with our discrete scheme? How does this error change with N?. In the following, we will error bounds for the abstract Galerkin problem above. As we will see, the dependence of the error with respect to N does (naturally) depend on the particular construction of the discrete space $V_{N,0}$, e.g., whether we are using a spectral Galerkin or a finite element method, etc.

2.4.1 Errors in the energy norm

We want to analyse the error commited by the Galerkin method. A natural way to look at this problem is to find bounds for the norm of the error function $u - u_N$. The easiest choice for the error norm is to pick the energy norm. First, let us state what we know about u and u_N. These functions are solution of the continuous and Galerkin problems:

$$\begin{aligned} a(u, v) &= \ell(v), \qquad \forall v \in V_0, \\ a(u_N, v_N) &= \ell(v_N), \qquad \forall v \in V_{N,0}. \end{aligned}$$

Thus, using the linearity of the bilinear form and the fact that $V_{N,0} \subset V_0$, we can also pick $v_N \in V_{N,0}$ as test function in the continuous problem. Choosing the same test function in both problems and subtracting, we readily get:

$$a(u - u_N, v_N) = 0, \qquad \forall v_N \in V_{N,0}. \tag{2.7}$$

This result is called *Galerkin orthogonality*; the error function is orthogonal to the discrete space $V_{N,0}$ with respect to the a-inner product.

It tells us that the Galerkin solution is the best possible solution in $V_{N,0}$ with respect to the energy norm. For any $v_N \in V_{N,0}$ we have

$$\begin{aligned} \|u - v_N\|_a^2 &= \|u - u_N\|_a^2 + \|u_N - v_N\|_a^2 + 2a(u - u_N, u_N - v_N) \\ &= \|u - u_N\|_a^2 + \|u_N - v_N\|_a^2. \end{aligned}$$

Theorem 2.4.2: Cea's lemma

Let us assume that the boundary conditions on the Dirichlet boundary $u = g$ on Γ_0 can be exactly imposed with the discrete finite element space V_N, i.e., we can pick $u_{N,g} \in V_N$ such that $u_{N,g} = g$ on Γ_0. The Galerkin discretisation in Definition 2.1.3 under Assumption 2.4.1

satisfies

$$\|u - u_N\|_a = \inf_{v_N \in V_{N,0}} \|u - v_N\|_a.$$

Proof. First, we note that due to the assumptions in the theorem, we can pick the same offset function for both the continuous and discrete problem, thus having exactly the same right-hand side. As a result, we can use the orthogonality in (2.7). Using the bilinearity of a, we readily get

$$\|u-u_N\|_a^2 = a(u-u_N, u-u_N) = a(u-v_N, u-u_N)+a(v_N-u_N, u-u_N)$$

for any $v_N \in V_{N,0}$. The last term in this expression vanishes due to the orthogonality of the Galerkin projection. Thus, using the Cauchy-Schwarz inequality we readily get

$$\|u - u_N\|_a^2 \leq \|u - v_N\|_a \cdot \|u - u_N\|_a.$$

Dividing this expression by $\|u - u_N\|_a$, and taking the infimum with respect to v_N, we prove the result. □

This result gives us an essential piece of information. The Galerkin solution is the best possible solution in the trial space, measured in terms of the energy norm error. As a result, we only need to check how accurately functions in $V_{N,0}$ can approximate the continuous solution.

Cea's lemma also tells us that if we consider two discrete spaces $V_{N,0}$ and $V_{M,0}$ such that $V_{N,0} \subset V_{M,0}$, then the error $\|u-u_N\|$ is smaller than $\|u-u_M\|$, since

$$\inf_{v_N \in V_{N,0}} \|u - v_N\|_a \leq \inf_{v_M \in V_{M,0}} \|u - v_M\|_a.$$

For spectral Galerkin methods, the solution improves as we increase the order since $V_{p+1} \subset V_p$. For finite element methods, it also holds when *refining* the mesh $\mathcal{M}_{N+1}$; in 1D, it implies splitting every cell into two cells to get $\mathcal{M}_{2N+2}$. Thus, $V_{2N+2,0} \subset V_{N+1,0}$. As a result, we can obtain more accurate approximations of our problem by enlarging the discrete space in the Galerkin method. In any case, we still do not know how this reduction is in terms of N.

2.5 Approximation theory

As we have commented above, error bounds on the Galerkin approximation can be attained if we can determine how accurately the discrete space can approximate functions in the continuous space. Approximation theory answers this question. First, we will introduce the concept of *interpolator operator*, which takes continuous functions and provides discrete functions. Then, we will use this interpolation operator to determine the *interpolation error* when using piecewise polynomial (finite element) spaces. It will provide a bound for the error of the Galerkin solution by the results in the previous section, i.e., Cea's lemma.

Definition 2.5.1: Interpolation operator for 1D finite elements

Given the linear 1D finite element space

$$V_{N,0} = \left\{ v \in C^0([a,b]) \ : \ v_{[x_{i-1},x_i]} \in \mathcal{P}_1([x_{i-1},x_i]) \right\},$$

we define the interpolation operator $\pi : V \to V_N$ *as follows:*

$$\pi(v) \doteq \sum_{i=0}^{N+1} v(x_i) b_N^i,$$

where b_N^i *stands for the hat functions in Definition 2.3.2.*

We can see that this operator is an interpolation operator. It has sense in 1D for $V = H^1((a,b))$ because $C^0([a,b]) \subset H^1((a,b))$ and thus, the functions have pointwise sense. Thus, given a function $v \in V$, to compute its interpolant $\pi(v) \in V_N$, we must evaluate v at the mesh nodes. These functionals (pointwise evaluations $v \mapsto v(x_i)$ for $i = 0, \ldots, N+1$) are the so-called *degrees of freedom* of the finite element space V_N.

Definition 2.5.2: Degrees of freedom

Let us consider a set of $\{\sigma_i\}_{i=0}^{N+1}$ *linear functionals in* V_N*. We say that* $\{\sigma_i\}_{i=0}^{N+1}$ *is an admissible basis of* degrees of freedom *if the operator* $I : V_N \to \mathbb{R}^{N+1}$ *such that*

$$I(v) \doteq [\sigma_0(v), \sigma_1(v), \ldots, \sigma_{N+1}(v)]^T$$

is a bijection.

Definition 2.5.3: Shape functions

Given a basis of degrees of freedom, we say that a basis $\{b_N^0, \ldots, b_N^{N+1}\}$ *of* V_N *is the basis of* shape functions *if* $\sigma_N^i(b_N^j) = \delta_{ij}$ *for* $i,\ j = 1, \ldots, N+1$.

We can infer from this definition that there is a unique basis of shape functions. Furthermore, the shape functions in Definition 2.3.2 are this basis for the finite element space V_N.

Now, let us prove some key interpolation theory results. We prove stability first and error bounds next.

Lemma 2.5.4: Continuity of the 1D interpolant

The map $\pi : H^1(\Omega) \to V_{N+1} \subset H^1(\Omega)$ *is continuous (bounded) uniformly with respect to the mesh width* h *for a bounded* $\Omega \subset \mathbb{R}$.

Proof. First, let us note that $V_N \subset H^1(\omega)$ is a consequence of the fact that hat functions are in $H^1(\omega)$. Besides, functions in $H^1(\omega)$ for a bounded $\omega \subset \mathbb{R}$ are continuous in one dimension. Thus, for any $v \in H^1(\omega)$ and $x,\ y \in \overline{\omega}$, we have:

$$|v(y) - v(x)| \leq \int_x^y |v'(s)| \mathrm{d}s \leq |y - x|^{\frac{1}{2}} |v|_{H^1(\omega)}, \tag{2.8}$$

where we have used the Cauchy-Schwarz inequality. Let us pick x as the point in which $|v|$ reaches its minimum on $\overline{\omega}$. We readily have that $|v(x)| \leq |\omega|^{-\frac{1}{2}} \|v\|_{L^2(\Omega)}$; $|\omega|$ denotes the size (length) of the domain. Thus, we have, by the triangle inequality:

$$\|v\|_{\infty,\omega} \doteq \sup_{y \in \omega} |v(y)| \leq |\omega|^{-\frac{1}{2}} \|v\|_{L^2(\omega)} + |\omega|^{\frac{1}{2}} |v|_{H^1(\omega)}. \tag{2.9}$$

Therefore, pointwise evaluations are bounded in $H^1(\omega)$. As a result, $\pi : H^1(\omega) \to H^1(\omega)$ is a bounded operator in $H^1(\omega)$. Now, we can

easily check that $\pi(v)'|_{[x_{i-1},x_i]} = \frac{v(x_i)-v(x_{i-1})}{h_i}$. Using the result (2.8) for $\omega \doteq [x_{i-1}, x_i]$, we get

$$|\pi(v)|_{H^1([x_{i-1},x_i])} = h_i^{-\frac{1}{2}} |v(x_i) - v(x_{i-1})| \leq |v|_{H^1([x_{i-1},x_i])}.$$

Adding up for all cells, we get

$$|\pi(v)|_{H^1(\Omega)} \leq |v|_{H^1(\Omega)}.$$

On the other hand,

$$\|\pi(v)\|_{L^2(\Omega)} \leq |b-a|^{\frac{1}{2}} \|\pi(v)\|_{\infty,\Omega}, \qquad \|\pi(v)\|_{\infty,\Omega} \leq \|v\|_{\infty,\Omega}$$

Using these expressions in (2.9) for the domain Ω, we get:

$$\|\pi(v)\|_{L^2(\Omega)} \leq \|v\|_{L^2(\Omega)} + |b-a||v|_{H^1(\Omega)} \leq c\|v\|_{H^1(\Omega)}.$$

As a result, $\|\pi(v)\|_{H^1(\Omega)} \leq c\|v\|_{H^1(\Omega)}$, and thus a continuous (bounded) operator with a constant that is independent of the mesh size h. □

The error estimates below require some regularity over the solution.

Definition 2.5.5: The $H^2(\Omega)$ space

The space $H^2(\Omega)$ is the subspace of functions in $H^1(\Omega)$ such that its second derivatives are in $L^2(\Omega)$. The corresponding norm reads:

$$\|v\|_{H^2(\Omega)} \doteq \|v\|_{H^1(\Omega)} + |\boldsymbol{\nabla} v|_{H^1(\Omega)}, \qquad |v|_{H^2(\Omega)} \doteq |\boldsymbol{\nabla} v|_{H^1(\Omega)}.$$

Lemma 2.5.6: Error bounds for the 1D interpolant

For all $v \in H^2(\Omega)$, the interpolant π in a mesh with mesh width h holds:

$$\|v - \pi(v)\|_{L^2(\Omega)} \leq h^2 |v|_{H^2(\Omega)}, \qquad |v - \pi(v)|_{H^1(\Omega)} \leq h|v|_{H^2(\Omega)}.$$

Proof. Let us consider the cell $[x_{i-1}, x_i]$ and $(v - \pi(v))|_{[x_{i-1},x_i]}$. Using the fact that $v - \pi(v)$ vanishes at the end-points of the interval, we can make use of the First Poincaré-Friedrichs inequality to bound the $L^2((x_{i-1}, x_i))$ norm by the $H^1((x_{i-1}, x_i))$ seminorm (which is a norm in $H_0^1((x_{i-1}, x_i))$). (Alternatively, you can just use (2.8).) In this case, the diameter of $((x_{i-1}, x_i))$ is h. We get

$$\|v - \pi(v)\|_{L^2((x_{i-1},x_i))} \leq h_i |v - \pi(v)|_{H^1((x_{i-1},x_i))}. \tag{2.10}$$

Now, let us consider the function $v' - \pi(v)'$. Due to the mean-value theorem, we know this function vanishes at some point in the cell (this is in fact all we need to be able to use the Poincaré-Friedrichs inequality, see (2.8)). On the other hand, $\pi(v)'' = 0$ on the cell since it is a first-order polynomial. Thus, we can use the previous bound also for the derivative and obtain

$$\|v - \pi(v)\|_{H^1((x_{i-1},x_i))} \leq h_i |v|_{H^2((x_{i-1},x_i))}.$$

Combining these results and adding them up for all cells in the mesh, we prove the results in the lemma. □

The following corollary is a direct consequence of the previous approximation error estimates and Cea's lemma.

Corollary 2.5.7: Error estimates for the finite element solution in 1D

Under the conditions in Cea's lemma and assuming that $\Omega \subset \mathbb{R}$ for a linear finite element space V_N for a mesh with mesh width h, the following a priori *error bound holds:*

$$\|u - u_N\|_{H^1(\Omega)} \leq h |u|_{H^2(\Omega)}$$

Results with respect to the $L^2(\Omega)$ norm also hold:

$$\|u - u_N\|_{L^2(\Omega)} \leq h^2 \|u\|_{H^1(\Omega)},$$

where we have weakened the norm of the error estimate and in turn, improved the convergence order. We will not prove this result since it involves *duality* arguments that are out of the scope of this course.

2.5.1 Higher order methods

Now that we have learned how to construct and analyse linear finite element methods, we can go one step further. We can now consider at every cell of the mesh a polynomial of an arbitrary order p. Given a $p \in \mathbb{N}$ and a mesh $\mathcal{M}_{N+1}$ of the domain $\Omega \doteq [a, b]$, we define the p-th order finite element space as:

$$V_N^p \doteq \left\{ v \in C^0([a,b]) \ : \ v_{\text{-}}[x_{i-1}, x_i] \in \mathcal{P}^p([x_{i-1}, x_i]) \right\}.$$

We must define its degrees of freedom and associated dual space of shape functions. It is clear that using only the mesh nodes of the *linear* mesh $\mathcal{M}_{N+1}$ is not enough to uniquely define elements of V_{n+1}^p. We need to add more degrees of freedom. To do that, let us consider the following set of DOFs

$$\begin{aligned} &\sigma_i^1 \doteq v(x_i), && i = 0, \dots, N+1, \\ &\sigma_i^k \doteq v\left(x_{i-1} + \frac{(x_i - x_{i-1})k}{p} \right), && i = 1, \dots, N+1, \ k = 1, \dots, p-1 \end{aligned} \tag{2.11}$$

Clearly, for linear finite elements ($p = 1$) we recover the previous definition. For *quadratic* elements ($p = 2$), we have to add one additional node at the mid-point of each cell. For an element of order p, we add $p - 1$ equidistant nodes at each cell of the mesh. It is easy to check that this set of degrees of freedom define a bijective map between V_{N+1}^p and $\mathbb{R}^{N_p}$, with $N_p = N + (N - 1) \cdot (P - 1)$.

Concerning the shape functions, we still keep locality properties. The shape functions associated to *interior* degrees of freedom (the high order ones) σ_i^k vanish in all cells but the cell $[x_{i-1}, x_i]$. At such cell, shape functions are zero on the cell boundary and all interior nodes, i.e., $\frac{x_{i-1}-x_i}{p}, \dots, \frac{(x_{i-1}-x_i)(p-1)}{p}$, but one. E.g., in the case of quadratic elements, there is only one interior node, one interior degree of freedom, and its corresponding shape function is a parabola that vanishes on the end-points of the cell and takes value on the mid-point. The shape functions associated to interior *linear* nodes have support on two cells, e.g., σ_i^1 for $i = 1, \dots, N$ has support in cells $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$.It takes the value one in such nodes and vanishes in the other end-points and interior nodes of these two cells.

High-order polynomials pay the price (in fact, they are much better than mesh refinement in terms of accuracy vs. number of degrees of freedom) as

soon as the solution of the problem at hand is regular enough. For example, we can now extend the definition of the interpolant π^p to arbitrary order by simply using the set of degrees of freedom in (2.11). The following error estimates provide us essential information.

Lemma 2.5.8: Error bounds for the 1D interpolant

Given a finite element space of order p, i.e., V_N^p, on a mesh with mesh width h, the interpolant π^p holds:

$$\|v - \pi^p(v)\|_{L^2(\Omega)} \leq h^{p+1}|v|_{H^{p+1}(\Omega)}, \qquad |v - \pi^p(v)|_{H^1(\Omega)} \leq h^p|v|_{H^{p+1}(\Omega)},$$

for any $v \in H^{p+1}(\Omega)$.

Proof. The proof of this result is analogous to the one of the linear case. The idea is to obtain a similar result as (2.10) for all derivates up to order p of the error function $v - \pi^p(v)$. Next, we use the same result for its $p+1$ derivative and observe that the $p+1$ derivative of the interpolant vanishes at each cell. □

We can readily combine these interpolation errors with Cea's lemma to get the following result.

Corollary 2.5.9: Error estimates for the finite element solution in 1D

Under the conditions in Cea's lemma and assuming that $\Omega \subset \mathbb{R}$ for a finite element space V_N^p of order p in a mesh with mesh width h and $u \in H^{q+1}(\Omega)$ for $0 < q \leq p$, the following a priori *error bound holds:*

$$\|u - u_N\|_{H^1(\Omega)} \leq h^q|u|_{H^{q+1}(\Omega)}.$$

2.6 Tutorial

1. We want to use a third-order finite element method in 1D and integrate the matrix corresponding to the following bilinear form

$$a(u,v) \doteq \int_a^b u(x)v(x) + \boldsymbol{\nabla} u(x) \cdot \boldsymbol{\nabla} v(x) \mathrm{d}x$$

exactly. Let us consider a Gauss quadrature for the integration in every mesh cell. How many points would we need in the Gauss quadrature at every cell? (Hint: Which is the maximum order of the polynomials we want to integrate in each spatial dimension?)

2. Can you compute the system matrix of the bilinear form

$$a(u,v) \doteq \int_a^b \kappa(x) u'(x) v'(x) \mathrm{d}x,$$

in a 1D mesh, in terms of the cell mesh sizes h_i. Do a diagram as in the finite element matrix above, providing all the information needed to determine the matrix entirely. Use the trapezoidal rule for the integration in each cell.

3. In a finite element code, students are told to show the error between the exact and the finite element solution (using the Galerkin method) in terms of the square of the energy norm, i.e., $\|u - u_N\|_a^2$. One student has implemented instead $\|u\|_a^2 - \|u_N\|_a^2$. Can you tell me which error is s/he committing doing that? In other words, what is the value of $\|u - u_N\|_a^2 - \|u\|_a^2 + \|u_N\|_a^2$.

4. Let us consider a linear finite element approximation of a problem with an analytical solution in $C^\infty(\Omega)$, with uniform mesh with N elements. We want a more accurate solution. We can consider refining the mesh using *bisection*, i.e., split every cell into two cells, to get a $2N$ mesh or to consider a second order finite element space on the same mesh. What is going to be more effective in terms of error reduction? Quantify it.

Convergence plots

In the tutorials for the unit, which can be found in this Github repos, we evaluate how the error decreases when refining the mesh (h-refinement) and

increasing the order (p-refinement). The primary motivation is to observe how the error behaves in terms of degrees of freedom, i.e., the size of the corresponding linear system of equations, which is a measure of the computational cost.

Let us start with linear finite elements on a 1D uniform mesh with N cells. As a result, the mesh width is $\frac{1}{N}$. In this case, the number of degrees of freedom $\sharp \doteq N - 1$. Thus, using the results in Corollary 2.5.9, if our solution is in $H^2(\Omega)$, we have:

$$e_N \doteq \|u - u_N\|_a \leq \frac{C_1}{N},$$

for some constant $C > 0$, or using Landau notation, $e_N \lesssim \mathcal{O}(N^{-1})$. We can now apply the log function on both sides of this inequality. We readily get

$$\log e_N \leq \log C_2 - \log N = C_3 - \log N.$$

If we consider now finite elements of order p on the same mesh, we can readily check that $\sharp \doteq pN - 1$, whereas $h = \frac{1}{N}$. Using the error bounds in Corollary 2.5.9, and *assuming that* $u \in H^{p+1}(\Omega)$, we obtain

$$e_{N,p} \doteq \|u - u_N\|_a \leq \frac{C_4}{N^p}.$$

Again, if we compute the log function at both sides of the inequality, we get

$$\log e_{N,p} \leq \log C_5 - p \log N = C_6 - p \log N, \qquad \sharp = pN + 1.$$

With these bounds, we can consider the following refinements strategies:

- h-refinement: This refinement involves increasing the cells in our mesh (e.g., using bisection) and keeping the polynomial order p. E.g., using uniform refinement we multiply the number of degrees of freedom $\sharp$ by 2 and reduce the error by a factor $\frac{1}{2}$.

  In this situation, we have $\sharp \approx pN$ and

  $$\log e_{N,p} \leq C_6 - p \log \sharp + p \log p = C_7 - p \log \sharp.$$

  Thus, $\log e_{N,p}$ must be reduced at least linearly with slope $-p$ in terms of $\log \sharp$. As an example, using bisection, increasing $\sharp$ by two, we reduce the error by two. This kind of convergence is called *algebraic* convergence.

- p-refinement: This refinement consists in fixing the mesh and increasing the polynomial order in the finite element method. Using the fact that $\sharp \approx pN$, we have that

$$\log e_{N,p} \leq C_6 - p \log N = C_6 - \frac{\log N}{N} \sharp.$$

  Thus, $\log e_{N,p}$ is reduced at least linearly in terms of $\sharp$.

As a result, *if the solution is smooth enough*, p-refinement is much more effective than h-convergence. Whereas p-refinement reduces $\log e_{N,p}$ linearly with the number of degrees of freedom $\sharp$, h-refinement does it with respect to $\log \sharp$. The difference is huge!

Chapter 3

n-dimensional finite elements

At this point, we have already introduced the Galerkin method for the numerical approximation of partial differential equations. The Galerkin method requires finite dimensional subspaces of the (infinite dimensional) functional spaces in which our weak formulation is well-posed. In order to do that, we have built finite element methods for 1D problems. In this chapter we are going to extend to arbitrary dimensions the definition of finite element spaces.

3.1 The boundary value problem in weak form

We are interested in problems governed by partial differential equations (PDEs) posed in a physical domain $\Omega \subset \mathbb{R}^d$ with boundary $\Gamma \doteq \partial\Omega$. In practice $d = 2, 3$ but we are also interested in $d > 3$ for some particular applications. Let us consider a differential operator A, e.g., the Laplace operator $-\Delta$, and a force term $f : \Omega \to \mathbb{R}$. Let us also consider a partition of Γ into a Dirichlet boundary Γ_{D} and a Neumann boundary Γ_{N}, and the corresponding boundary data $u_{\mathrm{D}} : \Gamma_{\mathrm{D}} \to \mathbb{R}$ and $g_{\mathrm{N}} : \Gamma_{\mathrm{N}} \to \mathbb{R}$. The boundary value problem reads as follows: find $u(\boldsymbol{x})$ such that

$$\begin{aligned} Au(\boldsymbol{x}) &= f(\boldsymbol{x}) \quad \text{in } \Omega, \\ B_{\mathrm{D}}u(\boldsymbol{x}) &= u_{\mathrm{D}}(\boldsymbol{x}) \quad \text{on } \Gamma_{\mathrm{D}}, \\ B_{\mathrm{N}}u(\boldsymbol{x}) &= g_{\mathrm{N}}(\boldsymbol{x}) \quad \text{on } \Gamma_{\mathrm{N}}. \end{aligned} \tag{3.1}$$

The operator B_D is a trace operator[1] and B_N is the flux operator.[2] Other boundary conditions, e.g., Robin (mixed) conditions can also be considered. We assume the unknown $u(\boldsymbol{x})$ in (3.1) can be a scalar, vector, or tensor field. (The case of multi-field problems is considered in Sect. 3.11.)

For finite element analysis, (3.1) must be understood in a weak sense. The weak formulation can be stated in an abstract setting. Let us consider an abstract problem determined by a Hibert space $\mathcal{X}$ (*trial space*), a continuous bilinear form $a : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and a continuous linear form $\ell : \mathcal{X} \to \mathbb{R}$. The abstract problem is stated as: find $u \in \mathcal{X}$ such that

$$a(u, v) = \ell(v), \qquad \text{for any } \; v \in \mathcal{X}. \tag{3.2}$$

The link between the weak and strong formulations has already been described in the previous sections. Without mathematical rigor, one can simply consider integration by parts (assuming continuous functions) to transfer derivatives from the trial to the test function. E.g., for the Laplace operator, the bilinear form reads $a(u, v) \doteq \int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \mathrm{d}\Omega$. Furthermore, homogeneous Dirichlet boundary conditions, i.e., $u = 0$ on Γ_D, are usually enforced in a strong way; the functions in $\mathcal{X}$ satisfy these boundary conditions. The extension to non-homogeneous boundary conditions can be done using the offset function method. One can define an arbitrary extension Eu_D of the Dirichlet data, i.e., $Eu_\mathrm{D} = u_\mathrm{D}$ on Γ_D. Next, we define the function $u_0 \doteq u - Eu_\mathrm{D}$ with zero trace on Γ_D and solve (3.2) for u_0 with the right-hand side

$$\ell(v) - a(Eu_\mathrm{D}, v).$$

Let us consider one classical example.

Example 3.1.1: Heat equation

Let us consider the Poisson problem $-\boldsymbol{\nabla} \cdot \boldsymbol{\kappa} \boldsymbol{\nabla} u = f$ *with* $u = u_\mathrm{D}$ *on* Γ_D *and* $\boldsymbol{n} \cdot \boldsymbol{\kappa} \boldsymbol{\nabla} u = g_\mathrm{N}$ *on* Γ_N*;* $\boldsymbol{n}$ *is the outward normal. Let us assume that* $\boldsymbol{\kappa} \in L^\infty(\Omega)^{d \times d}$, $f \in H^{-1}(\Omega)$, $g_\mathrm{N} \in H^{-\frac{1}{2}}(\Gamma_\mathrm{N})$, *and* $u_\mathrm{D} \in H^{\frac{1}{2}}(\Gamma_\mathrm{D})$.

[1] The trace operator tells us which are the right boundary conditions to be imposed on the boundary and in which sense do they have sense. E.g., for $H^1(\Omega)$, we can impose the full trace of the unknown. The space of traces is $H^{\frac{1}{2}}(\partial\Omega)$.

[2] We have already seen how to determine the Neumann or natural boundary conditions using integration by parts (assuming smooth trial and test functions). E.g., for problems posed in $H^1(\Omega)$, the fluxes are to be understood in $H^{-\frac{1}{2}}(\Gamma_\mathrm{N})$, the dual of the trace space.

Let us also consider an extension $Eu_{\mathrm{D}} \in H^1(\Omega)$ such that $Eu_{\mathrm{D}} = u_{\mathrm{D}}$ on Γ_{D}. The weak form of the problem reads as: find $u_0 \in H^1_{\Gamma_{\mathrm{D}}}(\Omega)$ (the subspace of functions in $H^1(\Omega)$ that vanishes on Γ_{D}) such that

$$\int_\Omega \boldsymbol{\kappa}\boldsymbol{\nabla} u_0 \cdot \boldsymbol{\nabla} v \mathrm{d}\Omega = \int_\Omega f v \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{N}}} g v \mathrm{d}\Gamma - \int_\Omega \boldsymbol{\kappa}\boldsymbol{\nabla} E u_{\mathrm{D}} \cdot \boldsymbol{\nabla} v \mathrm{d}\Omega,$$

for any $v \in H^1_{\Gamma_{\mathrm{D}}}(\Omega)$. The solution is $u \doteq u_0 + Eu_{\mathrm{D}}$.

3.2 Space discretization with finite elements

Problem (3.2) is an infinite-dimensional problem. In order to end up with a computable one, we must introduce finite-dimensional subspaces with some approximability properties. We restrict ourselves to *conforming* finite element schemes, which hold $\mathcal{X}_h \subset \mathcal{X}$, the discrete problem reads as: find $u_h \in \mathcal{X}_h$ such that

$$a(u_h, v_h) = \ell(v_h), \qquad \text{for any } \; v_h \in \mathcal{X}_h. \tag{3.3}$$

This is the *Galerkin* problem. One can also define the affine operator

$$\mathcal{F}_h(u_h) = a_h(u_h, \cdot) - \ell_h(\cdot) \in \mathcal{X}_h'. \tag{3.4}$$

We note that for a given u_h this is a bounded linear functional that takes test functions in $\mathcal{X}_h$ and return real values, thus in its dual space $\mathcal{X}_h'$. Thus, we can state (3.3) as: find $u_h \in \mathcal{X}_h$ such that $\mathcal{F}_h(u_h) = 0$.

In order to define finite element spaces, we require a triangulation $\mathcal{T}_h$ of the domain Ω into a set $\{K\}$ of *cells*. This triangulation is assumed to be conforming, i.e., for two neighbour cells K^+, $K^- \in \mathcal{T}_h$, its intersection $K^+ \cap K^-$ is a *whole* k-face ($k < d$) of both cells.[3] Thus, for every element $K \in \mathcal{T}_h$, we assume that there is a reference cell $\hat{K}_K$ and a diffeomorphism (smooth bijective map) $\boldsymbol{\Phi}_K : \hat{K} \to K$. In what follows, we usually use the notation $\hat{\boldsymbol{x}} \doteq \boldsymbol{\Phi}_K^{-1}(\boldsymbol{x})$ (see Figure 3.2 for a conforming mesh and the geometrical map definition).

[3] We note that k-face refers to a geometrical entity for a *polytope*. For a three-dimensional polytope (e.g., a hexahedron or tetrahedron) the celll itself is a 3-face, the faces are 2-faces, the edges are 1-faces and vertices are 0-faces. In finite element methods, the cells can be mapped to a particular type of mapping over a set of admissible geometries (polytopes).

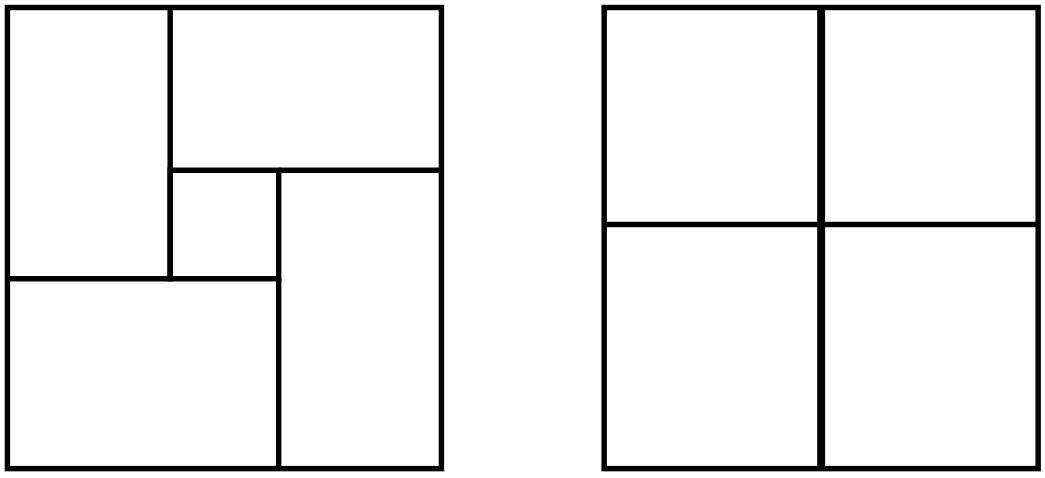


Figure 3.1: A non-conforming mesh on the left-hand side; the intersection of the closure of cells is not always the closure of a vertex or edge in all cells. The mesh on the right-hand side is conforming.

Next, we will see the standard procedure for building finite element functional spaces. They rely on a *reference cell* functional space as follows:

1. We define a functional space in the reference cell $\hat{K}$;
2. We define a set of functions in the physical cell K via function and geometrical maps;
3. We define the global space as the assemble of cell-based spaces plus continuity constraints between cells.

In order to present this process, we introduce the concept of reference finite element, finite element, and finite element space, respectively.

3.3 The finite element in reference and physical space

Using the abstract definition of *Ciarlet*, a finite element is represented by the triplet $\{K, \mathcal{V}, \Sigma\}$, where:

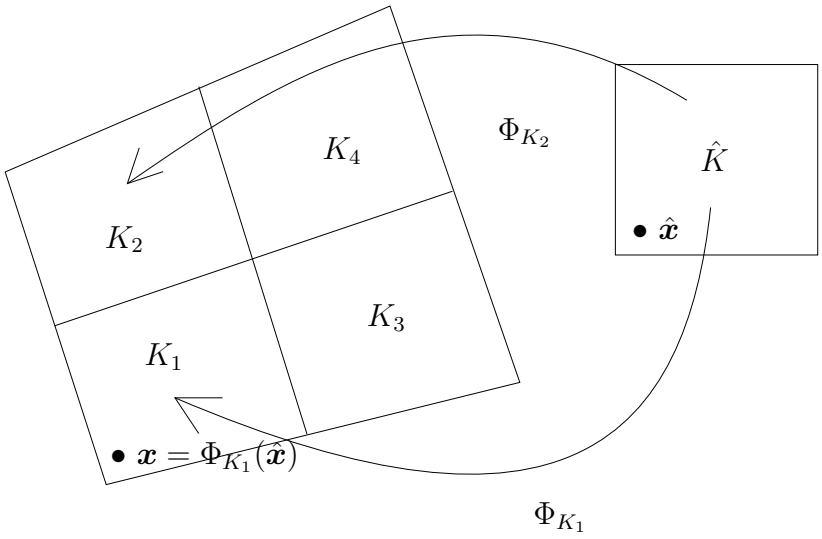


Figure 3.2: In this plot, we show a finite element mesh of 4 quadrilateral cells in the physical space on the left, and the corresponding reference squared cell $\hat{K}$ on the right. For every cell in the physical space K_i , we have a map that takes points $\hat{\boldsymbol{x}} \in \hat{K}$ and returns points $\boldsymbol{x} \in K_i$. These maps are assumed to be smooth and bijective, otherwise the mesh is not suitable for finite element analysis.

1. K is a compact, connected, Lipschitz subset of $\mathbb{R}^d$,
2. $\mathcal{V}$ is a vector space of functions,
3. and Σ is a set of linear functionals that form a basis for the dual space $\mathcal{V}'$.

The elements of Σ are the so-called degrees of freedom (DOFs) of the finite element. We denote the number of DOFs as n_Σ. The DOFs can be written as σ_a for $a \in \mathcal{N}_\Sigma \doteq \{1, \dots, n_\Sigma\}$. We can also define a basis $\{b^1, \dots, b^{n_\Sigma}\}$ for $\mathcal{V}$. In particular, we are interested in the basis $\{\phi^a\}_{a \in \mathcal{N}_\Sigma}$ for $\mathcal{V}$ such that $\sigma_a(\phi^b) = \delta_{ab}$ for $a, b \in \mathcal{N}_\Sigma$. These functions are the so-called *shape functions* of the finite element, and there is a one-to-one mapping between shape functions and DOFs.

3.3.1 The reference finite element

In the reference space, we build *reference* finite elements $(\hat{K}, \hat{\mathcal{V}}, \hat{\Sigma})$ as follows. First, we consider a bounded set of possible cell geometries, denoted by $\hat{K}$.[4] On $\hat{K}$, we build a functional space $\hat{\mathcal{V}}$ and a set of DOFs $\hat{\Sigma}$. In this chapter, we will use polynomials spaces for $\hat{\mathcal{V}}$ and the degrees of freedom will be the evaluation of the functions at a set of nodes (points). We consider some examples of reference finite elements later on.

In Figure 3.3 we can see on the left a bilinear squared reference finite element. The reference cell $\hat{K} \doteq (0,1)^2$. We define as a basis the so-called bilinear polynomials spanned by $\hat{\mathcal{V}} \doteq \text{span}\{1, \hat{x}, \hat{y}, \hat{x}\hat{y}\}$. If we represent the four vertices of $\hat{K}$ with $\{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \hat{\boldsymbol{x}}_3, \hat{\boldsymbol{x}}_4\}$, the space of degrees of freedom $\Sigma \doteq \{\hat{\sigma}_1, \hat{\sigma}_2, \hat{\sigma}_3, \hat{\sigma}_4\}$ are the node evaluations, i.e., $\hat{\sigma}_i(\hat{v}) \doteq \hat{v}(\hat{\boldsymbol{x}}_i)$.

As stated above, one can check that Σ is a basis of $\mathcal{V}'$ (the dual space of $\mathcal{V}$, which is the space of linear functionals from $\mathcal{V}$ to $\mathbb{R}$). Note that two bilinear polynomials that have the same nodal values are the same. Thus, it has the maximum dimension of four and thus a bijection.

3.3.2 From reference to physical spaces

In the physical space, the finite element triplet $(K, \mathcal{V}, \Sigma)$ on a mesh cell $K \in \mathcal{T}_h$ relies on:

[4] Admissible geometries are a segment in one-dimension, triangles and quadrilaterals in two-dimensions, and tetrahedra and hexahedra in three-dimensions.

1. a reference finite element $(\hat{K}, \hat{\mathcal{V}}, \hat{\Sigma})$;

2. a geometrical mapping $\boldsymbol{\Phi}_K$ such that $K \doteq \boldsymbol{\Phi}_K(\hat{K})$;

3. a linear bijective function mapping $\hat{\Psi}_K : \hat{\mathcal{V}} \to \hat{\mathcal{V}}$.[5]

With these ingredients, we construct the physical finite element $(K, \mathcal{V}, \Sigma)$ as follows.

1. The geometry is $K \doteq \boldsymbol{\Phi}_K(\hat{K})$;

2. The functional space in the physical space is defined as

$$\mathcal{V} \doteq \{\Psi_K(\hat{v}) \doteq \hat{\Psi}_K(\hat{v}) \circ \boldsymbol{\Phi}_K^{-1} : \hat{v} \in \hat{\mathcal{V}}\};$$

   where $\Psi_K : \hat{\mathcal{V}} \to \mathcal{V}$ is defined as

$$\Psi_K \doteq \hat{\Psi}_K(\hat{v}) \circ \boldsymbol{\Phi}_K^{-1}.$$

   Again, in the case of the Lagrangian elements it is just $\Psi_K(\hat{v}) \doteq \hat{v} \circ \boldsymbol{\Phi}_K^{-1}$. In Figure 3.3 we show this construction for grad-conforming finite elements, in which $\Psi_K(\hat{v}) = \hat{v} \circ \boldsymbol{\Phi}_K^{-1}$.

3. The set of DOFs in the physical space is defined as

$$\Sigma \doteq \{\hat{\sigma} \circ \Psi_K^{-1} : \hat{\sigma} \in \hat{\Sigma}\}.$$

   In Figure 3.4 we show this idea for grad-conforming finite elements. The idea is to take a function defined in the physical space $u(\hat{x})$ and transform it to the reference space $\hat{u}(\hat{x}) \doteq \Psi_K^{-1}(u)(\hat{x}) = u \circ \boldsymbol{\Phi}_K^{-1}(\hat{\boldsymbol{x}})$. Now we can apply the degrees on freedom on the resulting function

$$\hat{\sigma}_i\left(u \circ \boldsymbol{\Phi}_K\right) = u \circ \boldsymbol{\Phi}_K(\hat{\boldsymbol{x}}_i) = u(\boldsymbol{x}_i).$$

[5]This is the general definition for a finite element in the physical space. Nevertheless, the last ingredient is not required for the finite element spaces considered herein, Lagrangian finite element spaces for problems in $H^1(\Omega)$, a.k.a. *grad-conforming*[6] finite elements. As a result, you can think this operator $\hat{\Psi}_K$ *is just the identity.*

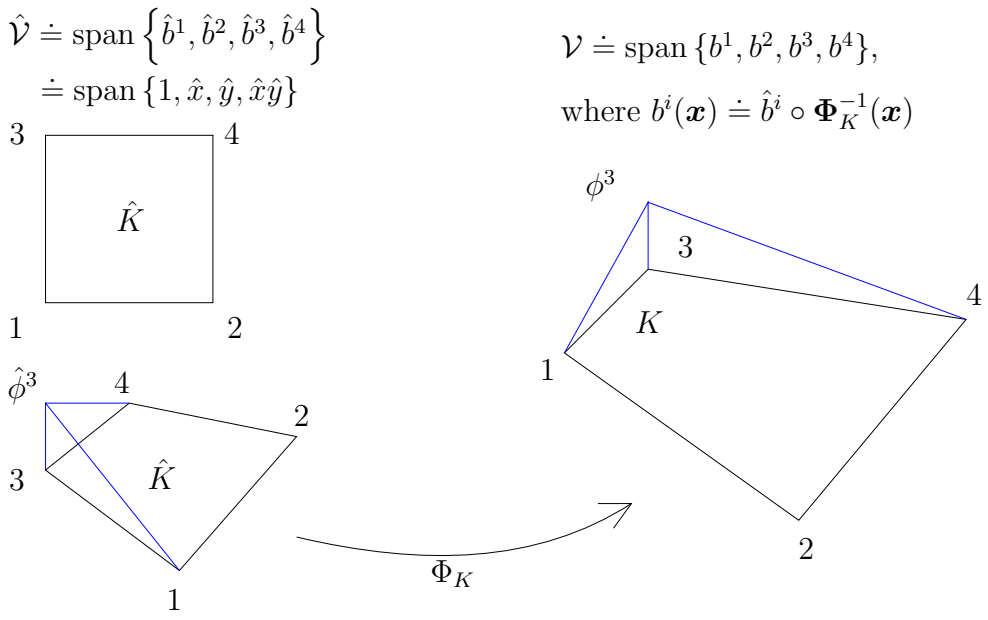


Figure 3.3: In this plot, we show the map from the reference cell $\hat{K}$ to a physical cell K. We consider a linear finite element, and show the indexing for its vertices/nodes. We have a basis for the Lagrangian reference finite element space $\hat{\mathcal{V}} \doteq \{1, \hat{x}, \hat{y}, \hat{x}\hat{y}\}$ and define the one in the reference finite element space from this one and the geometrical map $\mathcal{V}$. For node 3, we plot the shape function associated to the node both in the reference space $\hat{\phi}^3(\hat{\boldsymbol{x}}$ and the physical space $\phi^3(\boldsymbol{x}) = \hat{\phi}^3 \circ \boldsymbol{\Phi}_K^{-1}(\boldsymbol{x})$.

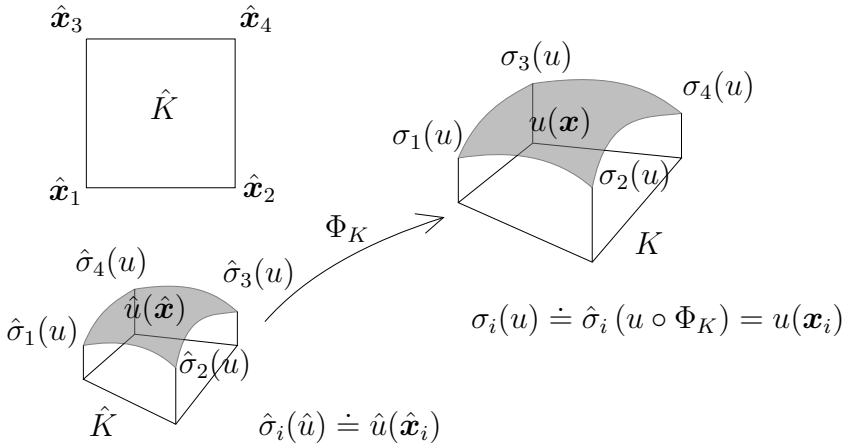


Figure 3.4: In this plot, we show the degrees of freedom for a linear finite element in two-dimensions. In Lagrangian finite elements, all degrees of freedom have a geometrical interpretation, i.e., they are linked to a node (point). For linear elements, the nodes are the vertices of the quadrilateral (idem for triangles). In the reference cell $\hat{K}$, given a function $u \in \mathcal{C}^0(\overline{\hat{K}})$, the degrees of freedom are just the evaluations of the function in the corresponding nodes, i.e., $\hat{\sigma}_i(\hat{u}) \doteq \hat{u}(\hat{\boldsymbol{x}}_i)$. In the physical space, we compose the solution in $\mathcal{C}^0(\overline{K})$, compose it with Φ_K^{-1} and now we can apply the reference degree of freedom, i.e., $\sigma_i(u) \doteq \hat{\sigma}_i\,(u \circ \Phi_K) = u(\boldsymbol{x}_i)$.

The reference finite element space $\hat{\mathcal{V}}$ is usually a polynomial space. Thus, the first ingredient is to define bases of polynomials, e.g., monomial bases; see Sect. 3.4. In Sect. 3.5, we show how to compute the basis of shape functions out of whatever basis $\hat{\mathcal{V}}$ in the reference space. Given the set of shape functions $\{\hat{\phi}^a : a \in \mathcal{N}_{\hat{\Sigma}}\}$ in the reference finite element, it is easy to check that $\{\phi_K^a \doteq \Psi_K(\hat{\phi}^a) : a \in \mathcal{N}_{\hat{\Sigma}}\}$ are the set of shape functions of the finite element in the physical space.

The definition of degrees of freedom in the physical space is essential to project functions in an infinite dimensional space, e.g., $\mathcal{C}^0\left(\overline{K}\right)$, into $\mathcal{V}$. Given a function v, we define the *local interpolator* for the finite element at hand as

$$\pi_K(v) \doteq \sum_{a \in \mathcal{N}_\Sigma} \sigma_a(v)\phi^a.$$

One can check that the interpolation operator is in fact a projection. This operation is, e.g., used to project Dirichlet boundary conditions into the finite element space using the offset function method.

Since the following exposition is restricted to Lagrangian FEs, we will not use $\hat{\Psi}$ (it is just the identity) and we will always use $\Psi_K(\hat{v}) = \hat{v} \circ \mathbf{\Phi}_K^{-1}$.

3.4 Construction of polynomial spaces

Local finite element spaces are usually polynomial spaces. Given an order $k \in \mathbb{N}$ and a set $\mathcal{N}_k$ of distinct points (nodes) in $\mathbb{R}$ (we will indistinctly represent nodes by their index i or position x_i), we define the corresponding set of Lagrangian polynomials $\{\ell_0^k, \ldots, \ell_k^k\}$ as:

$$\ell_m^k(x) \doteq \frac{\Pi_{n \in \mathcal{N}_k \setminus \{m\}}(x - x_n)}{\Pi_{n \in \mathcal{N}_k \setminus \{m\}}(x_m - x_n)}.$$

We can also define the Lagrangian basis $\mathcal{L}^k = \{\ell_i^k : 0 \leq i \leq k\}$. This set of polynomials are a basis for k-th order polynomials. We note that $\ell_m^k(x_l) = \delta_{ml}$, for $0 \leq m, l \leq k$.

For multi-dimensional spaces, we can define the set of nodes as the tensor product of 1D nodes. Given a d-tuple order $\boldsymbol{k}$, we define the corresponding set of nodes for n-cubes as: $\mathcal{N}^{\boldsymbol{k}} \doteq \mathcal{N}^{k_1} \times \cdots \times \mathcal{N}^{k_d}$ (see Figure 3.5).

Analogously, we define the multi-dimensional Lagrange basis

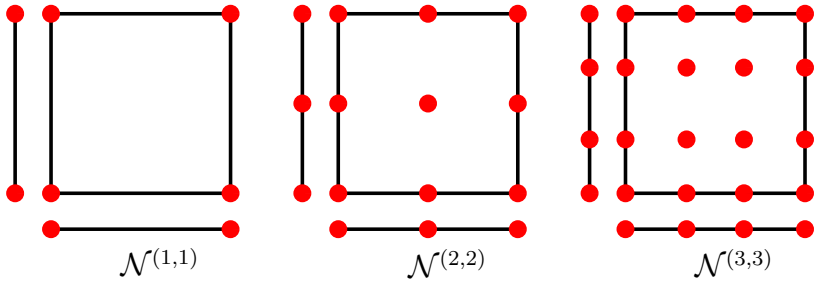


Figure 3.5: Equidistant nodes for two-dimensional Lagrangian polynomial spaces $\mathcal{Q}_{(1,1)}$, $\mathcal{Q}_{(2,2)}$ and $\mathcal{Q}_{(3,3)}$ obtained as tensor product of their one-dimensional counterparts. In short, these isotropic order spaces are expressed as $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$, resp.

$$\mathcal{L}^{\boldsymbol{k}} = \{\ell^{\boldsymbol{k}}_{\boldsymbol{m}} : \boldsymbol{m} \in \mathcal{N}^{\boldsymbol{k}}\}, \qquad \text{where} \quad \ell^{\boldsymbol{k}}_{\boldsymbol{m}}(\boldsymbol{x}) \doteq \Pi_{i=1}^{d} \ell^{k_i}_{m_i}(x_i).$$

Clearly, $\ell^{\boldsymbol{k}}_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{s}}) = \delta_{\boldsymbol{st}}$, for $\boldsymbol{s}, \boldsymbol{t} \in \mathcal{N}^{\boldsymbol{k}}$. [7]

This tensor product construction leads to a basis for the space of polynomials that are of degree less or equal to k with respect to each variable $x_1, \ldots, x_d$. We can define monomials by a d-tuple $\boldsymbol{\alpha}$ as $p_{\boldsymbol{\alpha}}(\boldsymbol{x}) \doteq \Pi_{i=1}^{d} x_i^{\alpha_i}$, and the polynomial space of order $\boldsymbol{k}$ as $\mathcal{Q}_{\boldsymbol{k}} = \text{span}\{p_{\boldsymbol{\alpha}}(\boldsymbol{x}) : 0 \leq \alpha_i \leq k_i,\ i = 1, \ldots, d\}$. This space is also spanned by the Lagrangian basis $\mathcal{Q}_{\boldsymbol{k}} = \text{span}\{\ell : \ell \in \mathcal{L}^{\boldsymbol{k}}\}$. For isotropic spaces in which the same order $p \in \mathbb{N}$ is used in all dimensions we simply write $\mathcal{Q}_p$.

3.4.1 Local finite element space in cubes

Let us consider the same order for all components, i.e., $k\mathbf{1} \doteq (k, \ldots, k)$. When the reference geometry $\hat{K}$ is an n-cube, we define the reference finite element space as $\mathcal{V}_k \doteq \mathcal{Q}_{k\mathbf{1}}$. The set of nodes $\mathcal{N}^{k\mathbf{1}}$ can be generated, e.g., from the equidistant Lagrangian nodes. Let us define the bijective mapping $\mathtt{i}(\cdot)$ from the set of nodes $\mathcal{N}^{k\mathbf{1}}$ to $\{1, \ldots, |\mathcal{N}^{k\mathbf{1}}|\} \equiv \mathcal{N}_\Sigma$, i.e., the local node

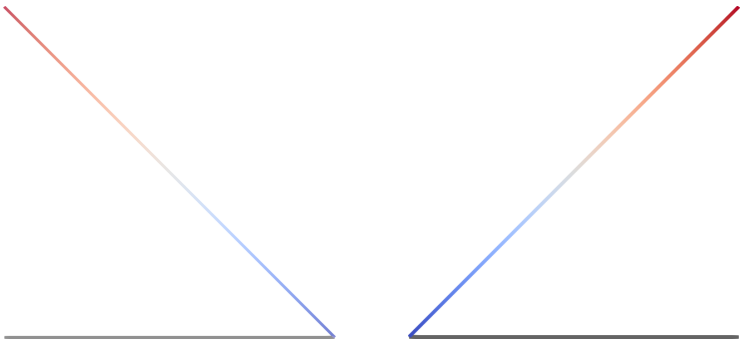
[7] A different polynomial order in each space dimension (anisotropic spaces) is required when building some finite element spaces. However, for the case we want to consider in this text, i.e., grad-conforming finite elements, we will use the same order in all dimensions (isotropic spaces). E.g., $\boldsymbol{k} \doteq (p, p)$ in two-dimensions and $\boldsymbol{k} \doteq (p, p, p)$ in three-dimensions, for $p \in \mathbb{N}$.

numbering. The set of local DOFs $\mathcal{N}_{\Sigma_K}$ are the nodal values, i.e., $\sigma_{\mathtt{i}(\boldsymbol{s})} \doteq v(\boldsymbol{x}_{\boldsymbol{s}})$, for $\boldsymbol{s} \in \mathcal{N}^{\boldsymbol{k}}$. Clearly, the reference finite element shape functions related to these DOFs are $\phi^{\mathtt{i}(\boldsymbol{s})} \doteq \ell^{k\mathbf{1}}_{\boldsymbol{s}}$. $\hat{\Psi}(v) \doteq v$.

Let us start with the one-dimensional spaces (see Figure 3.6 for linear and Figure 3.7 for quadratic shape functions, resp.). These Lagrangian bases span $\mathcal{Q}_1 \doteq \{1, x\}$ and $\mathcal{Q}_2 \doteq \{1, x, x^2\}$. Using the tensor product definition we can readily construct higher-dimensional shape functions. The isotropic space obtained as tensor product of linear polynomials in two-dimensions is the bilinear space $\mathcal{Q}_1 = \text{span}\,\{1, x, y, xy\}$. The one for second order polynomials is

$$\mathcal{Q}_2 = \text{span}\left\{1, x, y, xy, x^2, y^2, x^2y, xy^2, x^2y^2\right\}.$$

The corresponding Lagrangian bases (of shape functions) are collected in Figure 3.8 for linear and Figure 3.9 for quadratic shape functions, resp.).

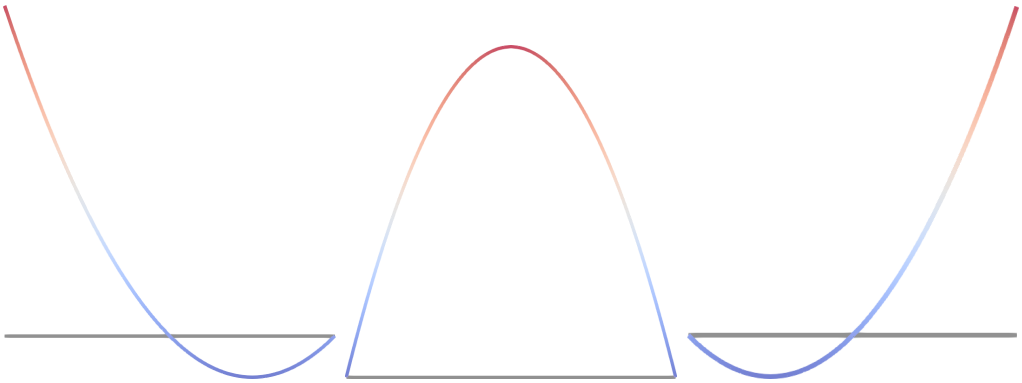


(a) 1 (b) 2

Figure 3.6: Shape functions for the one-dimensional linear finite element space $\mathcal{P}_1 = \mathcal{Q}_1$ in the reference segment. The label says which is the related Lagrangian node in $\mathcal{N}_1$.

3.4.2 Local finite element space in simplices

The definition of polynomial spaces on n-simplices is slightly different. It requires the definition of the space of polynomials of degree equal or less than k in the variables $x_1, \ldots, x_d$. It does not involve a full tensor product of 1D Lagrange polynomials (or monomials) but a truncated space, i.e., the corresponding polynomial space of polynomials up to order k, which is $\mathcal{P}_k = \text{span}\{p_{\boldsymbol{\alpha}}(\boldsymbol{x}) : |\boldsymbol{\alpha}| \leq k\}$, with $|\boldsymbol{\alpha}| \doteq \sum_{i=1}^{d} \alpha_i$.



(a) 1 (b) 2 (c) 3

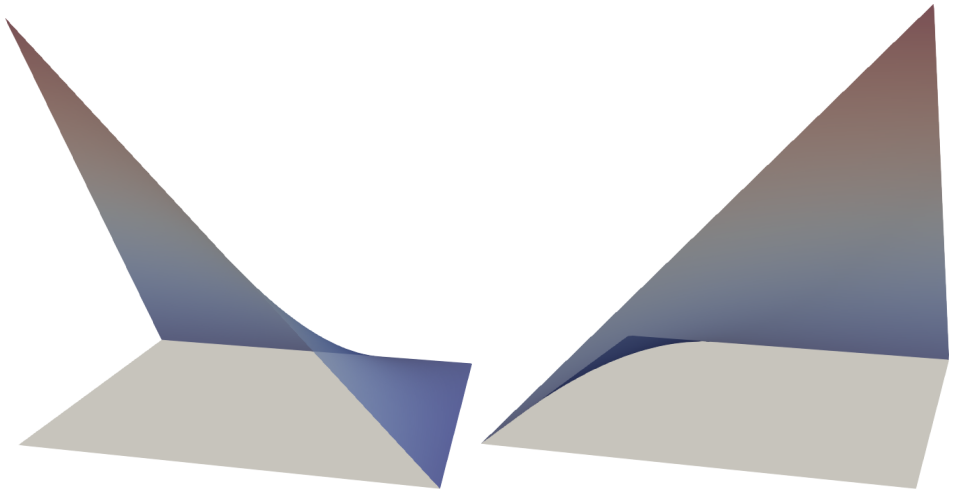
Figure 3.7: Shape functions for the one-dimensional quadratic finite element space $\mathcal{P}_2 = \mathcal{Q}_2$ in the reference segment. The label says which is the related Lagrangian node in $\mathcal{N}_2$.

As an example, in two-dimensions, the linear space $\mathcal{P}_1 \doteq \{1, x, y\}$ whereas the quadratic space $\mathcal{P}_2 \doteq \{1, x, y, x^2, xy, y^2\}$.

Analogously as for n-cubes, we can define a basis of the dual space, i.e., the degrees of freedom as nodal evaluations. A basis for the dual space of $\mathcal{P}_k$ are the values at the set of nodes $\tilde{\mathcal{N}}^k \doteq \{\boldsymbol{s} \in \mathcal{N}^{k\mathbf{1}} : |\boldsymbol{s}| \leq k\}$ (see Figure 3.10. It generates the typical grad-conforming finite elements on n-simplices. Whereas the construction of the shape function bases is somehow easy for quad meshes, using the tensor product of the Lagrangian basis, the situation is more complicated for tet meshes. In the following section, we show an easy method to transform a monomial basis into the shape functions basis in a general way. The shape functions for linear and quadratic elements can be found in Figures 3.11 and 3.12, resp.

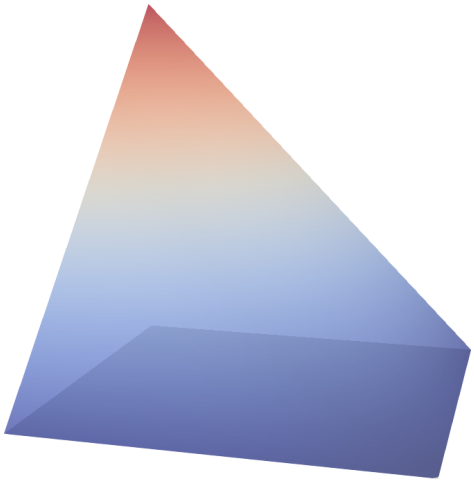
3.4.3 Shape functions in the physical space

Let us consider quadrilateral finite elements. For the computation of the geometrical map Φ_K, we can consider a d-linear approximation determined by the position of the vertices in the physical domain. Thus, we can use all the finite element machinery so far to compute this geometrical map. Given the nodal values, i.e., the position of the vertices $\boldsymbol{x}^a$ in the physical space,

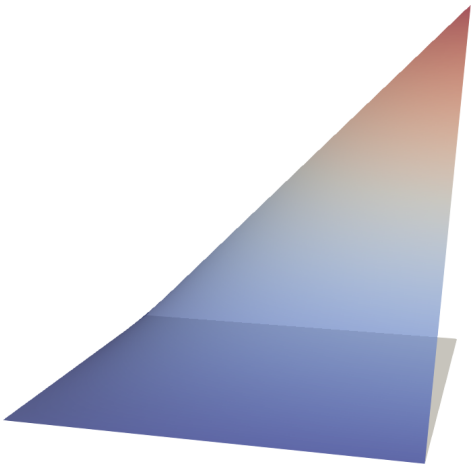


(a) 1

(b) 2

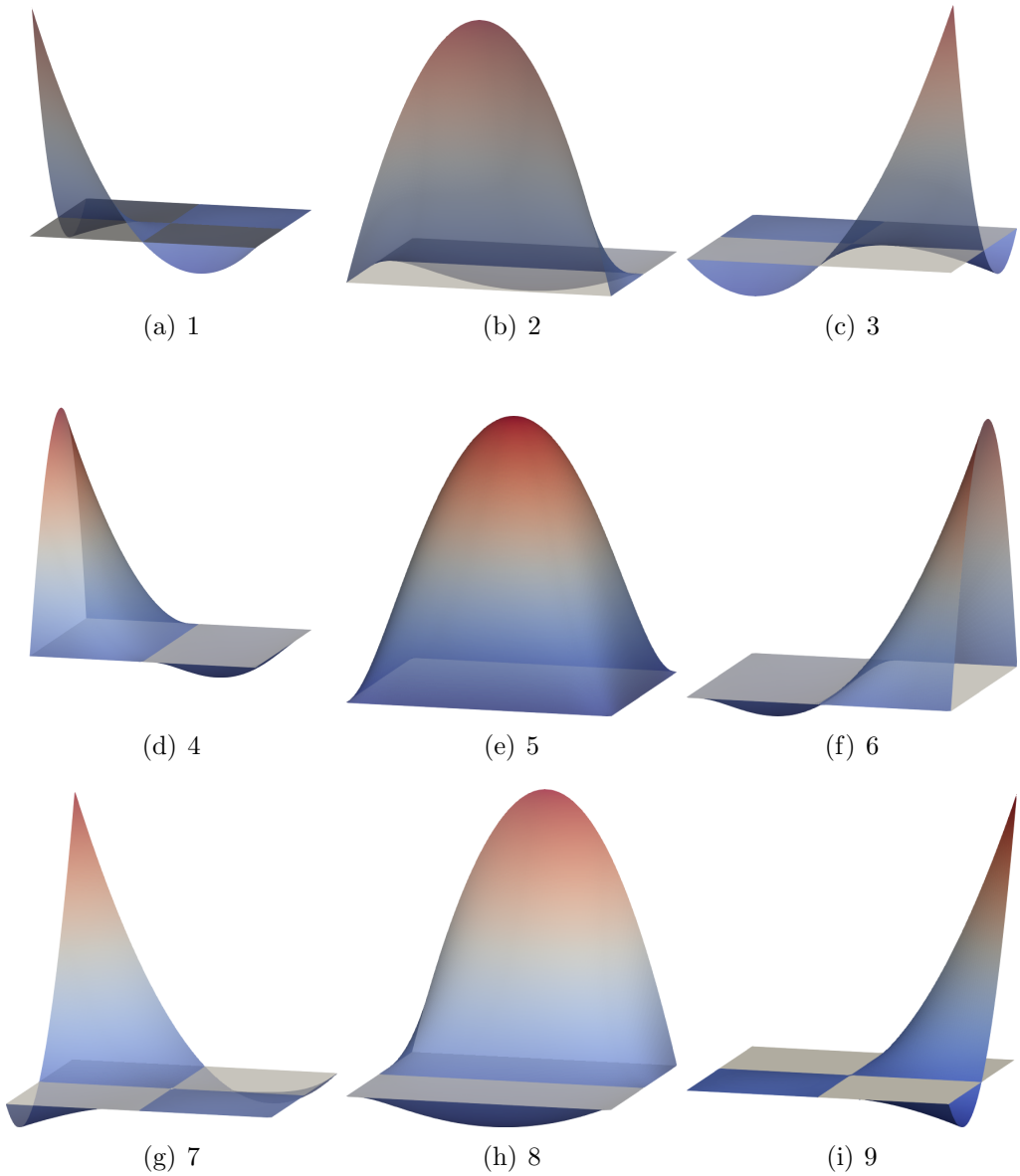


(c) 3



(d) 4

Figure 3.8: Shape functions for the two-dimensional bilinear finite element space $\mathcal{Q}_1$ in the reference square. The label says which is the related Lagrangian node in the set $\mathcal{N}^{(1,1)}$ (see Figure 3.5).



(a) 1 (b) 2 (c) 3

(d) 4 (e) 5 (f) 6

(g) 7 (h) 8 (i) 9

Figure 3.9: Shape functions for the two-dimensional biquadratic finite element space $\mathcal{Q}_2$ in the reference square. The label says which is the related Lagrangian node in the set $\mathcal{N}^{(2,2)}$ (see Figure 3.5).

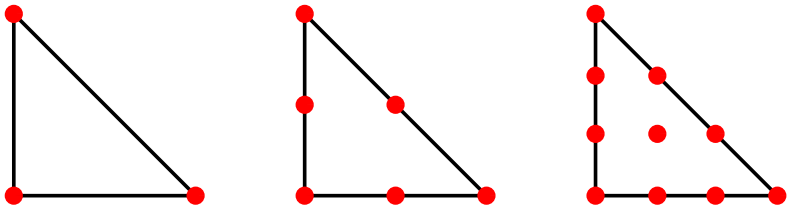


Figure 3.10: Equidistant nodes for two-dimensional Lagrangian polynomial spaces $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ obtained as tensor product of their one-dimensional counterparts.

we create the map as follows:

$$\boldsymbol{x} = \Phi_K(\hat{\boldsymbol{x}}) \doteq \sum_{a=1}^{n_\Sigma} \hat{\phi}_1^a(\hat{\boldsymbol{x}})\boldsymbol{x}^a,$$

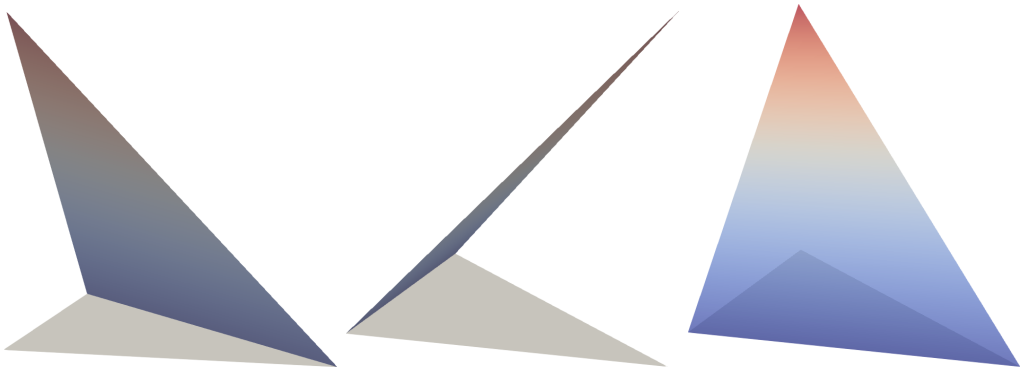
where $\left\{\hat{\phi}_1^a\right\}$ are the shape functions for $\hat{\mathcal{Q}}_1$. It is easy to check from the definition of shape functions and degrees of freedom for Lagrangian elements that Φ_K maps every vertex $\hat{\boldsymbol{x}}_a$ in the reference cell $\hat{K}$ to the corresponding vertex $\boldsymbol{x}_a$ in the physical cell K.

Using what we already know for quadrilateral finite elements, $\boldsymbol{x} \in \hat{Q}_1$; e.g., in two dimensions the physical coordinates can be expressed as a bilinear polynomial in terms of the reference coordinates. As a result the map is nonlinear (due to the $\hat{x}\hat{y}$ term).[8]

Thus, the space spanned by $\{\phi^a\}$ is not the space $\mathcal{Q}_1$ in general. This is only true when the cell K can be expressed as a stretching in each direction plus a rotation of the reference cell $\hat{K}$. In this case, one can easily check that the geometrical map Φ_K is in fact linear and so its inverse, which can be expressed as $\Phi_K^{-1}(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \mathbf{b}$. Given the shape functions $\left\{\hat{\phi}_p^a\right\}$ of $\hat{\mathcal{Q}}_p$, it is easy to check that the physical shape functions

$$\{\phi^a(\boldsymbol{x})\} \doteq \left\{\hat{\phi}^a \circ \Phi_K^{-1}(\boldsymbol{x})\right\} = \left\{\hat{\phi}^a(\mathbf{A}\boldsymbol{x} + \mathbf{b})\right\}$$

[8] In general, the cell K will not have flat faces in three-dimensions because we cannot find a plane that contains four points unless they are aligned. In two dimensions, the edges will be straight because two distinct points define a line.



(a) 1 (b) 2 (c) 3

Figure 3.11: Shape functions for the two-dimensional linear finite element space $\mathcal{P}_1$ in the reference triangle. The label says which is the related Lagrangian node in $\tilde{\mathcal{N}}^1$ (see Figure 3.10).

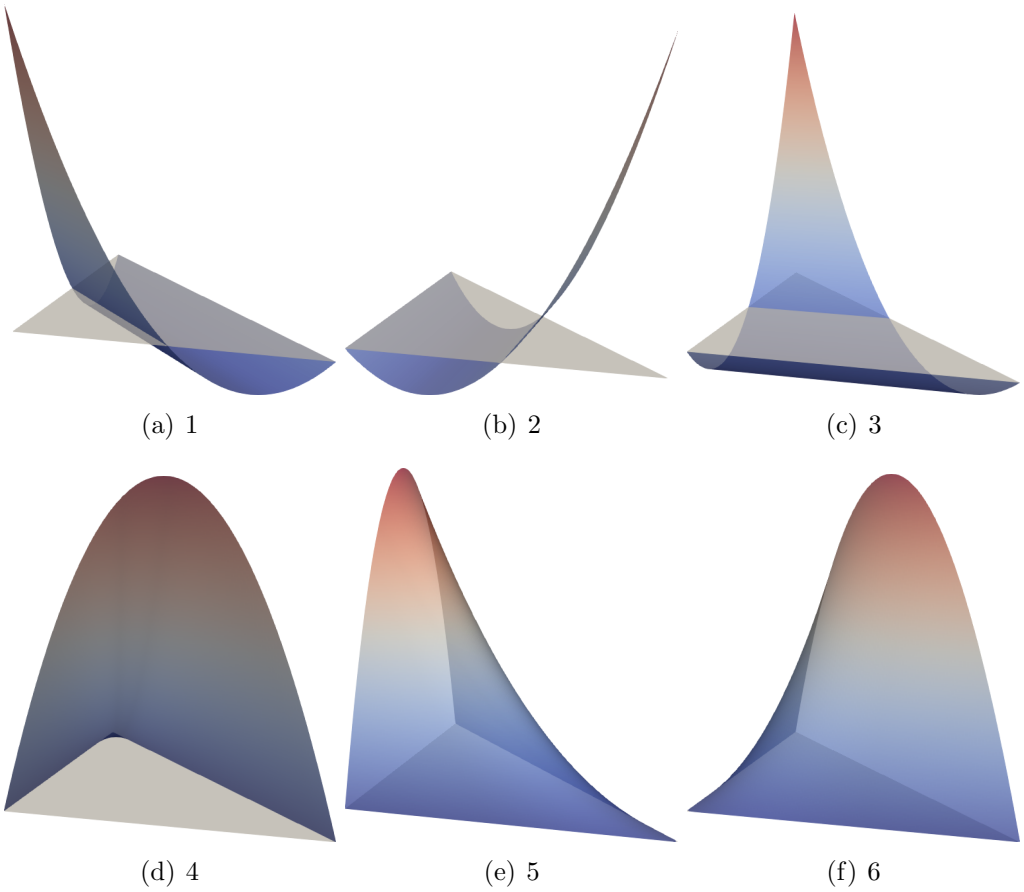
spans $\mathcal{Q}_p$.

3.5 Construction of the shape functions basis

The analytical expression of shape functions can become very complicated for high-order finite elements and non-trivial definitions of DOFs, e.g., for electromagnetic applications (see below). Furthermore, we require an automatic generator of shape functions to have a code that provides bases for arbitrarily high orders. When the explicit construction of the shape functions is not apparent, we proceed as follows.

Let us consider a finite element defined by $\{K, \mathcal{V}, \Sigma\}$.[9] First, we generate a *pre-basis* $\{\psi^b\}_{b\in\mathcal{N}_\Sigma}$ that spans the local finite element space $\mathcal{V}$, e.g., a Lagrangian polynomial basis (see Sect. 3.4). On the other hand, given the set of local DOFs, we proceed as follows. The shape functions can be written as $\phi^a = \sum_{b\in\mathcal{N}_\Sigma} \boldsymbol{\Xi}_{ab}\psi^b$, where ψ^b are the elements of the pre-basis. By definition, the shape functions must satisfy $\sigma_a(\phi^b) = \delta_{ab}$ for a, $b \in \mathcal{N}_\Sigma$. As a result, let

[9] In this section, we do not make difference between reference and physical spaces, e.g., using the $\hat{\cdot}$ symbol. In any case, all the following developments are usually performed at the reference finite element level.



(a) 1 (b) 2 (c) 3

(d) 4 (e) 5 (f) 6

Figure 3.12: Shape functions for the two-dimensional quadratic finite element space $\mathcal{P}_2$ in the reference triangle. The label says which is the related Lagrangian node in the set $\tilde{\mathcal{N}}^2$ (see Figure 3.10).

us define $\mathbf{C}_{ab} \doteq \sigma_a(\psi^b)$. We have (using Einstein's notation):

$$\sigma_a(\phi^b) = \sigma_a(\mathbf{\Xi}_{bc}\psi^c) = \sigma_a(\psi^c)\mathbf{\Xi}_{bc} = \delta_{ab},$$

or in compact form, $\mathbf{C}\mathbf{\Xi}^T = I$, and thus $\mathbf{\Xi}^T = \mathbf{C}^{-1}$. As a result, $\mathbf{\Xi}_{ab} = \mathbf{C}^{-1}_{ba}$. The shape functions are computed as a linear combination of the pre-basis functions.

3.6 Global finite element space and conformity

Now that we have defined the finite element spaces at the local level, we have to glue together these pieces keeping $\mathcal{C}^0$ continuity. In this section, we will consider how to build global (and conforming) finite element spaces. Next, we will learn how to integrate the bilinear forms in the corresponding weak formulation in Sect. 3.9.

Let us define the *global* finite element space. Conforming finite element spaces are defined as: $\mathcal{X}_h \doteq \{v \in \mathcal{X} \,:\, v|_K \in \mathcal{V}\}$. The main complication in this definition is to enforce the conformity of the finite element space, i.e., $\mathcal{X}_h \subset \mathcal{X}$. Since we want a conforming finite element space $\mathcal{X}_h \subset \mathcal{X}$, we must keep some continuity across inter-cell boundaries. E.g., for problems posed in $H^1(\Omega)$, it implies full continuity ($\mathcal{C}^0(\Omega)$ piecewise polynomials are in $H^1(\Omega)$).

In fact, the conformity constraint is the one that motivates the choice of $\hat{\Sigma}$ and Ψ, and as a consequence, Σ. In practice, the conformity constraint must be re-stated as a continuity constraint over finite element DOFs. For conforming meshes, these constraints are implicitly enforced via a global DOF numbering, even though it is not possible in general for adaptive schemes with non-conforming meshes or cell-dependent order, which require more involved constraints.

The idea of global DOF numbering is the following. We start with the cell-local finite element spaces we already know how to define. At every cell, we have a set of local DOFs. At the discrete level, we want to express that continuity in terms of DOFs, gluing together (enforcing the same values) on pairs of DOFs at different cells. Before providing the abstract definition, let us understand what it means in one dimension. In one dimension, let us consider two consecutive cell (segments) $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$. We can define a local Lagrangian basis of order p and its corresponding local $p+1$

nodes at every cell. In both cells, a node is geometrically located at x_i. If we do not enforce any continuity, functions could have a jump at x_i. To enforce continuity, we must enforce these two local nodes to have the same value. Conceptually, they represent the same *global* DOFs.

This idea can be stated in an abstract way. Let us define by $\mathcal{M}_h \doteq \{(b, K) : b \in \mathcal{N}_{\Sigma_K},\ K \in \mathcal{T}_h\}$ the Cartesian product of local DOFs for all cells. We define the global DOFs as the quotient space of $\mathcal{M}_h$ by an equivalence relation $\sim$. Using standard notation, given $\sim$, the equivalence class of $a \in \mathcal{M}_h$ with respect to $\sim$ is represented with $[a] \doteq \{b \in \mathcal{M}_h\ :\ a \sim b\}$, and the corresponding quotient set is $\mathcal{N}_h \doteq \{[a]\ :\ a \in \mathcal{M}_h\}$. The set $\mathcal{N}_h$ is the set of global DOF and $[\cdot]$ represents the *local-to-global* DOF map. Using the one-to-one mapping between DOFs and shape functions, the same operator allows one to define global shape functions $\phi^a = \sum_{(b,K)\sim a} \phi^b_K$. This construction should lead to conformity, i.e., $\mathcal{X}_h = \text{span}\{\phi^a\}_{a\in\mathcal{N}_h} \subset \mathcal{X}$.

For grad-conforming finite elements, the global finite element space is determined by the following equivalence relation. The set of local DOFs for n-cubes is $\mathcal{M}_h \doteq \{(\boldsymbol{s}, K)\ :\ \boldsymbol{s} \in \mathcal{N}^{k\mathbf{1}}, K \in \mathcal{T}_h\}$ due to the one-to-one mapping between DOFs and nodes; we replace the set of nodes by $\tilde{\mathcal{N}}^k$ for n-simplices. Furthermore, we say that $(\boldsymbol{s}, K) \sim (\boldsymbol{s}', K')$ iff $\boldsymbol{x}_{\boldsymbol{s}} = \boldsymbol{x}_{\boldsymbol{s}'}$, i.e., *two local degrees of freedom in two different cells are the same degree of freedom in the global space if their corresponding nodes are in the same spatial point in* Ω. The implementation of this equivalence relation, and thus, the global numbering, relies on the ownership relation between n-faces and DOFs. An illustration of the local-to-global map for grad-conforming finite elements in two dimensions is provided in Figure 3.13. In one dimension, we recover the *hat function* definition in the previous section.

With such global DOF definition, it is easy to check that the global finite element space is grad-conforming, i.e., $\mathcal{V} \subset \mathcal{C}^0(\overline{\Omega}) \subset H^1(\Omega)$. The idea is to check that, at any n-face (edge in two dimensions or face in two dimensions), the restriction of the finite element function to that n-face is a polynomial of the same order. The nodes on the closure of the n-face uniquely determine the polynomial on the n-face. Since these nodes are the same for all cells sharing the n-face, the function is continuous at the n-face.

3.7 Interpolant

Let us consider an infinite-dimensional space $\tilde{\mathcal{X}}$ such that 1) $\mathcal{X}_h \subset \tilde{\mathcal{X}} \subset \mathcal{X}$ and 2) for every function $v \in \tilde{\mathcal{X}}$ and global DOF $a \in \mathcal{N}_h$, all the local DOFs b, $b' \in [a]$ are such that $\sigma_b(v) = \sigma_{b'}(v)$, i.e., local DOFs related to the same global DOF are continuous among cells. The *global interpolator* is defined as:

$$\pi_{\mathcal{X}_h}(v) \doteq \sum_{K \in \mathcal{T}_h} \pi_K(v) = \sum_{K \in \mathcal{T}_h} \sum_{b \in \mathcal{N}_{\Sigma_K}} \sigma_b(v) \phi_K^b, \qquad \text{for } v \in \tilde{\mathcal{X}}. \tag{3.5}$$

Figure 3.4 shows this process at one physical cell, which for grad-conforming Lagrangian finite elements simply implies computing the nodal values (local degrees of freedom) at each cell and using the local-to-global map. It is easy to check that it is, in fact, a projector (using the definition of shape function). In any case, we use *projection operator* to refer to other projectors that involve the solution of a global finite element system, e.g., based on the minimization of the L^2 or H^1 norm.

Since Lagrangian DOFs involve point-wise evaluations of functions and $H_0^1(\Omega) \not\subset \mathcal{C}^0(\Omega)$ for $d > 1$, the interpolator (3.5) is not defined in such space. This is only true in one-dimension, the case considered in the previous section. Instead, we consider that functions to be interpolated belong, e.g., to the space $\tilde{\mathcal{X}} \doteq \mathcal{C}^0(\Omega)$. There are slightly more involved interpolators for grad-conforming finite element spaces that are bounded in $H^1(\Omega)$ for any space dimension but we do not consider them here for simplicity.

3.8 Assembly and linear system

Once we have defined a basis for the finite element space $\mathcal{X}_h$ using the finite element machinery presented above, every finite element function u_h can be uniquely represented by a vector $\mathbf{u} \in \mathbb{R}^{|\mathcal{N}_h|}$ as $u_h = \sum_{b \in \mathcal{N}_h} \phi^b \mathbf{u}_b$. In fact, problem (3.3) can be re-stated as: find $\mathbf{u} \in \mathbb{R}^{|\mathcal{N}_h|}$ such that

$$a(\phi^j, \phi^i)\mathbf{u}_j = \ell_h(\phi^i), \qquad \text{for any } \; i \in \mathcal{N}_h.$$

We have ended up with a finite-dimensional linear problem, i.e., a linear system. In matrix form, the problem can be stated as:

$$\text{Solve} \quad \mathbf{A}\mathbf{u} = \mathbf{f}, \qquad \text{with} \quad \mathbf{A}_{ij} \doteq a(\phi^j, \phi^i), \quad \mathbf{f}_i \doteq \ell_h(\phi^i).$$

Assuming that the bilinear form can be split into cell contributions as $a(\cdot,\cdot) = \sum_{K\in\mathcal{T}_h} a_K(\cdot,\cdot)$, e.g., by replacing $\int_\Omega$ by $\sum_{K\in\mathcal{T}_h}\int_K$, the construction of the matrix is implemented through a *cell-wise assembly process*[10], as follows:

$$\mathbf{A}_{[i][j]} = \sum_{K\in\mathcal{T}_h}\sum_{i,j\in\mathcal{N}_{\Sigma_K}} \mathbf{A}^K_{ij} \doteq \sum_{K\in\mathcal{T}_h}\sum_{i,j\in\mathcal{N}_{\Sigma_K}} a_K(\phi^j_K,\phi^i_K).$$

We remind that $[i]$ represents the global index for the local (cell) index i. The finite element affine operator (3.4) can be represented as $\mathcal{F}_h(u_h) \doteq \mathbf{Au}-\mathbf{f}$, i.e., it can be represented with a matrix and a vector of size $|\mathcal{N}_h|$.

Let us consider a practical example of this process. Let us assume that we are integrating the entries for cell K_2 in Figure 3.13. Thus, the element-local matrix reads:

$$\mathbf{A}^K \doteq \begin{bmatrix} a(\phi^1_{K_2},\phi^1_{K_2}) & a(\phi^1_{K_2},\phi^2_{K_2}) & a(\phi^1_{K_2},\phi^3_{K_2}) & a(\phi^1_{K_2},\phi^4_{K_2}) \\ a(\phi^2_{K_2},\phi^1_{K_2}) & a(\phi^2_{K_2},\phi^2_{K_2}) & a(\phi^2_{K_2},\phi^3_{K_2}) & a(\phi^2_{K_2},\phi^4_{K_2}) \\ a(\phi^3_{K_2},\phi^1_{K_2}) & a(\phi^3_{K_2},\phi^2_{K_2}) & a(\phi^3_{K_2},\phi^3_{K_2}) & a(\phi^3_{K_2},\phi^4_{K_2}) \\ a(\phi^4_{K_2},\phi^1_{K_2}) & a(\phi^4_{K_2},\phi^2_{K_2}) & a(\phi^4_{K_2},\phi^3_{K_2}) & a(\phi^4_{K_2},\phi^4_{K_2}) \end{bmatrix}$$

Now, we have to assemble the local matrix in the global one using the local to global map. The local degrees of freedom 1, 2, 3 and 4 correspond to the global ones 2, 3, 5 and 6. Thus, these entries will be added to the corresponding entries (in red) of the global matrix

$$\mathbf{A}+= \begin{bmatrix} \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \bigcirc & \color{red}{\bullet} & \color{red}{\bullet} & \bigcirc & \color{red}{\bullet} & \color{red}{\bullet} & \bigcirc & \bigcirc & \bigcirc \\ \bigcirc & \color{red}{\bullet} & \color{red}{\bullet} & \bigcirc & \color{red}{\bullet} & \color{red}{\bullet} & \bigcirc & \bigcirc & \bigcirc \\ \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \bigcirc & \color{red}{\bullet} & \color{red}{\bullet} & \bigcirc & \color{red}{\bullet} & \color{red}{\bullet} & \bigcirc & \bigcirc & \bigcirc \\ \bigcirc & \color{red}{\bullet} & \color{red}{\bullet} & \bigcirc & \color{red}{\bullet} & \color{red}{\bullet} & \bigcirc & \bigcirc & \bigcirc \\ \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \end{bmatrix}.$$

The operator $+=$ means add the object on the right to the one on the left.

[10] The implementation of finite element methods is cell-based, i.e., all the computations are at the cell level, and then, using the local-to-global map (the equivalence class) assembled in the global linear system. This is different from finite difference methods, that do not share the concept of cell and are nodal-based.

3.9 Numerical integration

In general, the local bilinear form can be stated as:

$$a_K(\phi_K^b, \phi_K^a) \doteq \int_K \boldsymbol{\mathcal{I}}(\boldsymbol{x})\mathrm{d}\Omega,$$

for some integrand $\boldsymbol{\mathcal{I}}(x)$, where the evaluation of $\boldsymbol{\mathcal{I}}(\boldsymbol{x})$ involves the evaluation of shape function derivatives. Let us represent the Jacobian of the geometrical mapping with $\boldsymbol{J}_K \doteq \frac{\partial \boldsymbol{\Phi}_K}{\partial \boldsymbol{x}}$. We can rewrite the cell integration in the reference cell, and next consider a quadrature rule Q defined by a set of points/weights $(\hat{\boldsymbol{x}}_{\mathrm{gp}}, \mathrm{w}_{\mathrm{gp}})$, as follows:

$$\int_K \boldsymbol{\mathcal{I}}(\boldsymbol{x})\mathrm{d}\Omega = \int_{\hat{K}} \boldsymbol{\mathcal{I}} \circ \boldsymbol{\Phi}_K(\hat{\boldsymbol{x}})|\boldsymbol{J}_K|\mathrm{d}\hat{\Omega} = \sum_{\hat{\boldsymbol{x}}_{\mathrm{gp}} \in \mathrm{Q}} \boldsymbol{\mathcal{I}} \circ \boldsymbol{\Phi}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}})\mathrm{w}(\hat{\boldsymbol{x}}_{\mathrm{gp}})|\boldsymbol{J}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}})|.$$

Here, the main complication is the evaluation of $\boldsymbol{\mathcal{I}} \circ \boldsymbol{\Phi}_k(\hat{\boldsymbol{x}}_{\mathrm{gp}})$. The evaluation of this functional requires the evaluation of $\partial_{\boldsymbol{\alpha}}\phi_K^b \circ \boldsymbol{\Phi}_k(\hat{\boldsymbol{x}}_{\mathrm{gp}})$ for some values of the multi-index $\boldsymbol{\alpha}$ (idem for the test functions). Usually, $|\boldsymbol{\alpha}| \leq 1$ in $\mathcal{C}^0$ finite elements, since higher-order derivatives would require higher inter-cell continuity.[11]

Let us consider the case of zero and first derivatives, i.e., the evaluation of $\phi_K^b \circ \boldsymbol{\Phi}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}})$ and $\boldsymbol{\nabla}\phi_K^b \circ \boldsymbol{\Phi}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}})$. $\boldsymbol{\nabla}_{\hat{\boldsymbol{x}}}$ represents the gradient in the reference space. The values of the shape functions (times the geometrical mapping) on the quadrature points is determined as follows:

$$\phi_K^b \circ \boldsymbol{\Phi}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}}) = \hat{\phi}^b(\hat{\boldsymbol{x}}_{\mathrm{gp}}),$$

which is simply $\hat{\phi}^b(\hat{\boldsymbol{x}}_{\mathrm{gp}})$ for grad-conforming (Lagrangian) finite elements, whereas shape function gradients are computed as:

$$\begin{aligned}\boldsymbol{\nabla}\phi_K^b \circ \boldsymbol{\Phi}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}}) &= \boldsymbol{\nabla}(\hat{\phi}^b \circ \boldsymbol{\Phi}_K^{-1}) \circ \boldsymbol{\Phi}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}}) \\ &= \boldsymbol{\nabla}_{\hat{\boldsymbol{x}}}\hat{\phi}^b(\hat{\boldsymbol{x}}_{\mathrm{gp}}) \cdot \boldsymbol{J}_K^{-1}(\hat{\boldsymbol{x}}_{\mathrm{gp}}),\end{aligned}$$

where we have used some elementary differentiation rules and the inverse function theorem in the last equality:

$$\boldsymbol{\nabla} f(x) = \boldsymbol{\nabla}\left(\hat{f} \circ \Phi_K^{-1}\right) = \boldsymbol{\nabla}_{\hat{\boldsymbol{x}}}\hat{f} \cdot \boldsymbol{\nabla}\Phi_K^{-1} = \boldsymbol{\nabla}_{\hat{\boldsymbol{x}}} f \cdot \boldsymbol{J}_K^{-1} \circ \Phi_K^{-1}.$$

[11] There exist $\mathcal{C}^1$ finite element methods that can approximate fourth order problems, e.g., the bi-harmonic problen, but we are not going to consider them here.

Using matrix notation, the term $\boldsymbol{\nabla}_{\hat{\boldsymbol{x}}} f \cdot \boldsymbol{J}_K^{-1}$ is usually written as $\boldsymbol{J}_K^{-T} \boldsymbol{\nabla}_{\hat{\boldsymbol{x}}} f$ Thus, one only needs to provide the values of the Jacobian matrix, its inverse, and its determinant from one side, and the value of the shape functions $\hat{\phi}^b$ and their gradients $\boldsymbol{\nabla}_{\hat{\boldsymbol{x}}} \hat{\phi}^b$ in the reference space, on the other side, at all quadrature points, to compute all the entries of the finite element matrices; second-order derivatives can be treated analogously.

As an example, a mass matrix[12] would be computed as

$$\begin{aligned}\int_K \phi^a(\boldsymbol{x})\,\phi^b(\boldsymbol{x})\,\mathrm{d}\Omega &= \int_{\hat{K}} \hat{\phi}^a(\hat{\boldsymbol{x}})\,\hat{\phi}^b(\hat{\boldsymbol{x}})\,|\boldsymbol{J}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}})|\mathrm{d}\hat{\Omega} \\ &= \sum_{\hat{\boldsymbol{x}}_{\mathrm{gp}}\in \mathrm{Q}} \hat{\phi}^a(\hat{\boldsymbol{x}}_{\mathrm{gp}})\,\hat{\phi}^b(\hat{\boldsymbol{x}}_{\mathrm{gp}})\,\mathrm{w}(\hat{\boldsymbol{x}}_{\mathrm{gp}})|\boldsymbol{J}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}})|.\end{aligned}$$

On the other hand, the Laplacian matrix would be computed as

$$\begin{aligned}&\int_K \boldsymbol{\nabla}\phi^a(\boldsymbol{x}) \cdot \boldsymbol{\nabla}\phi^b(\boldsymbol{x})\,\mathrm{d}\Omega \\ &= \int_{\hat{K}} [\boldsymbol{J}_K^{-T}\boldsymbol{\nabla}_{\hat{\boldsymbol{x}}}\hat{\phi}^a](\hat{\boldsymbol{x}}) \cdot [\boldsymbol{J}_K^{-T}\boldsymbol{\nabla}_{\hat{\boldsymbol{x}}}\hat{\phi}^b](\hat{\boldsymbol{x}})\,|\boldsymbol{J}_K(\hat{\boldsymbol{x}})|\mathrm{d}\hat{\Omega} \\ &= \sum_{\hat{\boldsymbol{x}}_{\mathrm{gp}}\in \mathrm{Q}} [\boldsymbol{J}_K^{-T}\boldsymbol{\nabla}_{\hat{\boldsymbol{x}}}\hat{\phi}^a](\hat{\boldsymbol{x}}_{\mathrm{gp}}) \cdot [\boldsymbol{J}_K^{-T}\boldsymbol{\nabla}_{\hat{\boldsymbol{x}}}\hat{\phi}^b](\hat{\boldsymbol{x}}_{\mathrm{gp}})\mathrm{w}(\hat{\boldsymbol{x}}_{\mathrm{gp}})|\boldsymbol{J}_K(\hat{\boldsymbol{x}}_{\mathrm{gp}})|.\end{aligned}$$

Quadrature rules for $\hat{K}$ being an n-cube can be obtained as a tensor product of a 1D quadrature rule, e.g., the Gauss-Legendre quadrature. As it is well known, considering n-cube topologies for $\hat{K}$, Gauss quadratures with n points per direction can integrate *exactly* $2n-1$ order polynomials. E.g., for a Lagrangian reference finite element of order p and an affine geometrical map, we choose $n = p+\mathrm{ceiling}(1/2) = p+1$ per direction to integrate exactly a mass matrix.

Symmetric quadrature rules on triangles and tetrahedra for different orders can also be constructed. In any case, to create arbitrarily large quadrature rules for n-simplices, one can consider the so-called Duffy transformation. The latter is a change of variables that transform our n-simplex integration domain into an n-cube, and integrate on the n-cube using tensor product quadratures.

[12] The mass matrix is the one that arises from a zero-order (a.k.a. reaction) term, e.g., the first term in $u - \Delta u = 0$. In weak form, it leads to a term $\int_\Omega uv\mathrm{d}\Omega$. The origin of the name comes from the fact that this term arises after the finite difference discretisation of a time derivative, and the time derivative term is the so-called inertia term.

Let us finally mention that in some instances, e.g., for integration Neumann boundary condition terms of the type $\int_{\Gamma_N} h(x)v(x)\mathrm{d}F$ we can use the same ideas above. Instead of integrating over the cell, we would integrate on the faces. We can consider a reference facet $\hat{F}$, and a mapping $\boldsymbol{\Phi}_F : \hat{F} \to F$ from the reference to the physical space. Let us represent the Jacobian of the geometrical mapping with $\boldsymbol{J}_F \doteq \frac{\partial \boldsymbol{\Phi}_F}{\partial \boldsymbol{x}}$, which has values in $\mathbb{R}^{(d-1)\times d}$. We can rewrite the facet integral in the reference facet, and next consider a quadrature rule Q on $\hat{F}$ defined by a set of points/weights $(\hat{\boldsymbol{x}}_{\mathrm{gp}}, \mathrm{w}_{\mathrm{gp}})$, as follows:

$$\int_F \boldsymbol{\mathcal{I}}(\boldsymbol{x})\mathrm{d}\Omega = \int_{\hat{F}} \boldsymbol{\mathcal{I}} \circ \boldsymbol{\Phi}_F(\boldsymbol{x})|\boldsymbol{J}_F|\mathrm{d}F = \sum_{\hat{\boldsymbol{x}}_{\mathrm{gp}}\in \mathrm{Q}} \boldsymbol{\mathcal{I}} \circ \boldsymbol{\Phi}_F(\hat{\boldsymbol{x}}_{\mathrm{gp}})\mathrm{w}(\hat{\boldsymbol{x}}_{\mathrm{gp}})|\boldsymbol{J}_F(\hat{\boldsymbol{x}}_{\mathrm{gp}})|.$$

$|\boldsymbol{J}_F|$ is defined as:

$$|\boldsymbol{J}_F| = \left\| \frac{\mathrm{d}\boldsymbol{\Phi}_F}{\mathrm{d}\hat{\boldsymbol{x}}} \right\|_2 \quad \text{and} \quad |\boldsymbol{J}_F| = \left\| \frac{\partial \boldsymbol{\Phi}_F^1}{\partial \hat{\boldsymbol{x}}} \times \frac{\partial \boldsymbol{\Phi}_F^2}{\partial \hat{\boldsymbol{x}}} \right\|_2,$$

for $d = 2, 3$, respectively. The facet map can easily be computed using similar ideas as in Sect. 3.4.3. The idea is to consider the shape functions related to the nodes on the closure of $\hat{F}$ their position in the physical space:

$$\boldsymbol{\Phi}_F(\hat{\boldsymbol{x}}) \doteq \sum_{a\in\overline{\hat{F}}} \hat{\phi}_1^a \boldsymbol{x}^a,$$

where $\left\{\hat{\phi}_1^a\right\}$ are the shape functions for $\hat{\mathcal{Q}}_1$.

3.10 Grad-conforming finite elements for vector fields

When one has to deal with vector or tensor fields, we can generate them as a Cartesian product of scalar spaces as follows. We define the local finite element space $\boldsymbol{\mathcal{V}}_k \doteq [\mathcal{Q}_k]^d$ (or $[\mathcal{P}_k]^d$). The local degrees of freedom are nodal values for a given component, i.e.,

$$\sigma_a^\alpha(\boldsymbol{u}) = \boldsymbol{u}_\alpha(\boldsymbol{x}_a) \qquad \text{for } a = 1, \ldots, n_\Sigma, \quad \alpha = 1, \ldots, d.$$

The equivalence class for the local-to-global map is such that two local degrees of freedom represent the same global one if the nodes are at the same

point in the physical space and they are related to the same component. Analogously, shape functions are computed as $\phi^a \doteq \sum_{(i,\boldsymbol{s},K)\sim a} \ell_{\boldsymbol{s}}^{k\mathbf{1}} \vec{\boldsymbol{e}}_i$; $\vec{\boldsymbol{e}}_i$ represents the i-th canonical basis vector of $\mathbb{R}^d$. E.g., for a two-dimensional vector field, the bi-linear local finite element space reads

$$\boldsymbol{\mathcal{V}} \doteq \left\{ \left(\phi^1, 0\right)^T, \ldots, \left(\phi^4, 0\right)^T, \left(0, \phi^1, \right)^T \ldots, \left(0, \phi^4\right)^T \right\}$$

We proceed analogously for n-simplices.

3.11 Cartesian product of finite elements for multi-field problems

Many problems governed by PDEs involve more than one field, e.g., the Navier-Stokes equations or any multi-physics problem. Let us consider a PDE that involves a set of unknown fields $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n) \in \mathcal{X}^1 \times \ldots \times \mathcal{X}^n$, defined as the Cartesian product of functional spaces. We can proceed as above, and define a finite element space for every field space separately, leading to a global finite element space $\mathcal{X}_h^1 \times \ldots \times \mathcal{X}_h^n$ defined by composition of finite element spaces. To define the global numbering of DOFs in the multi-field case, we consider that two DOFs are equivalent if they are related to the same field and satisfy the equivalence relation of the finite element space of this field.

The Cartesian product of finite element spaces is enough to define volume-coupling multi-physics problems governed on the same physical domain, i.e., the different physics are defined on the whole domain and coupled through volume terms in the formulation. However, many multi-physics problems are interface-based, i.e., the coupling between different physics that are defined on different subdomains is through transmission conditions on the interface. It is the case, e.g., of fluid-structure problems. In these cases, different finite element spaces could be defined on different parts of the global mesh, i.e., one must describe the set of subdomains $(\Omega_1, \ldots, \Omega_n)$ of the whole domain Ω in which the corresponding finite element spaces are defined and enforce the *transmission* conditions on the interface (usually, continuity of the unknown and its flux).

3.12 Approximation properties

The same error estimates we have proven for one-dimensional spaces hold in the multi-dimensional case. The proof is more technical and is out of this course scope. With these results about the approximability properties of the polynomial spaces considered so far, we can readily obtain bounds for the error committed by the finite element method by recalling Cea's lemma in the previous section.

Before showing error estimates, we need some properties for the family of meshes being used in the h-refinement process. As in the previous chapter, let us consider a parameter N that determines the level of refinement in our mesh; e.g., N can represent the number of uniform refinements through bisection for a given initial mesh $\mathcal{T}_0$.

Definition 3.12.1: Shape regular and quasi-uniform mesh

A family of meshes (a.k.a. triangulations) $\{\mathcal{T}_N\}$ is shape regular if there exists a positive constant $c > 0$ independent of N such that

$$\max_{\tau \in \mathcal{T}_N} \frac{\operatorname{diam}(\tau)^d}{|\tau|} \leq c, \qquad \forall N.$$

We can define the cell size $h_\tau = |\tau|^{\frac{1}{d}}$ for $\tau \in \mathcal{T}_N$. The family is also quasi-uniform if

$$\frac{\max_{\tau \in \mathcal{T}_N} |\tau|}{\min_{\tau \in \mathcal{T}_N} |\tau|} \leq \rho, \qquad \forall N.$$

For quasi-uniform families, we can define a mesh width $h_N \doteq \max_{\tau \in \mathcal{T}_N} h_\tau$.

Shape regularity means cells cannot be arbitrarily anisotropic as we increase refinement. E.g., Let us consider $\mathcal{T}_0$ to be a rectangle. At each level of refinement rectangles are split into two rectangles by a horizontal cut. This way, we can keep conforming meshes at all levels, but as $N \to \infty$ the vertical dimension of the cell goes to zero, whereas the horizontal remains constant. Thus, such a family of meshes is not shape-regular. Shape regularity prevents *flat* elements as $N \to \infty$ and permits the definition of a meaningful length size. Meshes constructed by uniform refinement in all directions, e.g., splitting the rectangle into four scaled rectangles makes the shape regularity

coefficient constant with respect to N.

Quasi-uniformity means that the size of all the cells in a mesh must be *similar*. We cannot refine some cells while keeping others untouched. It gives us a meaningful definition of mesh width in the multidimensional case. Quasi-uniformity is an essential ingredient for the statement of the multi-dimensional interpolation error estimates.

Lemma 3.12.2: Error bounds for the interpolant

Given a finite element space of order p $\mathcal{Q}^p$ (for quad meshes) or $\mathcal{P}^p$ (for tet meshes) on a family of quasi-uniform meshes with mesh width h, the interpolant π^p holds:

$$\|v - \pi^p(v)\|_{L^2(\Omega)} \leq h^{p+1}|v|_{H^{p+1}(\Omega)}, \qquad |v - \pi^p(v)|_{H^1(\Omega)} \leq h^p|v|_{H^{p+1}(\Omega)},$$

for any $v \in H^{p+1}(\Omega)$.

3.13 Tutorials

1. Consider a linear reference finite element $[-1, 1]^2$, use as prebasis of the local space the space of monomials that span $\mathcal{Q}_1$, and apply the change of basis to get the expression of the shape functions.

2. Let us consider the problem $-\Delta u = f$ on $\Omega \subset (0, 1)^2$. Can you compute the shape function gradients at the reference finite element $[-1, 1]^2$ space for a linear finite element (using the result from the previous exercise)? Can you compute the corresponding Jacobian in element $(0.0, 0.5)^2$? Can you compute the linear system matrix in that element? Let us consider the element with vertices $(0.0, 0.0)$, $(0.5, 0.0)$, $(0.0, 0.5)$ and $(0.6, 0.5)$. Can you compute the Jacobian in this case?

3. Consider a structured 2×2 uniform mesh of a unit square with four bilinear finite elements. Can you compute the global system matrix using this local-to-global numbering and the matrix in the previous exercise? Now consider biquadratic finite elements on the same mesh, i.e., $\mathcal{Q}_2$. Provide a local and global numbering using a drawing.

4. Can you prove why grad-conforming Lagrangian finite elements lead to continuous solutions across cells after *gluing* DOFs?

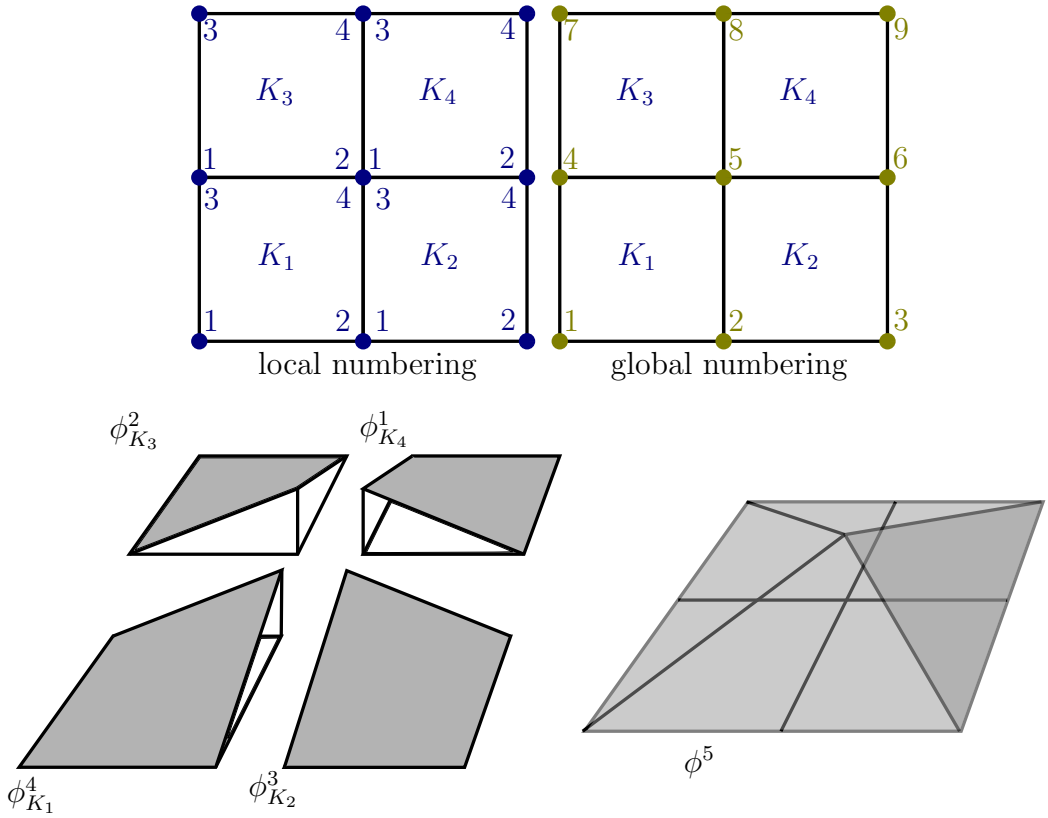


Figure 3.13: In the top-left corner we show the local numbering for each cell in a mesh. The top-right corner shows the global numbering for the same mesh. We can observe, e.g., that the local degrees of freedom (nodes) $\sigma^4_{K_1}$, $\sigma^3_{K_2}$, $\sigma^2_{K_3}$ and $\sigma^1_{K_4}$ correspond to the same global degree of freedom (node) σ_5. In the bottom-left part we show the corresponding local shape functions for these local degrees of freedom and the global shape function corresponding to the global degree of freedom.

Chapter 4

Singularly-perturbed problems

So far, we have applied the FE method to problems governed by elliptic PDEs, e.g., the Poisson equation, with very satisfactory results. Elliptic problems are related to coercive and continuous bilinear forms in the weak statement of the problem, as we have previously seen. Now, we focus on hyperbolic PDEs or so-called singular limits of elliptic PDEs. We note these problems are prevalent in physics, e.g., in fluid mechanics or transport problems.

Unfortunately, the Galerkin approximation of hyperbolic problems with standard finite element methods does not yield satisfactory results. Hyperbolic problems usually have solutions with jumps (or discontinuities). Furthermore, standard FE methods exhibit large oscillations in these cases. Fortunately, there are different ways to overcome these shortcomings. In this chapter, we will introduce so-called *stabilisation* techniques that add additional terms to the numerical approximation to recover optimal stability and convergence properties in these situations.

4.1 The convection-diffusion problem

In this section, we consider the convection-diffusion-reaction equation, which in strong form reads:

$$\begin{aligned} \boldsymbol{\nabla} \cdot (\boldsymbol{\beta} u) - \boldsymbol{\nabla} \cdot \mu \boldsymbol{\nabla} u + \sigma u &= f, && \text{in } \Omega, \\ u &= g, && \text{on } \Gamma_D, \\ \boldsymbol{n} \cdot \boldsymbol{\nabla} u &= q, && \text{on } \Gamma_N. \end{aligned} \tag{4.1}$$

This problem governs the concentration $u(\boldsymbol{x})$ of a contaminant, which is transported by chemical diffusion (μ being the diffusion coefficient), convective transport (e.g., the fluid velocity $\boldsymbol{\beta}$) and possibly changes due to chemical reactions (σ being the reaction term). f is a forcing term (a well or sink), g is the prescribed Dirichlet value on the boundary, and q a prescribed flux on the Neumann boundary.

The problem above is of mixed hyperbolic-parabolic nature, depending on the relative value of the physical coefficients ($\boldsymbol{\beta}$, μ, σ). When μ is *dominant*, it is an elliptic problem for which the FE methods work well. However, when the diffusive term vanishes, the problem is purely hyperbolic. We will see in this section that standard FE methods do not work well for this kind of PDEs. Thus, the limit of zero viscosity is problematic because, as $\mu \to 0$, the nature of the problem changes from elliptic to hyperbolic.

Let us consider the purely hyperbolic limit ($\mu = 0$) and assume that $\boldsymbol{\nabla} \cdot \boldsymbol{\beta} = 0$ (divergence-free convective field). We have the following equation:

$$(\boldsymbol{\beta} \cdot \boldsymbol{\nabla})u + \sigma u = f, \qquad \text{in } \Omega. \tag{4.2}$$

where $(\boldsymbol{\beta} \cdot \boldsymbol{\nabla}) \doteq \boldsymbol{\nabla}_{\boldsymbol{\beta}}$ is the directional derivative in the direction of β. Dirichlet boundary conditions for the hyperbolic problem can only be enforced in the inflow:

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on} \quad \Gamma_-, \qquad \Gamma_- \doteq \{\boldsymbol{x} \in \partial\Omega \ : \ \boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{\beta} < 0\}\,.$$

In order to provide some intuition about the solution of this equation, let us introduce the concept of *characteristics*. Given $\boldsymbol{x}_- \in \Gamma_-$, we can define the characteristic $\xi(\boldsymbol{x}_-)$ as follows. Let us consider a curve (the *characteristic*) $\boldsymbol{X}(s)$ for a given parameter $s \in [0, +\infty)$ such that:

$$\boldsymbol{X}(0) = \boldsymbol{x}_-, \qquad \frac{d\boldsymbol{X}}{ds} = \beta(\boldsymbol{X}(s)).$$

Using the chain rule, we readily get

$$\frac{du}{ds} = \frac{du}{d\boldsymbol{X}}\frac{d\boldsymbol{X}}{ds} = \boldsymbol{\beta} \cdot \boldsymbol{\nabla}(u) = f - \sigma u.$$

Thus, the solution can be computed independently for each characteristic. As a result, if the inflow data g is discontinuous, the solution u is discontinuous across characteristics.

When we add a diffusive term $-\boldsymbol{\nabla}\cdot\mu\boldsymbol{\nabla}u$ to (4.2) with $\mu>0$, the solution is in $H^1(\Omega)$ and cannot be discontinuous. In this case, diffusion smears out discontinuities on g across characteristics on regions of width $\mathcal{O}(\mu^{1/2})$, to make the solution continuous in Ω. Besides, when $\mu>0$, one can impose Dirichlet boundary conditions on the outflow $\Omega_+ \doteq \partial\Omega\setminus\Gamma_-$. If the boundary data is not compatible with the purely hyperbolic solution, it will also produce a *boundary layer* of size $\mathcal{O}(\mu)$.

From the discussion above, one can observe that the hyperbolic and diffusive regimes have a different nature. On one side, hyperbolic problems admit solutions with shocks and inflow-only Dirichlet boundary conditions. On the other side, diffusive problems (also called viscous, due to the similar nature of viscosity in fluids) cannot exhibit discontinuities and admit Dirichlet boundary conditions on the outflow. The limit $\mu\to 0$ is called a *singularly perturbed limit*, in which the nature of the underlying PDE changes. As one can expect, approximating problems in this limit will be challenging.

4.2 The Galerkin method for convection dominated problems

In this section, we propose the Galerkin discretisation of (4.1) using standard FE methods. We will analyse the stability and convergence properties of the method. The convergence analysis will provide a clear insight into why this discretisation is unsuitable for convection-dominated problems. In the following, we will consider no reaction ($\sigma=0$) for conciseness and leave this extension as an exercise.

For simplicity, we assume homogeneous Dirichlet conditions, i.e., $g=0$, $\Gamma_D=\partial\Omega$ and $\Gamma_N=\emptyset$. The extension to other boundary conditions can be considered as explained in the previous sections. The weak formulation reads:

$$u\in H^1_0(\Omega): \int_\Omega \mu\boldsymbol{\nabla}u\cdot\boldsymbol{\nabla}v\mathrm{d}\Omega+\int_\Omega v(\sigma u+\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u)\mathrm{d}\Omega=\int_\Omega fv\mathrm{d}\Omega, \quad (4.3)$$

for all $v\in H^1_0(\Omega)$. The first result we want to prove is the stability of the continuous problem. In the following, we assume for simplicity that $\mu>0$ and $\sigma\geq 0$ are constant and $\boldsymbol{\nabla}\cdot\boldsymbol{\beta}=0$.

Theorem 4.2.1: Stability of convection-diffusion problem

The solution u of (4.3) satisfies the following stability bound:

$$\frac{\mu}{2}|u|^2_{H^1(\Omega)} + \sigma\|u\|^2_{L^2(\Omega)} \leq \frac{C_P^2}{2\mu}\|f\|^2_{H^{-1}(\Omega)}.$$

Proof. Let us take $v = u$ in (4.3). We have:

$$\int_\Omega \mu|\boldsymbol{\nabla} u|^2 \mathrm{d}\Omega + \int_\Omega \sigma u^2 \mathrm{d}\Omega + \int_\Omega u((\boldsymbol{\beta}\cdot\boldsymbol{\nabla} u))\mathrm{d}\Omega = \int_\Omega fv\mathrm{d}\Omega.$$

We can manipulate the convective term using integration by parts and the fact that v vanishes on the boundary as follows:

$$\int_\Omega u\boldsymbol{\beta}_i\partial_i u\mathrm{d}\Omega = -\int_\Omega \partial_i(u\boldsymbol{\beta}_i)u\mathrm{d}\Omega = -\int_\Omega (\boldsymbol{\nabla}\cdot\boldsymbol{\beta})u^2\mathrm{d}\Omega = 0.$$

The forcing term can be bounded using the duality norm, Poincaré inequality $\|u\|_{H^1(\Omega)} \leq C_P|u|_{H^1(\Omega)}$, and Young inequality:

$$\begin{aligned}&\int_\Omega fu\mathrm{d}\Omega \leq \mu^{1/2}\|f\|_{H^{-1}(\Omega)}\mu^{1/2}\|u\|_{H^1(\Omega)}\\ &\leq C_P\mu^{1/2}\|f\|_{H^{-1}(\Omega)}\mu^{1/2}|u|_{H^1(\Omega)} \leq \frac{C_P^2}{2\mu}\|f\|^2_{H^{-1}(\Omega)} + \frac{\mu}{2}|u|^2_{H^1(\Omega)}.\end{aligned}$$

Combining these results, we get:

$$\frac{\mu}{2}|u|^2_{H^1(\Omega)} + \sigma\|u\|^2_{L^2(\Omega)} \leq \frac{C_P^2}{2\mu}\|f\|^2_{H^{-1}(\Omega)}.$$

□

4.3 Galerkin approximation

The Galerkin approximation of the problem above can readily be obtained by replacing the functional spaces in (4.3) by discrete FE spaces:

$$u_h \in V_h : \int_\Omega \mu \boldsymbol{\nabla} u_h \cdot \boldsymbol{\nabla} v_h \mathrm{d}\Omega + \int_\Omega v_h(\sigma u_h + \boldsymbol{\beta} \cdot \boldsymbol{\nabla}) u_h) \mathrm{d}\Omega = \int_\Omega f v_h \mathrm{d}\Omega, \quad (4.4)$$

for all $v_h \in V_h$. The discrete problem satisfies the same bound as in Theorem 4.2.1. However, we have discussed in the introduction that the results obtained with this formulation are not accurate. In order to show why this is the case, let us analyse the error committed by the FE solution.

Convergence

Theorem 4.3.1: Error estimates for convection-diffusion-reaction

The solution u of (4.3) and the FE solution u_h of (4.4)

$$\begin{aligned} & C(\mu |u - u_h|^2_{H^1(\Omega)} + C\sigma \|u - u_h\|^2_{L^2(\Omega)}) \\ & \leq \mu h^{2(r-1)} |u|^2_{H^r(\Omega)} + \sigma h^{2r} |u|^2_{H^1(\Omega)} \\ & \quad + \min \left(\frac{\|\boldsymbol{\beta}\|^2_{L^\infty(\Omega)}}{\sigma^2} \sigma h^{2(r-1)} \|u\|^2_{H^r(\Omega)}, \frac{\|\boldsymbol{\beta}\|^2_{L^\infty(\Omega)} h^2}{\mu^2} \mu \right) |u|^2_{H^r(\Omega)} \end{aligned}$$

for $r = p + 1$, p being the order of approximation.

Proof. First, we use the fact that the total error $e \doteq u - u_h$ can be split into two parts, the interpolation error $e_{Ih} \doteq u - I_h(u)$ and the approximation error $e_h \doteq I_h(u) - u_h$, i.e., $e = e_{Ih} + e_h$. $I_h(u)$ is an optimal interpolant of u (see Lemma 2.5.6).
Let us denote the right-hand side of (4.4) as $a(u, v)$. Now, we use first the Galerkin orthogonality, to get:

$$\begin{aligned} a(u - u_h, v_h) &= a(I_h(u) - u_h, v_h) + a(u - I_h(u), v_h) \\ &= a(e_h, v_h) + a(e_{Ih}, v_h) = 0, \qquad \forall v_h \in V_h. \end{aligned}$$

We can now take $v_h = e_h$ and the stability result in Theorem 4.2.1, to

get:

$$\frac{\mu}{2}|e_h|^2_{H^1(\Omega)} + \sigma\|e_h\|^2_{L^2(\Omega)} \leq a(e_h, e_h) = -a(e_{Ih}, e_h).$$

Next, we must bound the right-hand side of the previous equation. We have:

$$-a(e_{Ih}, e_h) = \int_\Omega \mu \boldsymbol{\nabla} e_{Ih} \cdot \boldsymbol{\nabla} e_h \mathrm{d}\Omega + \int_\Omega \sigma e_{Ih} e_h \mathrm{d}\Omega + \int_\Omega e_h((\boldsymbol{\beta} \cdot \boldsymbol{\nabla} e_{Ih}))\mathrm{d}\Omega.$$

We can bound the first two terms as follows:

$$\begin{aligned}&\int_\Omega \mu \boldsymbol{\nabla} e_{Ih} \cdot \boldsymbol{\nabla} e_h \mathrm{d}\Omega + \int_\Omega \sigma e_{Ih} e_h \mathrm{d}\Omega \\ &\leq \mu|e_{Ih}|^2_{H^1(\Omega)} + \sigma|e_{Ih}|^2_{H^1(\Omega)} + \frac{\mu}{4}|e_h|^2_{H^1(\Omega)} + \frac{\sigma}{4}|e_h|^2_{H^1(\Omega)}.\end{aligned}$$

The convective term can be treated as follows:

$$\begin{aligned}\int_\Omega e_h(\boldsymbol{\beta} \cdot \boldsymbol{\nabla} e_{Ih})\mathrm{d}\Omega &= -\int_\Omega e_{Ih}(\boldsymbol{\beta} \cdot \boldsymbol{\nabla} e_h)\mathrm{d}\Omega \\ &\leq \|\boldsymbol{\beta}\|_{L^\infty(\Omega)}\|e_{Ih}\|_{L^2(\Omega)}\|\boldsymbol{\nabla} e_h\| \\ &\leq \frac{\|\boldsymbol{\beta}\|^2_{L^\infty(\Omega)}}{\mu}\|e_{Ih}\|^2_{L^2(\Omega)} + \frac{\mu}{4}\|\boldsymbol{\nabla} e_h\|^2.\end{aligned}$$

Alternatively, if $\sigma > 0$, we can also bound the convective term as follows:

$$\int_\Omega e_h(\boldsymbol{\beta} \cdot \boldsymbol{\nabla} e_{Ih})\mathrm{d}\Omega \leq \frac{\|\boldsymbol{\beta}\|^2_{L^\infty(\Omega)}}{\sigma}\|e_{Ih}\|^2_{H^1(\Omega)} + \frac{\sigma}{4}\|e_h\|^2_{L^2(\Omega)}.$$

Thus, we get

$$\begin{aligned}&\frac{\mu}{4}|e_h|^2_{H^1(\Omega)} + \frac{\sigma}{4}\|e_h\|^2_{L^2(\Omega)} \\ &\leq \mu|e_{Ih}|^2_{H^1(\Omega)} + \sigma|e_{Ih}|^2_{H^1(\Omega)} \\ &\quad + \min\left(\frac{\|\boldsymbol{\beta}\|^2_{L^\infty(\Omega)}}{\sigma}\|e_{Ih}\|^2_{H^1(\Omega)}, \frac{\|\boldsymbol{\beta}\|^2_{L^\infty(\Omega)}}{\mu}\|e_{Ih}\|^2_{L^2(\Omega)}\right)\end{aligned}$$

Using the interpolation properties of FE spaces, assuming that the order of approximation is $p > 0$ and that the exact solution $u \in H^r(\Omega)$ for $1 < r < p$, we get

$$|e_{Ih}|_{H^1(\Omega)} \leq Ch^r |u|_{H^{r+1}(\Omega)}, \qquad |e_{Ih}|_{L^2(\Omega)} \leq Ch^{r+1} |u|_{H^{r+1}(\Omega)}.$$

Using these interpolation error bounds, we finally get:

$$\begin{aligned}
&\frac{\mu}{2}|e_h|^2_{H^1(\Omega)} + \|\hat{\sigma} e_h\|^2_{L^2(\Omega)} \\
&\leq \frac{\mu}{2} h^{2(r-1)} |u|^2_{H^r(\Omega)} + \frac{\sigma}{2} h^{2r} |u|^2_{H^1(\Omega)} \\
&+ \min\left(\frac{\|\boldsymbol{\beta}\|^2_{L^\infty(\Omega)}}{\sigma^2}\sigma, \frac{\|\boldsymbol{\beta}\|^2_{L^\infty(\Omega)} h^2}{\mu^2}\mu \right) h^{2(r-1)} \|u\|^2_{H^r(\Omega)}.
\end{aligned}$$

We can prove the final result using the triangle inequality and the interpolation error. □

The last bound shows how the convective term affects the accuracy of the FE method. Let us consider the case $\sigma = 0$ for simplicity. In this case, we get

$$\frac{\mu}{2}|e|_{H^1(\Omega)} + \sigma \|e\|_{L^2(\Omega)} \leq C(1 + \frac{\|\boldsymbol{\beta}\|_{L^\infty(\Omega)} h}{\mu}) \frac{\mu}{2} h^{(r-1)} |u|_{H^r(\Omega)}. \tag{4.5}$$

for a constant $C > 0$. The constant in the error blows up as $\|\boldsymbol{\beta}\|_{L^\infty(\Omega)} \to \infty$. As a result, as convection becomes more dominant for a fixed mesh, we should expect a loss of accuracy in the FE method. However, the problem can be *solved* by refining the mesh, recovering good results for $h \sim \frac{\mu}{\|\boldsymbol{\beta}\|_{L^\infty(\Omega)}}$. Unfortunately, in many cases, this mesh size is not practical (especially in 3D) due to limited computational resources. In the following sections, we will present two methods that aim to solve this issue by adding new *stabilisation* terms to the discrete formulation.

We note that the discussion is very similar when approximation fluid problems using the Navier-Stokes equations. The limit of high convection represents a highly turbulent, very chaotic flow. The approximation using FE methods is not accurate unless the mesh size can capture *viscous* scales. The Navier-Stokes equation is nonlinear and indefinite, so we will not consider it in this introductory course.

4.4 Artificial diffusion

The artificial diffusion method was the first to tackle the problem described above. The method dates back to 1950, proposed by Von Neumann and Richtmyer, who worked at Los Alamos National Labs for the Manhattan project. The idea is simple. Looking at (4.5), it is clear that if $\mu \sim \|\boldsymbol{\beta}\|_{L^\infty(\Omega)} h$, the problem is solved, i.e., the error for the highly convective regime will be similar to the one at the viscous regime. Thus, they proposed to add an artificial diffusion $\mu_{AD} = \|\boldsymbol{\beta}\|_{L^\infty(\Omega)} h$ to the problem. Let us define $\bar{\mu} \doteq \mu + \mu_{AD}$. The discretisation of the problem with artificial diffusion reads:

$$u_h \in V_{h,g} : \int_\Omega \bar{\mu} \boldsymbol{\nabla} u_h \cdot \boldsymbol{\nabla} v_h \mathrm{d}\Omega + \int_\Omega v_h (\sigma u_h + \boldsymbol{\beta} \cdot \boldsymbol{\nabla}) u_h) \mathrm{d}\Omega = \int_\Omega f v_h \mathrm{d}\Omega. \quad (4.6)$$

Thus, this method is not a Galerkin method anymore since we have added a new term to the Galerkin formulation to fix the accuracy issues for convection-dominant problems.

The stability of the method can be obtained as above, the only difference being the value of the diffusion. However, the addition of the artificial diffusion has an effect on the order of convergence of the method. Let us denote with a_h the bilinear form with the artificial difussion, i.e., the right-hand side in (4.6). Subtracting the continuous equation in (4.3) and the artificial diffusion method in (4.6), we observe that

$$a(u - u_h, v_h) = \int_\Omega \mu_{AD} \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v_h \mathrm{d}\Omega, \qquad \forall v_h \in V_h.$$

Thus, after the perturbation, Galerkin orthogonality does not hold anymore. The right-hand side of this equation is the so-called *consistency* error. Using the definition of the errors above, we get:

$$a(e_h, e_h) = a(e_{Ih}, e_h) + \int_\Omega \mu_{AD} \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} e_h \mathrm{d}\Omega.$$

The interpolation error can be treated as above. On the other side, the consistency error can be bounded as:

$$\int_\Omega \mu_{AD} \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} e_h \mathrm{d}\Omega \leq \mu_{AD} \|\boldsymbol{\nabla} e_h\| \|\boldsymbol{\nabla} u\| = \|\boldsymbol{\beta}\|_{L^\infty(\Omega)} h \|\boldsymbol{\nabla} e_h\| \|\boldsymbol{\nabla} u\|.$$

The stability of the FE solution, the continuous solution, and its interpolant show that the right-hand side is of order $\mathcal{O}(h)$. Thus, the method is at most first order, destroying the accuracy of the FE method for higher order interpolations.

4.5 Streamline diffusion

A slightly less dissipative method can be defined by adding the following *streamline* diffusion to the discrete Galerkin formulation in (4.4):

$$\tau \int_\Omega (\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})v_h \mathrm{d}\Omega, \qquad \text{with } \tau \doteq \frac{h}{\|\boldsymbol{\beta}\|_{L^\infty(\Omega)}} \tag{4.7}$$

being a numerical coefficient whose choice will become clear when carrying out the error analysis of the method. However, it can already be seen that this choice leads to a dimensionally correct term, i.e., all the terms in the equation (e.g., compare against the convective term) have the same dimension. Dimension consistency is a necessary condition for a good numerical stabilisation. In order to check the dimensional equivalence, we note that, due to the discrete inequality, $\|\boldsymbol{\nabla} u_h\| \leq h^{-1}\|u_h\|$ and so, we can understand $\boldsymbol{\nabla}$ over FE functions as h^{-1}. It can be shown that this term also solves the problem with the Galerkin formulation, but it is still a first-order method.

We can now see how effective this method is. The stability analysis follows the same line as above. Taking $v_h = u_h$, we have the additional stability

$$\tau\|\boldsymbol{\beta}\cdot\boldsymbol{\nabla} u_h\|^2.$$

Let us proceed to the error analysis. We only provide the boudns that are different from the Galerkin analysis. We can now handle the error related to the convective term as follows, taking benefit of the additional stability:

$$\begin{aligned}\int_\Omega e_h(\boldsymbol{\beta}\cdot\boldsymbol{\nabla} e_{Ih})\mathrm{d}\Omega = -\int_\Omega e_{Ih}(\boldsymbol{\beta}\cdot\boldsymbol{\nabla} e_h)\mathrm{d}\Omega &\leq \|e_{Ih}\|_{L^2(\Omega)}\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})e_h\| \\ &\leq \tau^{-1}\|e_{Ih}\|^2_{L^2(\Omega)} + \frac{\tau}{4}\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})e_h\|^2.\end{aligned}$$

$$\tau\int_\Omega (\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})v_h\mathrm{d}\Omega \leq \tau\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})e_{Ih}\|^2 + \frac{\tau}{4}\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})e_h\|^2.$$

Now, we can check that the two new interpolation error terms can be bounded as follows:

$$\begin{aligned}\tau^{-1}\|e_{Ih}\|^2_{L^2(\Omega)} &\leq \|\boldsymbol{\beta}\|_{L^\infty(\Omega)}h^{2(r-1/2)}|u|^2_{H^r(\Omega)}, \\ \tau\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})e_{Ih}\|^2 &\leq \|\boldsymbol{\beta}\|_{L^\infty(\Omega)}h^{2(r-1/2)}|u|^2_{H^r(\Omega)}.\end{aligned}$$

However, the consistency error for this problem can only be bounded as:

$$\tau \int_\Omega (\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})e_h \mathrm{d}\Omega \leq h\|\boldsymbol{\nabla}u\|\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})e_h\|.$$

Again, the method can only be first order, regardless of the order of the FE approximation.

4.6 SUPG stabilisation

In this section, we will finally see how to fix the Galerkin method in a consistent way. We will define a method with no consistency error, i.e., the exact solution is also the solution of the discrete method (even though not Galerkin). The proposed method will be optimally convergent in terms of h. In this section, we consider $\sigma = 0$ for simplicity.

The question is how to add the *magic* streamline diffusion term in (4.7) without introducing a consistency error. The idea can be to add a term that involves the residual of the PDE in strong form. However, second derivatives of FE functions are not well-posed (these functions are only $\mathcal{C}^0$). Instead, we will add the residual at the interior of each cell in a *broken* cell-wise way. This is the idea behind the Streamline-upwinding Petrov-Galerkin (SUPG) method, which adds the following term:

$$\sum_{K\in\mathcal{T}_h} \tau_K \int_K (-\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}u_h) + (\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h)((\boldsymbol{\beta}\cdot\boldsymbol{\nabla})v_h)\mathrm{d}\Omega. \tag{4.8}$$

Two key observations apply. First, the term (4.7) that provides streamline diffusion is part of (4.8). Second, this term vanishes for a smooth enough solution $u \in H^2(\Omega)$. Thus, it introduces no consistency error.

Now, let us analyse the stability of this method. We have, for $v_h = u_h$:

$$\sum_{K\in\mathcal{T}_h} \tau_K \int_K (-\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}u_h) + (\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h - f)((\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h)\mathrm{d}\Omega$$

$$= \sum_{K\in\mathcal{T}_h} \tau_K \|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h\|^2_{L^2(K)} - \sum_{K\in\mathcal{T}_h} \tau_K \int_K (\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}u_h) + f)((\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h)\mathrm{d}\Omega.$$

Let us bound the last term in the expression above. On one side, we can bound the forcing term

$$\sum_{K\in\mathcal{T}_h} \tau_K \int_K f((\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h) \leq \sum_{K\in\mathcal{T}_h} 4\tau_K \|f\|^2_{L^2(K)} + \sum_{K\in\mathcal{T}_h} \frac{\tau_K}{4}\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h\|^2_{L^2(K)}.$$

Next, we bound the stabilisation term related to the Laplacian. We recall that we assume that μ is constant for simplicity. The viscous term in the SUPG stabilisation reads:

$$-\sum_{K\in\mathcal{T}_h}\tau_K\int_K(\mu\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}u_h)((\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h)\mathrm{d}\Omega.$$

We note that this term vanishes for first order FEs on simplicial meshes (triangles or tetrahedra), i.e., when the local FE spaces are $\mathcal{P}_1(K)$ or $\mathcal{P}_2(K)$. For hex meshes (quadrilateral or hexahedra), this term only vanishes for first order spaces, i.e., for the local FE space $\mathcal{Q}_1(K)$. In any other situation, the bound for this term is more involved. Let us consider the following discrete inverse inequality for a given FE space:

$$\|\Delta u_h\|_K \leq C_{inv}h_T^{-1}\|\boldsymbol{\nabla}u_h\|_K,$$

where the constant C_{inv} depends on the order p of the FE space (it can be proved that the constant is of order $\mathcal{O}(p^2)$). Using this inequality, we get

$$\begin{aligned}
&-\sum_{K\in\mathcal{T}_h}\tau_K\int_K(\mu\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}u_h)((\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h)\mathrm{d}\Omega\\
&\leq\sum_{K\in\mathcal{T}_h}\tau_K^{1/2}h^{-1}\mu C_{inv}\|\boldsymbol{\nabla}u\|_{L^2(K)}\tau_K^{1/2}\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h\|_{L^2(\Omega)}\\
&\leq\sum_{K\in\mathcal{T}_h}\tau_K h^{-2}\mu^2C_{inv}^2\|\boldsymbol{\nabla}u\|_{L^2(K)}^2+\frac{\tau_K}{4}\|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h\|_{L^2(\Omega)}^2.
\end{aligned}$$

Under the assumption that

$$\tau_K<\frac{h^2}{4\mu C_{inv}^2},$$

we can prove the stability of the method. For example, a typical choice of τ that satisfies the requirements is the following:

$$\tau_K\doteq\left(\frac{4\mu}{C_{inv}^2h^2}+\frac{\|\boldsymbol{\beta}\|_{L^\infty(K)}}{h}\right)^{-1}.$$

It provides the desired dissipation in the limit of zero diffusion and satisfies the upper bound required for stability. Combining these results, we can obtain the stability of the SUPG method.

Lemma 4.6.1: Stability of SUPG

The solution u_h of the SUPG stabilisation of (4.4) satisfies the following stability bound:

$$\frac{\mu}{2}|u|^2_{H^1(\Omega)} + \sum_{K\in\mathcal{T}_h} \tau_K \|(\boldsymbol{\beta}\cdot\boldsymbol{\nabla})u_h\|^2 \leq \frac{C_P^2}{2\mu}\|f\|^2_{H^{-1}(\Omega)} + \sum_{K\in\mathcal{T}_h} 4\tau_K\|f\|^2_{L^2(K)},$$

Under the assumption that $\tau_K < \frac{h^2}{4\mu C_{inv}^2}$.

We note that, in general, the value of C_{inv} depends on not only the local FE space but also the cell shape. It can be computed locally by solving an eigenvalue problem. However, this is quite expensive. For structured meshes (all cells have the same shape), these constants can be computed beforehand though.

The optimal convergence analysis of this method is left as an exercise.

4.7 Tutorials

1. Consider the 1D one-dimensional problem

   $$-\mu u'' + \beta u' = 0, \quad 0 < x < 1, \quad u(0) = 1, u(1) = 0.$$

   The analytical solution of this problem reads

   $$u(x) = (1 - e^{-\frac{1}{\mu}})^{-1}(1 - e^{-\frac{1-x}{\mu}}).$$

   Write the linear equation corresponding to interior nodes in the mesh (the end-point values are known) for a linear FE approximation of this problem. Here we are asking for the expression of the rows in the linear system arising from the discretisation. Since all the rows are identical, we only ask you to provide one. E.g., for the node i in the mesh, the equation must be in terms of the nodal values U^{i-1}, U^i, U^{i+1}.

2. Consider the nodal equation above and take the limit when $\mu \to 0$. Assume that N is odd. In the limit, which solution do you get?

3. In the problem above, take $\beta = 1$ and $\mu = 1/10$. First, choose a value of h for which you would expect reasonably good behavior of the Galerkin method. Justify your choice.

4. Define a (consistent) SUPG method for the problem with reaction. Perform the stability analysis. Which additional requirements do you need on τ_K to get stability? Can you define an expression of τ that satisfies this other requirement and the previous ones (for absorbing the diffusive term and for having the right amount of added diffusion in the convection-dominated case)?

5. Next, use the code provided in the lectures for this problem (using Gridap) with the values in the previous question. Observe what happens when $\mu = 10^{-k}$, $k = 1, 2, 3, 4$. Compute the error $\|u - u_h\|_{H^1(\Omega)}$ with respect to μ. Explain what you observe. (Provide figures or tables with the error and plots of obtained solutions).

6. Take $\mu = 10^{-3}$. Observe what happens when reducing $h = 10^{-l}$, $l = 1, 2, 3$. Compute the error $\|u - u_h\|_{H^1(\Omega)}$ with respect to h. Justify what you observe. (Provide figures or tables).

7. Repeat the last two questions for the streamline diffusion and SUPG method.

8. Now, consider the previous question for quadratic elements instead of linear. What do you observe? Can you explain the results?

Chapter 5

Parabolic equations

5.1 Introduction

In this section, we analyse time-dependent problems. We will work with the heat equation as a model problem, even though we can readily use the techniques proposed herein for other parabolic problems, e.g., the transient convection-diffusion equation.

We will introduce the method of lines for the discretisation of parabolic PDEs. The idea is to carry out the spatial discretisation using FEs but considering the DOFs values as time-dependent. After this step, the problem will be discrete in space but continuous in time. More specifically, we end up with a system of ordinary differential equations (ODEs) for the DOF values of the FE approximation. Next, we will use an ODE solver for this problem.

This unit does not focus on providing a detailed presentation of numerical methods for ODEs, which is the main ingredient we will need in this chapter. Instead, we rely on the material from unit MTH2051. We refer the interested student to the lecture notes of this unit for more details. We will focus on the θ-method for simplicity, which involves both explicit and implicit methods and first and second-order (in time) schemes.

We will do the stability and convergence analysis of the semi-discrete problem in space. We will also discuss the stability of the fully discrete problem. The convergence analysis of the fully discrete problem is quite technical and not considered.

5.2 The heat equation

Let us consider the heat equation, i.e., the transient version of the Poisson problem. We consider a physical domain in space Ω and the time interval $I \doteq (0, T]$. The strong form of this problem reads:

$$\begin{aligned} u_t - \boldsymbol{\nabla} \cdot \mu \boldsymbol{\nabla} u &= f, && \text{in } \Omega \times I \\ u &= g, && \text{on } \Gamma_D \times I \\ \mu \boldsymbol{n} \cdot \boldsymbol{\nabla} u &= q, && \text{on } \Gamma_N \times I \\ u(\boldsymbol{x}, 0) &= u^0(\boldsymbol{x}), && \boldsymbol{x} \in \Omega, \end{aligned}$$

where we have considered Dirichlet and Neumann boundary conditions and a *initial condition* u^0 at $t = 0$. The imposition of an initial condition is essential for the problem to be well-posed. In the following, we consider $g = 0$, $\Gamma_D = \partial\Omega$ and $\Gamma_N = \emptyset$ (i.e., homogeneous Dirichlet conditions) for simplicity. In any case, the extension can be carried out using the same techniques as in previous sections.

Let us make some comments about the nature of this system. We can observe that the time derivative term is a *convective* term. If we consider the $d+1$ space-time domain $\Omega \times I$, $u_t = \boldsymbol{\beta} \cdot \boldsymbol{\nabla}_{(\boldsymbol{x},t)} u$ for $\boldsymbol{\beta} = (\boldsymbol{0}, 1)^T$. On the other hand, we only have diffusion in the spatial direction. This observation tells us that a standard Galerkin FE approximation of this problem in space-time is not going to work.

Another critical observation is time *causality*, the solution at a given time depends on past time values but not future time values. The *arrow of time* should be preserved at the discrete level since it is very beneficial from a computational point of view. Thus, standard FE approximations of the space-time problem (possibly with stabilisation in the time direction convection) are not a clever choice. They couple all the time values in all directions.

The methods proposed below preserve time causality and can be stable. We consider first space discretisation and move next to time discretisation.

5.3 Space discretisation using FEs

To discretise the problem in space, we start writing it in weak form at any time value $t \in I$:

$$u(t) \in H_0^1(\Omega)) \ : \ \int_\Omega u_t v \mathrm{d}\Omega + \int_\Omega \mu \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \mathrm{d}\Omega = \int_\Omega f v \mathrm{d}\Omega, \tag{5.1}$$

for any $v \in H_0^1(\Omega)$.

Let us introduce some notation. The space of functions $L^2(I;V)$ is the one that contains functions such that

$$\|u\|^2_{L^2(I;V)} \doteq \int_0^T \|u\|^2_V \mathrm{d}\Omega \leq \infty.$$

E.g., V can be $L^2(\Omega)$ or $H^1(\Omega)$. We also define the norms

$$\|u\|^2_{\mathcal{C}^j(I;V)} \doteq \sup_{t\in I} \sum_{i=0}^{j} \|d_t^j u\|^2_V \leq \infty.$$

This norm requires the function (and its j derivatives) to be continuous in time.

With this notation, we can state some properties of (5.1). The solution of this problem $u \in \mathcal{C}^0(I; L^2(\Omega)) \cup L^2(I; H^1(\Omega))$. We will show this stability at the discrete level below.

We now consider the method of lines, in which we consider a spatial FE space V_h (the grad-conforming FE space introduced in previous sections), but consider the DOFs to be time-dependent: $u_h(\boldsymbol{x}, t) \doteq \sum_{a=1}^N \varphi^a(\boldsymbol{x}) U^a(t)$, where $V_h \doteq \mathrm{span}\{\varphi_1, \ldots, \varphi_N\}$ and $U(t) \in \mathbb{R}^N$ is the vector of DOF values. Using the Galerkin method at every time value $t \in I$, we obtain:

$$u_h(t) \in V_h \ : \ \int_\Omega (u_h)_t v_h \mathrm{d}\Omega + \int_\Omega \mu \boldsymbol{\nabla} u_h \cdot \boldsymbol{\nabla} v_h \mathrm{d}\Omega = \int_\Omega f v_h \mathrm{d}\Omega, \tag{5.2}$$

for any $v_h \in V_h$. This is a semi-discrete problem in time, since we have performed the discretisation in space via FE method but not in time. We can now proceed as for steady problems and end up with a system of ODEs:

$$\left[\int_\Omega \varphi^a(\boldsymbol{x}) \varphi^b(\boldsymbol{x}) \mathrm{d}\Omega\right] U_t^b(t) + \left[\int_\Omega \boldsymbol{\nabla}\varphi^a(\boldsymbol{x}) \cdot \boldsymbol{\nabla}\varphi^b(\boldsymbol{x}) \mathrm{d}\Omega\right] U^b(t) = \int_\Omega f \varphi^a \mathrm{d}\Omega,$$

for any $a = 1, \dots, N$. Using compact matrix notation, we have the ODE system:

$$MU_t + KU = F, \qquad \text{or} \qquad U_t + M^{-1}KU = M^{-1}f,$$

where we have used the fact that the mass matrix is non-singular. The analysis of this ODE system has been studied in MTH2051. We make use of Picard-Lindelöf theorem to prove the existence and uniqueness of a vector $U(T)$ with $\mathcal{C}^0(I)$ entries.

To prove error estimates, we will make use of the Gronwall inequality.

Lemma 5.3.1: Gronwall inequality

Let $\phi \in \mathcal{C}^1(I)$ and $h \in \mathcal{C}^0(I)$ such that $\phi_t \leq \beta\phi + h$, for $\beta \in \mathbb{R}$. Then, it holds

$$\phi(t) \leq e^{\beta t}\phi(0) + \int_0^t e^{\beta(t-s)} f(s)\mathrm{d}s, \qquad \forall s \in I.$$

Proof. Multiply the inequality by $e^{-\beta t}$ and integrate with respect to t. □

Theorem 5.3.2: Stability of semi-discrete heat equation

The solution u_h of (5.2) satisfies the stability bound for any $s \in I$:

$$\|u_h(s)\|^2_{L^2(\Omega)} + \int_0^s \mu \|\boldsymbol{\nabla} u_h\|^2_{L^2(\Omega)}\mathrm{d}s \leq \int_0^s \frac{1}{\mu}\|f(s)\|^2_{H^{-1}(\Omega)}\mathrm{d}s.$$

Thus, $u_h \in \mathcal{C}^0(I; L^2(\Omega)) \cap L^2(I; H^1(\Omega))$.

Proof. Let us take $u_h = v_h$ in (5.1). Treating the diffusive term and forcing term as for the steady case (but integrated over time), we obtain:

$$\int_\Omega u_h (u_h)_t \mathrm{d}\Omega + \frac{\mu}{2}\|\boldsymbol{\nabla} u_h\|^2_{L^2(\Omega)}\mathrm{d}\Omega \leq \frac{1}{2\mu}\|f\|^2_{H^{-1}(\Omega)}.$$

For the time derivative term, we use:

$$\int_\Omega u_h (u_h)_t \mathrm{d}\Omega = \frac{1}{2}\int_\Omega (u_h^2)_t \mathrm{d}\Omega$$

Combining these two equations, and integrating from $t = 0$ to $t = s \in I$, we get:

$$\|u_h(s)\|^2_{L^2(\Omega)} + \int_0^s \mu \|\boldsymbol{\nabla} u_h\|^2_{L^2(\Omega)} \mathrm{d}s \leq \int_0^s \|f(s)\|^2_{H^{-1}(\Omega)} \mathrm{d}s.$$

□

Now, let us prove error estimates in space for the semi-discrete problem. The proof is quite long but accessible. It relies on a specific projector that allows us to eliminate the interpolation error related to the diffusive term to obtain optimal error bounds in the $\mathcal{C}^0(I; L^2(\Omega))$ norm. We make use of the so-called Riestz projector. Let us consider at each time $s \in I$ the solution of the following problem:

$$\mathcal{R}_h(u) \in V_h \ : \ \int_\Omega (\boldsymbol{\nabla} u - \boldsymbol{\nabla}\mathcal{R}_h(u)) \cdot \boldsymbol{\nabla} v_h \mathrm{d}\Omega = 0, \qquad \forall v_h \in V_h. \tag{5.3}$$

This is just a Laplacian problem, for which we can readily prove optimal error estimates, i.e.,

$$\|u - \mathcal{R}_h u\|_{L^2(\Omega)} + h\|u - \mathcal{R}_h u\|_{H^1(\Omega)} \leq C h^{r+1} |u|_{H^r(\Omega)}. \tag{5.4}$$

Using this projector we can eliminate the interpolation error for the viscous term (that can only be bounded by h^r) and obtain optimal error estimates in $\mathcal{C}^0(I; L^2(\Omega))$ that decrease with h^{r+1} (see proof below). Optimal error estimates of order h^r in the norm $L^2(I; H^1(\Omega))$ could be obtained without this projector, in a more standard way. In the proof below, we use $e_h \doteq \mathcal{R}_h u - u_h$.

Theorem 5.3.3: Error estimates for the semi-discrete heat equation

Let us assume that the solution u of (5.1) is $u \in \mathcal{C}^1(I; H^r(\Omega))$ and the solution u_h of the semi-discrete counterpart (5.2) for a FE space of

order r. They satisfy the following error bound:

$$\begin{aligned}\|u(t)-u_h(t)\|_{\mathcal{C}^0(I;L^2(\Omega))} \leq & h^{r+1}|u(0)|_{H^{r+1}(\Omega)}e^{-\mu C_P\frac{T}{2}} \\ & + \frac{1}{\mu C_P}\left(h^{r+1}|u\|_{H^{r+1}(\Omega)} + \frac{h^{r+1}}{\mu C_P}|u_t\|_{H^{r+1}(\Omega)}\right), \\ \mu^{\frac{1}{2}}\|u(t)-u_h(t)\|_{L^2(I;H^1(\Omega))} \leq & h^{r+1}|u(0)|_{H^{r+1}(\Omega)} \\ & + \frac{\sqrt{T}}{\sqrt{\mu}}h^{r+1}|u_t|_{H^{r+1}(\Omega)} + \mu^{\frac{1}{2}}h^r|u|_{H^{r+1}(\Omega)}.\end{aligned}$$

Proof. We can use Galerkin orthogonality and the definition of $\mathcal{R}_h$ in (5.3) to get:

$$\begin{aligned}&((\mathcal{R}_h u-u_h)_t, v_h) + (\mu\boldsymbol{\nabla}(\mathcal{R}_h u-u_h), \boldsymbol{\nabla}v_h) \\ &= ((\mathcal{R}_h u-u)_t, v_h) + (\mu\boldsymbol{\nabla}(\mathcal{R}_h u-u), \boldsymbol{\nabla}v_h) \\ &= ((\mathcal{R}_h u-u)_t, v_h), \qquad \forall v_h \in V_h.\end{aligned}$$

Now, we can take $v_h = \mathcal{R}_h u - u_h$ to get, using the stability in Theorem 5.3.3:

$$\begin{aligned}&d_t\|\mathcal{R}_h u-u_h\|^2_{L^2(\Omega)} + \mu\|\boldsymbol{\nabla}(\mathcal{R}_h u-u_h)\|^2_{L^2(\Omega)} \\ &\leq ((\mathcal{R}_h u-u)_t, \mathcal{R}_h u-u_h).\end{aligned} \tag{5.5}$$

We split the proof into two parts. We obtain $L^2(\Omega)$ errors first, and $H^1(\Omega)$ errors next.
(1.) $L^2(\Omega)$ error: We can treat the right-hand side as follows:

$$\begin{aligned}&((\mathcal{R}_h u-u)_t, \mathcal{R}_h u-u_h) \\ &\leq \frac{1}{2\mu C_P}\|(\mathcal{R}_h u-u)_t\|^2_{L^2(\Omega)} + \frac{\mu C_P}{2}\|\mathcal{R}_h u-u_h\|^2_{L^2(\Omega)},\end{aligned}$$

Using the Poincaré inequality on the left-hand side of (5.5), we get:

$$2d_t\|\mathcal{R}_h u-u_h\|^2_{L^2(\Omega)} + C_P\mu\|\mathcal{R}_h u-u_h\|^2_{L^2(\Omega)} \leq \frac{1}{\mu C_P}\|(\mathcal{R}_h u-u)_t\|^2_{L^2(\Omega)}.$$

Gronwall's inequality leads to (for $e_h \doteq \mathcal{R}_h u - u_h$):

$$\begin{aligned}\|e_h(t)\|^2_{L^2(\Omega)} \leq & \|e_h(0)\|^2_{L^2(\Omega)} e^{-\mu C_P t} \\ & + \frac{1}{\mu C_P}\int_0^t e^{-\mu C_P(t-s)}\|(\mathcal{R}_h u - u)_t\|^2_{L^2(\Omega)}\mathrm{d}s.\end{aligned}$$

We can treat the last term as:

$$\begin{aligned}&\int_0^t e^{-\mu C_P(t-s)}\|(\mathcal{R}_h u - u_h)_t\|^2_{L^2(\Omega)}\mathrm{d}s \\ &\leq \max_{s\in[0,t]}\|(\mathcal{R}_h u - u)_t\|^2_{L^2(\Omega)}\int_0^t e^{-\mu C_P(t-s)}\mathrm{d}s \\ &\leq \frac{1}{\mu C_P}(1-e^{-\mu C_P t})\max_{s\in[0,t]}\|(\mathcal{R}_h u - u)_t\|^2_{L^2(\Omega)}.\end{aligned}$$

Using the triangle inequality, we finally get:

$$\begin{aligned}&\|u(t)-u_h(t)\|_{L^2(\Omega)} \leq \|e_h(0)\|_{L^2(\Omega)} e^{-\mu C_P \frac{t}{2}} \\ &+ \frac{1}{\mu C_P}\max_{s\in[0,t]}\left(\|(\mathcal{R}_h u - u)_t\|_{L^2(\Omega)}\right) + \|\mathcal{R}_h u(t) - u(t)\|_{L^2(\Omega)}.\end{aligned}$$

Using now approximability results for the initial error and the Riestz projector in (5.4), we obtain the desired bound

$$\begin{aligned}\|u(t)-u_h(t)\|_{\mathcal{C}^0(I;L^2(\Omega))} \leq & h^{r+1}|u(0)|_{H^{r+1}(\Omega)} e^{-\mu C_P \frac{T}{2}} \\ & + \left(h^{r+1}\|u\|_{H^{r+1}(\Omega)} + \frac{h^{r+1}}{\mu C_P}\|u_t\|_{H^{r+1}(\Omega)}\right).\end{aligned}$$

(2.) $H^1(\Omega)$ error: We can also obtain H^1 bounds for the error as follows. We start again from (5.5), but now treat the right-hand side as follows:

$$\begin{aligned}&((\mathcal{R}_h u - u)_t, \mathcal{R}_h u - u_h) \\ &\leq \frac{1}{2\mu\alpha}\|(\mathcal{R}_h u - u)_t\|^2_{H^{-1}(\Omega)} + \frac{\mu\alpha}{2}\|\mathcal{R}_h u - u_h\|^2_{H^1(\Omega)}\end{aligned}$$

for an arbitrary $\alpha > 0$. Combining this result with (5.5), using Poincaré and picking α small enough, we get:

$$\|\mathcal{R}_h u - u_h\|^2_{L^2(\Omega)} + \mu|\mathcal{R}_h u - u_h|^2_{H^1(\Omega)} \leq \frac{C}{\mu}\|(\mathcal{R}_h u - u)_t\|^2_{H^{-1}}.$$

Integrating in time this inequality, we get:

$$\mu \int_0^t \|\boldsymbol{\nabla} e_h\|^2_{L^2(\Omega)} \mathrm{d}s \leq \frac{1}{\mu} \int_0^t \|(\mathcal{R}_h u - u)_t\|^2_{H^{-1}(\Omega)} \mathrm{d}s + \|e_h(0)\|^2_{L^2(\Omega)}.$$

As above, we can use error bounds for the Riestz projector (5.4), to get:

$$\begin{aligned}\mu^{\frac{1}{2}} \|u(t) - u_h(t)\|_{L^2(I;H^1(\Omega))} \leq & h^{r+1} |u(0)|_{H^{r+1}(\Omega)} \\ & + \frac{\sqrt{T}}{\sqrt{\mu}} h^{r+1} |u_t|_{H^{r+1}(\Omega)} + \mu^{\frac{1}{2}} h^r |u|_{H^{r+1}(\Omega)}.\end{aligned}$$

□

5.4 Time discretisation

Now, we must discretise (5.2) in time. For that, we can use an ODE solver, e.g., θ-method, Runge-Kutta, etc. If you have done or are doing MTH2051, you have already seen these methods before. We will focus on the θ-method.

In the θ-method, we use a finite difference method. In general, the θ-method splits the time interval $I \doteq [0, T]$ using a partition $0 \doteq t_0 < t_1 < \ldots < t_{N-1} < T_M \doteq T$. We denote the time step size $\tau_i \doteq t_{i+1} - t_i$. For simplicity, we can assume that the time step size is the same at all steps. At each segment $[t_{i+1}, t_i]$, $i = 0, \ldots, M-1$, we consider a linear variation of the unknown $U(t)$, i.e.,

$$U_t = \frac{U^{i+1} - U^i}{\tau_i}, \quad t \in [t_i, t_{i+1}], \quad U(t^i + \theta\tau_i) = \theta U^{i+1} + (1-\theta) U^i,$$

and solve the ODE at the value $t_i + \theta\tau_i$:

$$M \frac{U^{i+1} - U^i}{\tau_i} + K U^{i+\theta} = F^{i+\theta},$$

where we assume that U^i is known. We note that we can make use of *causality* in time. At each time step, we know U^i from the previous time step (or initial condition) and can compute for the next value U^{i+1}. That is the reason why this kind of methods are usually called *time marching* methods. This fact

has very important consequences in the reduction of the computational cost of transient problems.

The most common values for θ are the following. The case $\theta = 1$ is the so-called backward Euler method (BE). In this case, we get the following discrete problem at each time value:

$$(M + \tau_i K)U^{i+1} = \tau_i F^{i+1} + MU^i. \tag{5.6}$$

Using the fact that the matrix M and K are positive-definite, we will readily check the stability of this method (see below).

The case $\theta = 0$ is the so-called forward Euler method (FE). In this case, we get the following discrete problem at each time value:

$$MU^{i+1} = \tau_i F^{i+1} + (M - \tau_i K)U^i.$$

This method is explicit since all terms but the time derivative term is treated using the expression from the previous time step. Thus, the problem to be solved at each time step involves a mass matrix, which is very easy to invert (spectrally equivalent to an identity matrix). However, the *big* price to pay is a loss of unconditional stability. The stability of explicit methods relies on a so-called CFL (for Courant-Friedrichs-Lewy) condition. I.e., the time step size must be small *enough* for stability to hold.

The two methods above are at most first order in time. The error only decreases linearly with τ. A quadratic method is obtained by picking $\theta = \frac{1}{2}$, which is known as the mid-point rule or Crank-Nicholson (CN) scheme. Doing some algebraic manipulations, we finally get:

$$(M + \frac{1}{2}\tau_i K)U^{i+1} = \tau_i F^{i+\frac{1}{2}} + (M - \frac{1}{2}\tau_i K)U^i.$$

Again, this method is unconditionally stable but second order in time.

Let us now prove the stability of these methods. We use the following matrix-norm notation, $\|U\|_X^2 = U^T X U$.

Lemma 5.4.1: Stability of BE-FE method

The solution of the BE discretisation in time of the heat equation in

(5.6) satisfies the following bound for $m = 1, \dots, M$:

$$\|U^m\|_M^2 + \sum_{i=1}^{m} \|U^{i+1} - U^i\|_M^2 + \sum_{i=1}^{m} \tau \|U^i\|_K^2 \leq \|F\|_{K^{-1}}^2 + \|U^0\|_M^2.$$

Proof. In order to prove stability of the Backward-Euler method, we use the following trick:

$$(a+b)a = \frac{1}{2}(a^2 - b^2 + (a+b)^2).$$

Now, we pre-multiply the equation by U^{i+1}, use the formula above and add up to time value m to get the desired result. □

We note that the first and third terms on the right-hand side of the stability bound in the previous lemma are the L^2 and H^1 stability terms we obtained at the semi-discrete level. However, after BE integration, we have a new stability term from the discrete integration. This term comes from the *numerical diffusion* introduced by this method. The method adds artificial diffusion to the problem at hand, which can negatively affect the energy conservation of the scheme (it is unphysically dissipating energy).

Now, let us look at the CN scheme stability.

Lemma 5.4.2: Stability of CN-FE method

The solution of the CN discretisation in time of the heat equation in (5.6) satisfies the following bound for $m = 1, \dots, M$:

$$\|U^m\|_M^2 + \sum_{i=1}^{m} \tau \|U^i\|_K^2 \leq \|F\|_{K^{-1}}^2 + \|U^0\|_M^2.$$

Proof. For CN, we have to use the expression:

$$\frac{1}{2}(a-b)(a+b) = a^2 - b^2.$$

In this case, we pre-multiply by $U^{i+\frac{1}{2}} \doteq \frac{1}{2}U^{i+1} + \frac{1}{2}U^i$. Using the

expression above, we readily get the desired result. □

Thus, we can observe that this method introduces no numerical dissipation at all but is stable. Furthermore, one can prove this method is second-order accurate in time.

The last method to analyse is FwE. We note that this method is not unconditionally stable. Instead, the stability of this scheme depends on the size of the time step value.

Lemma 5.4.3: Stability of FwE-FE method

The solution of the FwE discretisation in time of the heat equation in (5.6) satisfies the following bound for $m = 1, \dots, M$:

$$\|U^m\|_M^2 + C\sum_{i=1}^{m}\|U^{i+1} - U^i\|_M^2 + \sum_{i=1}^{m}\tau\|U^i\|_K^2 \leq \|F\|_{K^{-1}}^2 + \|U^0\|_M^2,$$

assuming that the CFL condition $\tau < \frac{\mu h^2}{C_{inv}}$ holds.

Proof. For FwE, we can do the same as for BE for the time derivative. The problem is that instead of $U^{i+1}\tau KU^{i+1}$, we now have $U^{i+1}\tau KU^i$. We can do the following:

$$\begin{aligned} U^{i+1}\tau KU^i &= U^{i+1}\tau KU^{i+1} - U^{i+1}\tau K(U^{i+1} - U^i) \\ &= \tau\|U^{i+1}\|_K^2 - U^{i+1}\tau K(U^{i+1} - U^i) \\ &\geq \frac{\tau}{2}\|U^{i+1}\|_K^2 - \frac{\tau}{2}\|U^{i+1} - U^i\|_K^2. \end{aligned}$$

We can now try to absorb the last term with $\frac{1}{2}\|U^{i+1} - U^i\|_M^2$. We know that

$$\|V\|_M^2 = \|v_h|_{L^2(\Omega)}^2, \qquad \|V\|_K^2 = \mu\|\boldsymbol{\nabla} v_h|_{L^2(\Omega)}^2.$$

Using an inverse inequality, we get:

$$\tau\|V\|_K^2 = \mu\|\boldsymbol{\nabla} v_h\|_{L^2(\Omega)}^2 \leq C_{inv}\frac{\tau}{\mu h^2}\|v_h\|_{L^2(\Omega)}^2 = C_{inv}\frac{\tau}{\mu h^2}\|V\|_M^2.$$

Assuming that $\tau < \frac{\mu h^2}{C_{inv}}$ holds, i.e., the CFL condition, we can readily get the desired result. □

The CFL condition $\tau < \frac{\mu h^2}{C_{inv}}$ implies that the time step size must decrease with h^2 as $h \to 0$, which is not acceptable in general. Second-order space derivatives are rarely treated explicitly due to this quadratic relation with the time step size. Only convective terms (first order) are suitable for explicit treatment since the corresponding CFL is milder, $\tau \sim h$. As a result, a semi-implicit method (that considers zero and first-order terms explicitly and second-order terms implicitly) is an excellent choice in these situations.

5.5 Total error

We know from the numerical analysis of ODEs that the BE method is a first order scheme in time, while the CN scheme is second-order. Combining these results with the semi-discrete error estimates for the heat equation in Theorem 5.3.3, we get the space-time errors below. We do not prove them for the sake of brevity.

Theorem 5.5.1: Error estimates for the fully discrete heat equation

Let us assume that the solution u of (5.1) is $u \in \mathcal{C}^1(I; H^r(\Omega))$ and the solution u_h of the fully-discrete counterpart (5.6) for a FE space of order r. They satisfy the following error bound:

$$\|u(t) - u_h^i\|_{L^2(\Omega)} \leq C(h^{r+1} + \delta t^\beta), \qquad i = 1, \dots, M,$$

$$\left(\sum_{n=1}^{M} \mu \|u(t) - u_h(t)\|^2_{H^1(\Omega)} \right)^{\frac{1}{2}} \leq C(h^r + \partial t^\beta),$$

where $\beta = 1$ for $\theta = 1$ and $\beta = 2$ for $\theta = 0.5$.

5.6 Tutorial

1. Can you write the CN method in terms of $U^{n+\frac{1}{2}}$ and U^n. Compare the cost of BE and CN schemes.

2. Write the discrete problem for the transient convection-diffusion equation using θ-method. Can you provide a CFL condition for the FwE

method applied to this problem? Next, we consider a semi-implicit method in which the diffusive term is evaluated implicitly (at $n+1$, like in BE) and the convective term explicitly (at t^n, like in CN). Can you provide the CFL condition for this scheme?

3. Some of the schemes above can be understood using Petrov-Galerkin schemes and FE spaces, using a weak space-time form. Prove that the Crank-Nicholson method for the heat equation can be recovered using a tensor product space $V_h \times \mathcal{P}_1(I)$ for the trial space and $V_h \times \mathcal{P}_0(I)$ for the tets space.

Chapter 6

Solving the linear system

6.1 Introduction

Any physical problem which involves PDEs (solid mechanics, fluid mechanics, aeroelasticity, etc.) can be discretized (by finite elements, finite volumes or finite differences). After linearisation (for nonlinear problems) e.g., using Newton's method, we end up with a linear system to be solved, as we have seen in the previous chapters for finite element discretisations. Assuming that we have n DOFs in the FE problem, we have to solve a system

$$Au = f, \qquad A \in \mathbb{R}^{n \times n}, \qquad f \in \mathbb{R}^n,$$

and obtain the vector $u \in \mathbb{R}^n$ of DOFs. The finite element solution is the linar combination of the FE shape function basis for these values of u, i.e.,

$$u_h(x) = \sum_{a=0}^{n} u^a \phi^a(x).$$

By construction, A is a square matrix and, assuming that the problem is well-posed, non-singular, as we have already proved. Thus, this problem admits a unique solution. Finally, we note that while u_h is independent of the basis chosen to describe our FE space, the specific expression of the linear system, i.e., A, b and u, do depend on the basis.

This linear system size increases with mesh refinement. Larger linear systems translate to higher computational power requirements.

The main objective of this chapter is to provide a short introduction about how to solve linear systems arising from FE discretizations.

6.2 Preliminaries

6.2.1 Eigenvalues and eigenvectors

Given a square matrix $A \in \mathbb{R}^{n\times n}$, its eigenvalues $\lambda \in \mathbb{C}$ and eigenvectors $u \in \mathbb{R}^n \backslash \{0\}$ are solutions of

$$Au = \lambda u.$$

We make use of the following definitions:

1. The set of eigenvalues of A is called spectrum $\sigma(A)$ of A. It is a set in the complex plane.

2. The *spectral radius* is $\rho(A) \doteq max_{\lambda\in\sigma(A)}|\lambda|$, i.e. the maximum of the modulus of the eigenvalues.

3. The *kernel* is $Ker(A) \doteq \{u \in \mathbb{R}^n : Au = 0\}$. Thus, a singular matrix has a non-empty kernel.

4. The *matrix norm* of $B \in \mathbb{R}^{n\times n}$ is

$$\|B\|_* \doteq \max_{u\in\mathbb{R}^n\backslash\{0\}} \frac{\|Bu\|_*}{\|u\|_*}$$

   where $\|\cdot\|_*$ is a norm, i.e. $\|u\|_2 = \sqrt{u^T u}$ (Euclidean norm) or $\|u\|_A \doteq \sqrt{u^T Au}$ (norm of a vector induced by a matrix).

5. The *condition number* is defined by

$$\kappa_*(A) \doteq \|A\|_*\|A^{-1}\|_*.$$

   If $\kappa_*(A) \to \infty$ the matrix is becoming closer to singular; $\kappa_*(A) = \infty$ for a singular matrix. The condition number ($\kappa_*(A)$) is a measure of how easy one can solve a linear system with the matrix A.

6. A is *positive definite* if $u^T Au > 0$ for every $u \in \mathbb{R}^n$ (the product of $u^T Au \mapsto \mathbb{R}$ is an scalar).

7. If in addition A is *symmetric*, its eigenvalues are real and strictly positive, i.e., the eigenvalues lay on the positive part of the real line.

8. If the matrix A is positive definite and symmetric (s.p.d.), there exists an orthonormal basis of eigenvectors, i.e. linearly independent $\{v_1, ..., v_n\}$ with $v_i^T v_j = \delta_{ij}$ and $Av_i = \lambda_i v_i$. A is diagonalizable.

9. In a s.p.d. matrix A, the condition number (with respect to the Euclidean norm) is $\kappa(A) \doteq \kappa_2(A) \doteq \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$. Which can be proved as follows,

$$\|A\|_2 = \max_{u \in \mathbb{R}^n \setminus \{0\}} \frac{\|Au\|_*}{\|u\|_*} = \max_{u \in \mathbb{R}^n \setminus \{0\}} \frac{\|\alpha_i \lambda_i v_i\|_*}{\|\alpha_i v_i\|_*} \leq \lambda_{max}$$

where $u = \sum \alpha_i v_i$, since $u \in \{v_i, ..., v_n\}$ and $Av_i = \lambda_i v_i$. Moreover, we readily get $\frac{1}{\lambda_i} v_i = A^{-1} v_i$, i.e., the eigenvectors of A^{-1} are the same as the ones of A and its eigenvalues the inverse of the ones of A. As a result, we get

$$\|A^{-1}\|_2 = \sup_{u \in \mathbb{R}^n \setminus \{0\}} \frac{\|A^{-1}u\|_*}{\|u\|_*} = \sup_{u \in \mathbb{R}^n \setminus \{0\}} \frac{\|\alpha_i \frac{1}{\lambda_i} v_i\|_*}{\|\alpha_i v_i\|_*} \leq \frac{1}{\lambda_{min}}$$

Hence,

$$\kappa_2(A) \doteq \|A\|_2 \|A^{-1}\|_2 \doteq \lambda_{max} \cdot \frac{1}{\lambda_{min}}$$

10. If the condition number κ is close to 1, the matrix is *well-conditioned*. And, as $\kappa \to \infty$, the matrix is *ill-conditioned*. So that, if $\lambda_{min} \to 0$ or $\lambda_{max} \to \infty$ the matrix becomes harder to solve (since $\kappa \geq 1$ as $\lambda_{max} \geq \lambda_{min}$)

11. We can also use the Courant-Fischer (also called min-max) theorem to define eigenvalues of a symmetric matrix. We can define the Rayleigh quotient R_A of a symmetric matrix:

$$R_A(u) = \frac{(Au, u)}{(u, u)}.$$

Then, the eigenvalues of the matrix are

$$\lambda_{max} \doteq \max\{R(x) : x \neq 0\}, \qquad \lambda_{min} \doteq \min\{R(x) : x \neq 0\}.$$

Lemma 6.1: Let A and B be two s.p.d. matrices. The n eigenvalues of the following problem are equivalent:

$$\begin{aligned} Au &= \lambda Bu, \\ B{-}1Au &= \lambda u, \\ B^{-\frac{1}{2}}AB^{-\frac{1}{2}}u &= \lambda u, \end{aligned}$$

where $B^{-\frac{1}{2}}$ exists, since B^{-1} is symmetric and can be diagonalized, and so that $\sqrt{B^{-1}}$ exists. (We note that we can take the square root of a matrix by a change-of-basis transformation onto the one that diagonalises it, which exists due to the fact that B is s.p.d. Then, we take the squared root of the eigenvalues, which are all positive. Then, we transform the resulting matrix back to the original basis.) Thus, $\kappa(B^{-1}A) = \kappa(B^{-\frac{1}{2}}AB^{-\frac{1}{2}})$ Further,

$$\begin{aligned} \lambda_{min} &= \min_{u\in\mathbb{R}^n} \frac{u^TAu}{u^TBu} \\ \lambda_{max} &= \max_{u\in\mathbb{R}^n} \frac{u^TAu}{u^TBu} \end{aligned}$$

6.3 Finite Element Matrices

In this section, we want to analyse the condition number of finite element matrices. We will consider the matrices that arise from a reaction term (mass matrix) and a Laplacian term:

$$\int_\Omega u_h v_h \mathrm{d}\Omega, \qquad \int_\Omega \boldsymbol{\nabla} u_h \cdot \boldsymbol{\nabla} v_h \mathrm{d}\Omega, \qquad u_h,\ v_h \in V_h.$$

We consider the basis of shape functions, such that $V_h \doteq \mathrm{span}\{\phi^1, \dots, \phi^n\}$. The corresponding matrices thus read:

$$M_{ab} \doteq \int_\Omega \phi^a \phi^b \mathrm{d}\Omega, \qquad L_{ab} \doteq \int_\Omega \boldsymbol{\nabla}\phi^a \cdot \boldsymbol{\nabla}\phi^b \mathrm{d}\Omega, \qquad a,\ b \in \{1, \dots, n\}.$$

We note that any $u_h \in V_h$ can be written as $\sum_{a\in\{1,\dots,n\}} \phi^a u^a$. Thus, we have:

$$u^TMv = \langle Mv, u\rangle = \int_\Omega u_h v_h \mathrm{d}\Omega,$$

which is the interior product in $L^2(\Omega)$. Analogously,

$$u^T L v = \langle Lv, u\rangle = \int_\Omega \boldsymbol{\nabla} u_h \cdot \boldsymbol{\nabla} v_h \mathrm{d}\Omega,$$

which is an interior product in $H_0^1(\Omega)$. As a result,

$$u^T (M+L) v^T = \int_\Omega u_h v_h \mathrm{d}\Omega + \int_\Omega \boldsymbol{\nabla} u_h \cdot \boldsymbol{\nabla} v_h \mathrm{d}\Omega,$$

is an interior product in $H^1(\Omega)$. If we replace v by u above, we recover the corresponding (squared) norms. E.g.,

$$\begin{aligned} &u^T M u = \|u_h\|^2_{L^2(\Omega)}, \quad u^T L u = \|\boldsymbol{\nabla} u_h\|^2_{L^2(\Omega)} = \|u_h\|^2_{H_0^1(\Omega)}, \\ &u^T (M+L) u = \|u_h\|^2_{H^1(\Omega)}. \end{aligned}$$

Mass matrix

In this section, we will check that the mass matrix is conditioned as follows:

$$c h^d \leq \frac{\langle Mv, v\rangle}{\langle v, v\rangle} \leq C h^d, \quad \forall v \in \mathbb{R}^n.$$

Using the expression of Rayleigh quotients above and the relation with the eigenvalues, this is equivalent to say that λ_{min} and λ_{max} are equal to a constant times h^2.

In order to prove the upper bound, we can do the following:

$$\begin{aligned} \int_\Omega u_h^2 &= \sum_{a,b\in(1,\dots,n)} \int_\Omega (u^a \phi^a)(u^b \phi^b) \\ &= \sum_{a\in(1,\dots,n)} \int_\Omega (u^a \phi^a)(u^a \phi^a) + \sum_{a\in(1,\dots,n)} \sum_{b\in(1,\dots,n)\setminus a} \int_\Omega (u^a \phi^a)(u^b \phi^b) \end{aligned}$$

We can prove that

$$\sum_{a\in(1,\dots,n)} \sum_{b\in(1,\dots,n)\setminus a} \int_\Omega (u^a \phi^a)(u^b \phi^b) \lesssim \sum_{a\in(1,\dots,n)} \int_\Omega (u^a \phi^a)(u^a \phi^a)$$

using Cauchy-Schwarz and Young inequalities, and using the fact that a node has a finite number of neighbours independent of the number of nodes

(it also depends on the shape regularity of the mesh). The lower bound can be bounded by decomposing the integral into cell components $K \in \mathcal{T}_h$ ($\mathcal{T}_h$ is the mesh) and writing the integral in the reference element:

$$\int_\Omega u_h^2 = \sum_{a,b\in(1,\dots,n)} u^a u^b \sum_{K\in\mathcal{T}_h} \int_{\hat{K}} \hat{\phi}^a \cdot \hat{\phi}^b |J_K| \mathrm{d}\hat{K}.$$

We have that $|J_k| = Ch^d$ by a scaling argument, since the geometrical map transforms $\hat{K}$ (of diameter 1) into K (of diameter h). So, we only need to look at the eigenvalues of the reference FE mass matrix

$$\hat{M} = \int_{\hat{K}} \hat{\phi}^a \hat{\phi}^b \mathrm{d}\hat{K}, \quad a,\ b \in \{1,\dots,\hat{n}\},$$

where $\hat{n}$ are the shape functions in the reference element (e.g., $\hat{n}$ is 2 for a linear element in 1D).

Denoting its minimum eigenvalue by $\lambda_{min}(\hat{M})$ and using again the fact that the number of neighbours is bounded, we readily obtain:

$$\int_\Omega u_h^2 \geq Ch^d \lambda_{min}(\hat{M}) \langle u, u\rangle .$$

We can use the same idea to bound the maximum eigenvalue, i.e.,

$$\int_\Omega u_h^2 \leq Ch^d \lambda_{max}(\hat{M}) \langle u, u\rangle .$$

We can readily check that the maximum and minimum eigenvalues of $\hat{M}$ are constants. They cannot depend on h, since they are defined at the reference element. The eigenvalues can be explictly computed but the specific value is not important here. It is easy to check that the reference mass matrix is also non-singular, since it is a s.p.d. matrix. Thus, we have proved the desired bound on the Rayleigh quotient and proved that the eigenvalues are $\mathcal{O}(h^d)$.

As a result, the condition number is $\kappa(M) = \mathcal{O}(1)$. It does not depend on h. Thus, mesh refinement does not negatively affect the condition number.

Laplacian matrix

We proceed similarly for the Laplacian matrix. We want to check the following bound:

$$ch^2 \leq \frac{\langle Av, v\rangle}{\langle v, v\rangle} \leq C, \quad \forall \in \mathbb{R}^n.$$

It is equivalent to say that $\lambda_{\min} \geq ch^2$ and $\lambda_{\max} \leq C$. The proof of the upper bound is a direct consequence of the inverse inequality and the previous estimates for the mass matrix:

$$\langle Au, u \rangle = \|\boldsymbol{\nabla} u_h\|^2 \leq Ch^{-2}\|u_h\|^2 \leq Ch^{d-2} \langle u, u \rangle .$$

The upper bound relies on the Poincaré inequality and the mass matrix bounds:

$$\langle Au, u \rangle = \|\boldsymbol{\nabla} u_h\|^2 \geq C\|u_h\|^2 \geq Ch^{d-2} \langle u, u \rangle .$$

So, the condition number $\kappa(A) = (O)(h^{-2})$. As a result, as we refine the mesh and reduce the FE characteristic size, the condition number increases quadratically with $1/h$. Thus, as we refine the mesh the problem becomes more ill-conditioned and it will be harder to solve.

In the following questions, we briefly introduced two different types of algorithms to solve linear systems, i.e., direct and iterative methods.

6.4 Direct methods

Direct methods are very robust but computationally intensive. Those methods are applicable to any matrix, not only FE matrices. Most of them are based on *Gaussian elimination*. The Gaussian elimination process works as follows: Given $A := A^{(0)}$, compute

$$\begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \dots & a_{1n}^{(0)} \\ a_{21}^{(0)} & a_{22}^{(0)} & \dots & a_{2n}^{(0)} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1}^{(0)} & a_{n2}^{(0)} & \dots & a_{nn}^{(0)} \end{bmatrix} \rightarrow \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \dots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{bmatrix} \dots$$

$$\rightarrow \begin{bmatrix} a_{11}a_{n1}^{(0)} & a_{12}^{(0)} & \dots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{nn}^{(n-1)} \end{bmatrix}.$$

The first step (related to the first column) can be represented as

$$A(2,:) - A(1,:) \times \frac{a_{21}}{a_{11}}, \qquad \dots, \qquad A(n,:) - A(1,:) \times \frac{a_{n1}}{a_{11}},$$

which can be written as $A_{(1)} = L_{(1)}^{-1} A$, where

$$L_{(1)}^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -\frac{a_{21}^{(0)}}{a_{11}^{(0)}} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{a_{n1}^{(0)}}{a_{11}^{(0)}} & 0 & \dots & 1 \end{bmatrix},$$

with inverse

$$L_{(1)} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \frac{a_{21}^{(0)}}{a_{11}^{(0)}} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{a_{n1}^{(0)}}{a_{11}^{(0)}} & 0 & \dots & 1 \end{bmatrix}.$$

The we proceed for all columns (in a compact form), $A_{(n-1)} \doteq U = L_{(n-1)}^{-1} \dots L_{(1)}^{-1} A$. Finally, we get the LU factorization of $A = LU$, with

$$L = L_{(1)} \dots L_{(n-1)} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \frac{a_{21}^{(0)}}{a_{11}^{(0)}} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{a_{n1}^{(0)}}{a_{11}^{(0)}} & \frac{a_{n2}^{(1)}}{a_{22}^{(1)}} & \dots & 1 \end{bmatrix}, \quad U = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \dots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{nn}^{(n-1)} \end{bmatrix}$$

where L is a lower diagonal matrix composed of the product of $L_{(i)}$ matrices as $L = \prod_{i=1}^{n-1} L_{(i)}$ and U is the resulting upper diagonal matrix. Then, we can solve $Ax = f$ via forward ($Ly = f$) and backward ($Ux = y$) substitution, which are very fast algorithms. However, the computational cost is the computation of L and U matrices. The system is solved as follows,

$$Ax = f \quad \rightarrow \quad L(Ux) = f \quad \rightarrow \quad Ly = f, \quad Ux = y.$$

Some properties of direct methods are stated below.

First, we note that the complexity is $\mathcal{O}(n^3)$ (number of floating-point operations) and $\mathcal{O}(n^2)$ (required memory) for dense matrices. The complexity estimates can be obtained by counting the number of operations (numbers to be stored) in the Gaussian elimination process.

However, FE matrices are highly sparse ($nnz << n$, where nnz is the number of non-zeros per row). The number of non-zeros depend on the spacial dimension and the order of discretization, i.e. it does depends on the local DOFs. For sparse matrices, the complxity is reduced to $\mathcal{O}(n^2)$ flops and $\mathcal{O}(n^{1.5})$ memory for 3D FE matrices (the constants depend on nnz). For a sparse matrix, one can readily check that the memory is $\mathcal{O}(n^1)$ (more especifically, $n \times nnz$) but the filling process of L and U increase the memory up to $\mathcal{O}(n^{1.5})$.

As indicated above, direct methods have high computational and memory demands. Direct methods are also very hard to parallelise (not scalable on more than $\mathcal{O}(100)$ processors on distributed memory machines). As a result, they are not acceptable for large scale simulations of 3D complex physical phenomena or realistic engineering applications. In order to alleviate this problem, we usually rely onon iterative solvers with robust preconditioners in these situations, which we briefly describe below.

6.5 Iterative solvers

Iterative solvers approximate the solution of the linear system in an iterative way and up to some tolerance for a given convergence criterium. The most advanced iterative algorithms (e.g., multigrid algorithms) can attain linear complexity in terms of flops and memory consumption (with respect to the problem size), which is a dramatic improvement with respect to direct methods. Furthermore, they are more amenable for parallel computations on distributed memory computers. Other very effective methods are the so-called domain decomposition methods, which are based on a divide-and-conquer approach, and thus, very well-suited for large scale parallel computions on millions of processors. However, the multigrid and domain decomposition methods are quite involved, and we do not cover them in this unit. Instead, we consider some basic iterative solvers that are building blocks for multigrid methods.

6.5.1 Richardson method

Let us consider a matrix B such that,

$$B = A + E,$$

where E is the perturbation matrix, i.e. the error of approximating A by using B. We call B a *preconditioner* for A. The ideal preconditioner B should be such that the action of B^{-1} is cheap to compute and/or easy to parallelise and B is close to A ($B^{-1}A$ is close to the identity matrix, i.e., its condition number is close to one).

We aim at solving $Au = f$, which is now equivalent to $Bu = f + Eu$. In the (preconditioned) Richardson method, we consider the following iterative procedure: Given u^0, compute for $k = 0, \ldots$

$$Bu^{k+1} = f + Eu^k.$$

If $B = I$, the solver is called *Richardson* method. Otherwise, we call it the *preconditioned Richardson method*.

First, we can check that, if the iterative process converges, then it converges to u. Let us define the so-called residual $r \doteq f - Au^k$. Since

$$Bu^{k+1} = f + Eu^k = f - Au^k + Bu^k = r^k + Bu^k,$$

we can write each iterate as

$$u^{k+1} = u^k + B^{-1}r^k = u^k + B^{-1}(f - Au^k).$$

The method stops when the r^k is zero, i.e., u^k is solution of the linear system. Since we assume that A is nonl-singular, $u^k = u$.

In order to improve the convergence of the method, we usually make us of a *relaxation*, i.e., we compute

$$u^{k+1} = u^k + \alpha B^{-1}(f - Au^k), \qquad \alpha \in (0, 1) \tag{6.1}$$

where $\alpha > 0$ is the *relaxation factor*. It only takes part of the proposed correction to improve convergence. α small provides a more stable algorithm but it can also make convergence slower in some cases (think what happens when $\alpha = 0$).

Error of Richardson method

Let us analyse the error $e^k = u - u^k$ at each iterate. We subtract u on both sides of (6.1) to get:

$$u^{k+1} - u = u^k - u + \alpha B^{-1}(f - Au^k).$$

Then, since $f = Au$, we get:

$$e^{k+1} = e^k - \alpha B^{-1}(Ae^k) = (I - \alpha B^{-1}A)e^k.$$

Let us assume that A and B are s.p.d. matrices. From the previous lemma, we know that the eigenvalues $\lambda_i(B^{-1}A)$ of $B^{-1}A$ are real and positive. Then, the eigenvalues of $(I - \alpha B^{-1}A)$ are $(1 - \alpha\lambda_i(B^{-1}A))$. As a result, using the definition of the matrix norm, we have:

$$\begin{aligned}\|e^{k+1}\| &= \|(I - \alpha B^{-1}A)e^k\| \leq \|I - \alpha B^{-1}A\|\|e^k\| \\ &\leq \max_i |1 - \alpha\lambda_i(B^{-1}A)|\|e^k\| = \rho(I - \alpha B^{-1}A)\|e^k\|.\end{aligned}$$

As a result, we have (using recurrency):

$$\|e^{k+1}\| \leq \rho(I - \alpha B^{-1}A)^k\|e^0\|.$$

We can readily observe that the method will converge if $\rho(I-\alpha\sigma(B^{-1}A)) < 1$, which requires a matrix B close to A. However, in general, it does not converge. Besides, one can also check that

$$\begin{aligned}\rho(I - \alpha B^{-1}A) &= \max_i |1 - \alpha\lambda_i(B^{-1}A)| \\ &= \max(|1 - \alpha\lambda_{\min}(B^{-1}A)|, |1 - \alpha\lambda_{\max}(B^{-1}A)|).\end{aligned}$$

As a result, it is easy to check that the method will always converge when $0 < \alpha < \frac{2}{\lambda_{max}}$, whereas the optimal choice is $\alpha = \frac{2}{\lambda_{max}+\lambda_{min}}$. However, the computing of λ_{max} and λ_{min} is usually more expensive than solving the system and this optimal value cannot be used in practise.

In summary, this method is simple but does not converge in general. With relaxation, the convergence is slow. The value of α that ensures converge requires to know $\sigma(A)$, which is far more expensive that solving the linear system itself. Besides, there is no maximum bound for the number of iterations required for convergence. As we can see, there is room for improvement.

6.5.2 Steepest descent method

This method is based on the fact that the solution of $Ax = f$ is the minimum of

$$\Phi(x) = \frac{1}{2}x^T Ax - f^T x$$

for A s.p.d., since $\partial_x \Phi(x) = Ax - f \doteq -r$. Considering the unpreconditioned Richardson method,

$$u^{k+1} = u^k + \alpha_k r^k,$$

we can compute tha value of α that minimises the functional. We can check that (try to prove it!)

$$\alpha_k = \text{argmin}_{\alpha \in \mathbb{R}} \Phi(u^k + \alpha r^k) = \frac{\langle r^k, r^k \rangle}{\langle r^k, Ar^k \rangle}.$$

Then, the *steepest descent method* is defined by: Given u^0, compute for $k = 0, \ldots$

$$u^{k+1} = u^k + \alpha_k M^{-1}(f - Au^k),$$

with $\alpha_k = \frac{\langle r^k, r^k \rangle}{\langle r^k, Ar^k \rangle}$.

This method represents an improvement with respect to Richardson method. No knowlege of the spectrum of A ($\sigma(A)$) is required. One can prove that the error is reduced as

$$\|e^k\|_A \leq \left(\frac{\kappa(A) - 1}{\kappa(A) + 1} \right)^k \|e^0\|_A,$$

which implies convergence. The computational cost per iteration is the same as Richardson (plus some inner products). At each iteration, we perform matrix-vector products with sparse matrices, which have a complexity $\mathcal{O}(n)$. However, the number of iterations required to attain convergence is not bounded. As a result, the total cost, which is $\mathcal{O}(n)$ times the number of iterations is not bounded either.

6.5.3 Conjugate Gradient

In the steepest descent algorithm, the distance can be optimally chosen but the direction is not. In order to get an optimal direction, assume we have a basis of *conjugate directions* $\{p^k\}_{i=0}^{n-1}$, i.e.

$$\langle p^i, Ap^j \rangle = 0, \qquad i \neq j$$

i.e., an A-orthogonal basis. Then, we compute for $k = 0, \ldots, n-1$

$$u^{k+1} = u^k + \alpha_k p^k$$

with $\alpha^k = \frac{\langle r^k, p^k \rangle}{\langle p^k, Ap^k \rangle}$ as above. It is easy to check that this algorithm converges to the solution. The number of iterations is fixed (equal to the dimension of the problem). In fact, CG was originally proposed as a direct solver. However, the algorithm is severely affected by rounding errors and the residual can be not close to zero up to machine precision after n iterations. Years after it was originally proposed, the method was rediscovered as a powerful iterative method, probably the most used method in practise. The reason is that after far less iterations than the total size n, one can get residual below acceptable tolerances for most practical applications.

The open question is how to build the A-orthogonal basis. The CG method performs this computation in an iterative way. Given the residual r^k, we can compute:

$$p^k = r^k + \sum_{i=0}^{k-1} \beta^i p^i,$$

with $\beta^i = -\frac{\langle Ar^k, p^i \rangle}{\langle Ap^i, p^i \rangle}$, i.e. β^i s.t. $\langle Ap^k, p^i \rangle = 0$.

This approach requires full orthogonalization with respect to the previous directions, i.e. in each step it has to orthogonalise with all the previous directions. All the previous directions have to be stored in memory and the computational cost increases with k.

The good news is that full orthogonalization is not required (in exact arithmetics). The *key Property of CG* is that we only need to orthogonalize with respect to the previous direction. Hence, there is no need to store all the previous directions and the computational cost is dramatically reduced. The algorithm is shown as follows with x^0 as input and x as output:

$p^0 := r^0 := b - Ax^0$
for $j = 0, \ldots,$ till CONV **do**
  $s^{j+1} = Ap^j$
  $\alpha_j := (r^j, r^j)/(s^{j+1}, p^j)$
  $x^{j+1} := x^j + \alpha_j p^j$
  $r^{j+1} := r^j - \alpha_j s^j$
  $\beta_j := (r^{j+1}, r^{j+1})/(r^j, r^j)$
  $p^{j+1} := r^{j+1} + \beta_j p^j$
end for

We stress that this algorithm is only applicable on s.p.d. matrices, since it relies on the re-statement of the problem as a quadratic form. The con-

vergence analysis of the method provides the following bound:

$$\|e^k\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^k \|e^0\|_A.$$

We observe that this method always converges (in exact arithmetics).

6.5.4 Preconditioned Conjugate Gradient

Looking at the convergence estimates, in order to reduce the number of iterations of CG, we would like $\kappa(A) \sim 1$. Since we cannot change the matrix, what we can do is to precondition the system. Let us consider a s.p.d. B preconditioner for A. In this situation, one can compute $B^{1/2}$ (using diagonalisation and taking the square root of the eigenvalues). Next, we can define the equivalent system:

$$Au = b \quad \rightarrow \quad B^{-1/2}AB^{-1/2}v = B^{-1/2}b, \quad u = B^{1/2}u.$$

If B is a good approximation of A, $\kappa(B^{-1/2}AB^{-1/2}) \sim 1$ and the CG methods will converge very fast. We note, as explained above, that $\kappa(B^{-1/2}AB^{-1/2}) = \kappa(B^{-1}A)$. Hence, we have that

$$\|e^k\|_A \leq 2\left(\frac{\sqrt{\kappa(B^{-1}A)}-1}{\sqrt{\kappa(B^{-1}A)}+1}\right)^k \|e^0\|_A.$$

CG can readily be applied for the preconditioned system above, with system matrix $B^{-1/2}AB^{-1/2}$, since the matrix is s.p.d. However, the computation of $B^{-1/2}$ is very costly and complex. Instead, we would rather use the (left) preconditioned system

$$Au = b \rightarrow B^{-1}Au = B^{-1}b.$$

Unfortunately, $B^{-1}A$ is not symmetric and CG cannot be applied. Fortunately, we can re-arrange the algorithm in such a way that we can still use CG without the need to apply $B^{-1/2}$. The preconditioned CG method only requires the action of B^{-1} but is equivalent to CG for $B^{-1/2}AB^{-1/2}$:

$p^0 := r^0$, $z^0 = B^{-1}r^0$
for $j = 0, \ldots,$ till CONV **do**

$s^{j+1} = Ap^j$
$\alpha_j := (z^j, r^j)/(s^{j+1}, p^j)$
$x^{j+1} := x^j + \alpha_j p^j$
$r^{j+1} := r^j - \alpha_j s^j$
$z^{j+1} := B^{-1} r^{j+1}$
$\beta_j := (z^{j+1}, r^{j+1})/(z^j, r^j)$
$p^{j+1} := z^{j+1} + \beta_j p^j$
end for

The computational cost per PCG iteration is a matrix-vector product with A and the action of B^{-1}. Now, the open question is how to define a cheap and robust preconditioner B for A. And more especifically, how to do that for A coming from the FE discretisation of physical problems. As commented above, two kind of techniques are multigrid and domain decomposition methods. This is an active field of research and there is no time to explore the design of preconditioners in this unit.