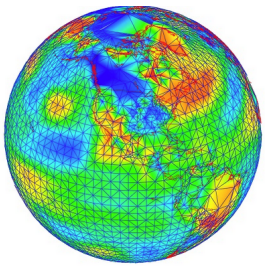
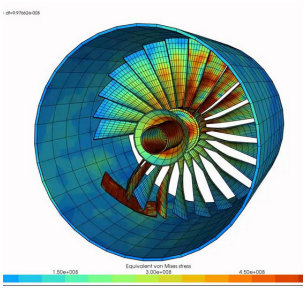




(L) A FE mesh in Earth Sciences. (M) A FE analysis of an aircraft engine: Von Mises stress values. (R) Model reduction
Images source: ...

INSA Toulouse, Applied Mathematics department

Finite Element Methods, PDE Models Reductions

by J. Monnier (jerome.monnier@insa-toulouse.fr)

Spring 2024

*

1-Analysis of Elliptic Problems: weak solutions

2-Finite Element methods: fundamentals

3-Finite Element methods: complements

4-Weak constraints: mixed formulations

5-Model Reductions: reduced basis (POD) & Neural Networks (NN) based methods

*

Full content of the course, exercises and programming practicals: consult the INSA Moodle page.

Contents

Part I

Mathematical Analysis

Chapter 1

Analysis of Elliptic Problems: Variational Forms, Weak Solutions

This chapter follows in great part the presentation done in the excellent book of G. Allaire entitled " Numerical analysis and optimization", Oxford scientific publications, 2007 [?].

Prerequisites for this chapter are :

- Basic knowledge on the classical linear PDEs: elliptic (Laplace-Poisson's equation), parabolic (heat equation), hyperbolic (advection equation and wave equation),
- Basics concepts of weak derivatives, the Sobolev space $H^1(\Omega)$.

In this chapter you will learn how to:

- write the weak formulation of a Boundary Value Problem (BVP) (PDE-based model),
- write the energy expression in the symmetric case,
- impose various boundary conditions including the transmission b.c.,
- prove the well-posedness of a linear elliptic problem (Lax-Milgram's theory),
- identify the potential origin of a local singularity.

To the INSA students 4th year, Applied Math. department.
The sections 1.1, 1.2 and 1.3 and 1.4.1 have already been studied in great part during the INSA PDE course, 4th year, 1st semester.

The section indicated "To go further" may be skipped as a first reading. These additional information are dedicated to students particularly interested in mathematical analysis.

In Appendix 1.4, basic properties of the Laplace operator are presented. The reader is invited to consult this section by its own before studying the present analysis chapter.

1.1 Introduction

The typical elliptic boundary value problem we will consider is the Laplace-Poisson equation accompanied with mixed boundary conditions:

$$\begin{cases} -\Delta u(x) & = f(x) \text{ in } \Omega \\ u(x) & = 0 \text{ on } \Gamma_d \\ -\partial_n u(x) & = \varphi(x) \text{ on } \Gamma_n \end{cases} \tag{1.1.1}$$

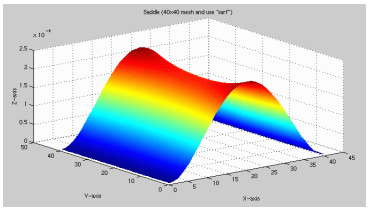


Figure 1.1.1: Solution de l'équation biharmonique: déplacement d'une plaque compressée sur deux de ses bords (u est imposé sur ces bords-ci) et libre sur les deux autres bords.

with $\partial\Omega = \Gamma_n \cup \Gamma_d$.
It is the typical 2nd order linear elliptic equation modeling diffusive phenomena in a wide range of applications.

An other typical problem is the biharmonic problem:

$$\begin{cases} -\Delta^2 u(x) & = f(x) \text{ in } \Omega \\ u(x) & = 0 \text{ on } \partial\Omega \end{cases}$$

with:$\Delta^2 u(x) = \sum_{i=1}^n \frac{\partial^4 u}{\partial x_i^4}(x) + 2\sum_{i,j=1;i<j}^n \frac{\partial^4 u}{\partial x_i^2 \partial x_j^2}(x)$.
The latter is a 4th order linear elliptic equation modeling for example a flexion plate (linear elasticity): u represents the plate displacement, cf Fig.

We will recall the adequate mathematical framework to obtain these elliptic boundary value problems well-posed: the variational (or weak) formulation.

Recall that *a problem is well-posed if* it has a unique solution *and* it depends continuously on the data (e.g. the R.H.S. also source term f(x)).

The variational formulation has a natural physical interpretation of the mathematical system, and it is the key step to derive a finite element scheme.

1.2 From classical formulation to weak formulation

Let us consider the Boundary Value Problem (BVP) (1.1.1). Given "regular" data, in the sense $f \in C(\bar{\Omega})$ and $\varphi \in C^1(\Gamma_n)$, we can expect that the solution u is in $C^2(\Omega) \cap C^0(\bar{\Omega})$. However, in this case the solution may not exist (at least without additional regularity on the source term f for example).
This classical formulation of the problem, see (1.1.1), also called "strong" formulation, leads to the classical (or "strong") solution.

In a great majority of real-world modeling problems, data are not regular. Mind for example to a simple discontinuity of the source term f.

Then the right framework to analyse this Boundary Value Problem (BVP) is the *variational approach,* leading to the so-called *variational or weak formulation* and the corresponding weak solution.

The principle of the variational approach is to multiply the equation by an arbitrary function, called a test function, next to integrate it using Green's formula.

In the physical / mechanical communities the weak formulation is called the *principle of virtual work.*

1.2.1 Domain regularity and basic recalls

Let Ω be a regular open set of $\mathbf{R}^n$. Let n be the unit the outward normal vector at the boundary $\partial\Omega$.

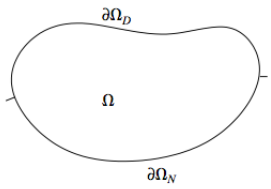


Figure 1.2.1: Regular domain (with a partition of its boundary)

Figure 3.2. Two examples of a nonregular open set: open set with a crack on the left, open set with a cusp on the right.

Figure 1.2.2: Non-regular domains: local singularities...

What is a "regular" domain ?

The domains plotted below are regular: corners are not locally C^1, however they are Lipschitz and that regularity is enough to apply all the properties presented in this course (in particular the Sobolev spaces properties).

What is a non-regular domain ?

The typical example is an internal crack which generates a local singularity of the solution at the crack front... e.g. an infinite electrical field at your umbrella top during a storm... (the model of the electrical potential is the simple Laplace equation).

In all the sequel we assume that the domain Ω is a regular bounded open-set of $\mathbf{R}^d$.

Green's formula. Let w be a regular function, $w \in C^1(\bar{\Omega})$, then

$$\int_\Omega \partial_i w(x)\ dx = \int_{\partial\Omega} w(x) \cdot n_i(x)\ ds, \quad i = 1, .., d \tag{1.2.1}$$

Integration by parts. Let u and v be regular functions, i.e. in $C^1(\bar{\Omega})$, then:

$$\int_\Omega u(x)\partial_i v(x)\ dx = -\int_\Omega v(x)\partial_i u(x)\ dx + \int_{\partial\Omega} u(x)v(x)n_i(x)\ ds, \quad i = 1, .., d \tag{1.2.2}$$

And for u in $C^2(\bar{\Omega})$,

$$\int_\Omega \Delta u(x)v(x)\ dx = -\int_\Omega \nabla u(x)\nabla v(x)\ dx + \int_{\partial\Omega} \partial_n u(x)v(x)\ ds \tag{1.2.3}$$

where $\partial_n u$ denotes $\nabla u \cdot n$.

It will be recalled in next section that *these Green's formula and integration by parts remain valid for u and v in Sobolev spaces.*

In the sequel, we need the following result. We denote by $C_c^\infty(\Omega)$ the space of functions C^∞ with compact support in Ω. (This space is often denoted $\mathcal{D}(\Omega)$ too).

Lemma 1. *Let $g \in C^0(\Omega)$ (resp. $L^2(\Omega)$). If*

$$\int_\Omega g(x)\varphi(x)dx = 0\ \forall \varphi \in C_c^0(\Omega)$$

then $g = 0$ in Ω (resp. almost everywhere in Ω).

Sketch of the proof.

In the continuous case, this can be easily proved by contradiction (assume that g strictly positive at one point x_0 and consider $\varphi > 0$... See details in [**?**] p72.

Next the result in L^2 follows by density of $C_c^0(\Omega)$ in $L^2(\Omega)$, see details in [?] p80.

1.2.2 Weak formulation in the classical spaces $C^k(\bar{\Omega})$

Recall that Ω denotes a regular bounded open-set of $\mathbf{R}^d$.

1.2.2.1 Weak formulation

Let us assume that all the *data of the BVP are regular enough*; typically in the Laplace problem (1.1.1), it means that f is $C^0(\Omega)$ and $\varphi \in C^0(\Gamma_n)$.

Then we have

Proposition 2. *Let X be the space defined by $X = \{v \in C^1(\bar{\Omega}),\ v = 0 \text{ on } \Gamma_d\}$.*
Then u is solution of the BVP (1.1.1) in $C^2(\bar{\Omega})$ if and only if $u \in X$ and u satisfies:

$$\int_\Omega \nabla u(x)\nabla v(x)\ dx = \int_\Omega f(x)v(x)\ dx - \int_{\Gamma_n} \varphi(x)v(x)\ ds\ \forall v \in X \tag{1.2.4}$$

Proof. We multiply the equation (1.1.1) by $v \in X$. An integration by part gives:

$$\int_\Omega \nabla u(x)\cdot\nabla v(x)dx - \int_{\Gamma_n} \nabla u \cdot n v(x)\ ds = \int_\Omega f(x)v(x)dx$$

since $v = 0$ on Γ_d. Hence the weak formulation (1.2.4).

Conversely, let $u \in X$ be solution of the weak formulation.
By making the "reverse" integration by part on Eqn (1.2.4), it follows:

$$\int_\Omega -\Delta u(x)v(x)\ dx + \int_{\Gamma_n} \nabla u \cdot n\ v(x)\ ds = \int_\Omega f(x)v(x)\ dx - \int_{\Gamma_n} \varphi(x)v(x)\ ds\ \forall v \in X$$

Hence for $v = 0$ on $\partial\Omega$ (v still belongs to X):

$$\int_\Omega (\Delta u(x) + f(x))\ v(x)\ dx = 0$$

This equation holds for any v in $C^0(\Omega)$.
Moreover $(\Delta u + f)$ is a continuous function; therefore:

$$-\Delta u(x) = f(x) \qquad \forall x \in \Omega$$

The Dirichlet b.c. on Γ_d is a direct consequence of $u \in X$.
Next for all $v \in X$, $\int_{\Gamma_n} \nabla u \cdot n\ v(x)\ ds = -\int_{\Gamma_n} \varphi(x)v(x)\ ds$. Therefore:

$$-\nabla u \cdot n = \varphi(x) \text{ on } \Gamma_n$$

Therefore u is solution of the classical formulation (1.1.1). □

Equation (1.2.4) is the *variational formulation* (or weak form) of the BVP (1.1.1).
The function v is called the *test function.*
On the opposite, (1.1.1) is called the *classical form* (or strong form) of the model.

Remarks

- The weak form requires $u \in C^1(\bar{\Omega})$ "only" while the classical form requires $u \in C^2(\bar{\Omega})$. As a consequence the weak form is more general since considering a greater set of potential solutions.
- In the mechanical community, the variational formulation is called the *principle of virtual work.*

1.2.2.2 Why a variational (weak) formulation ? Why the Sobolev spaces ?

The variational (weak) formulation is the right framework to consider the BVP for few reasons.

It requires a less regular solution than in the classical form hence less regular data too. In real-world modeling problems, data are not necessarily regular...

It is the right formulation to mathematically analyse the problem, in particular to establish the existence and uniqueness of the solution, see the Lax-Milgram theory in next section.

However this theory works only if the space in which we look at the solution is a *Hilbert space*. This is not the case of X defined as above since X is a Banach space with its norm not defined from a scalar product. (If so, it would not be complete for its induced norm, see e.g. [**?**] for more details).

Then for linear elliptic equations, the natural spaces are the Sobolev spaces $H^m(\Omega)$, $m \geq 1$ (see their definitions in next paragraph).

The variational (weak) formulation of (1.1.1) reads as follows.

Find $u \in X$ such that:

$$a(u,v) = b(v) \ \forall v \in X \tag{1.2.5}$$

With:

$$a(u,v) = \int_\Omega \nabla u(x) \nabla v(x) \ dx \text{ and } b(v) = \int_\Omega f(x)v(x) \ dx - \int_{\Gamma_n} \varphi(x)v(x) \ ds$$

It can noticed that the applications $v \mapsto a(\cdot, v)$ and $v \mapsto l(v)$ are *linear* by construction.

Furthermore the application $u \mapsto a(u, \cdot)$ is linear since the original PDE is linear; therefore $a(u,v)$ is a *bilinear* form on $X \times X$.

1.2.3 Recalls of functional analysis

Recall that Ω denotes a regular bounded open set of $\mathbf{R}^d$.

Weak derivatives Let us recall the definition of the weak derivative for functions in $L^2(\Omega)$, the space of measurable functions which are square integrable in Ω.

(Recall that measurable functions are defined almost everywhere: if we change the value of a measurable function on a subset of measure zero, the measurable function is not changed).

Weak derivatives is a generalization of the classical derivation. (Note that it is a particular case of the derivation in the sense of distributions).

Definition 3. Let $v \in L^2(\Omega)$. v is differentiable in the weak sense in $L^2(\Omega)$ if it exists $w_i \in L^2(\Omega)$, $i = 1, .., d$, such that:

$$\int_\Omega v(x) \partial_i \varphi(x) dx = - \int_\Omega w_i(x) \varphi(x) dx \quad \forall \varphi \in C_c^\infty(\Omega)$$

Then $w_i \equiv \partial_i v$ is the ith partial derivative of v.

Definition 4. Of course, if v is differentiable in the classical sense (and its partial derivatives belong to $L^2(\Omega)$) then the classical and the weak derivatives of v are the same.

A *practical criteria* to determine if a function is differentiable in the weak sense is as follows.

Lemma 5. *Let $v \in L^2(\Omega)$. If it exists $C > 0$ such that for $i = 1, .., d$,*

$$\int_\Omega v \partial_i \varphi(x) \ dx \leq C \|\varphi\|_{L^2} \quad \forall \varphi \in L^2(\Omega) \text{ with } \partial_i \varphi \in L^2(\Omega)$$

then v is differentiable in the weak sense.

Note that the definition the standard differential operators like $\text{div}, \nabla, \Delta, \Delta^2, curl$ etc can be naturally extended to the weak sense.

The Sobolev spaces $H^1(\Omega)$, $H^1_0(\Omega)$, $H^m(\Omega)$ Let us recall the Sobolev spaces which are the natural spaces to solve the variational formulations of *elliptic PDEs* and particularly the *linear* ones (see next paragraph).

The Sobolev space $H^1(\Omega)$ is defined by:

$$H^1(\Omega) = \{v \in L^2(\Omega) \text{ s.t. } \partial_i v \in L^2(\Omega),\ \forall i = 1, .., d\}$$

where $\partial_i v$ denotes the *weak partial derivative* of v.

This space is equipped with the scalar product $(u, v)_{H^1} = \int_\Omega (u(x)v(x) + \nabla u(x)\nabla v(x)]\ dx$ and with the corresponding norm $\| \cdot \|_{H^1} = (\cdot, \cdot)^{1/2}_{H^1}$. Then it is a *Hilbert space.*

The space $H^1_0(\Omega)$ is the subspace of $H^1(\Omega)$ such that: $v = 0$ on $\partial\Omega$. Equipped with the scalar product of $H^1(\Omega)$, the Sobolev space $H^1_0(\Omega)$ is a Hilbert space too.

Remarks

- The functions of $H^1(\Omega)$ are a-priori neither continuous nor bounded (excepted in dimension $d = 1$; which is an exception).
- Regular functions in the sense $C^k(\bar{\Omega})$, $k \geq 1$, are dense in $H^1(\Omega)$. This feature is crucial to establish many properties first for regular functions next to functions in H^1 by the "density argument".
  This means that for all $f \in H^1(\Omega)$, it exists a sequence $f \in C^\infty_c(\bar{\Omega})$ such that $\lim_n \|f - f_n\|_{H^1} = 0$.

Let us denote $\alpha = (\alpha_1, ..., \alpha_d)$, $|\alpha| = \sum_i \alpha_i$, and: $\partial^\alpha v(x) = \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} ... \partial x_d^{\alpha_d}}(x)$. At higher orders, the Sobolev space $H^m(\Omega)$, $m \geq 1$, is naturally defined as follows:

$$H^m(\Omega) = \{v \in L^2(\Omega) \text{ s.t. } \forall \alpha \text{ with } |\alpha| \leq m,\ \partial^\alpha v \in L^2(\Omega)\}$$

Obviously the more m is large, the more the functions of $H^m(\Omega)$ are regular, that is differentiable in the classical sense. More precisely, we have:

Lemma 6. *Let Ω be a bounded open set of $\mathbf{R}^d$. If $m > \frac{d}{2}$ then $H^m(\Omega) \subset C^0(\bar{\Omega})$.*

More generally, for k integer, $k \geq 0$ such that $m - \frac{d}{2} > k$, then:

$$H^m(\Omega) \subset C^k(\bar{\Omega}) \tag{1.2.6}$$

In particular, we have:

- In 1d geometry (unusual case), $H^1(\Omega) \subset C^0(\bar{\Omega})$, $H^2(\Omega) \subset C^1(\bar{\Omega})$ etc
- In 2d and 3d geometries, $H^1(\Omega)$ functions are not continuous on Ω !...
  However, $H^2(\Omega)$ functions are: $H^2(\Omega) \subset C^0(\bar{\Omega})$, $H^3(\Omega) \subset C^1(\bar{\Omega})$ etc

Lemma 7. ***Poincaré inequality***

Let us consider $v \in H^1(\Omega)$, with $v = 0$ on $\partial\Omega$. It exists a constant $C > 0$ such that:

$$\int_\Omega v^2 dx \leq C \int_\Omega |\nabla v|^2 dx$$

The Poincaré inequality (4.9) is not true for functions of $H^1(\Omega)$ but remains true if $v = 0$ on a part of the boundary only.

Values at boundary: trace concept *To go further...*

For $d \geq 2$, the functions v of $H^1(\Omega)$, which are measurable functions "only", are a-priori not continuous. Then it is not clear whether the boundary value of v has any sense (recall that $\partial\Omega$ is a set of measure zero...). However, it is possible to define the boundary value of v from the so-called *trace* as follows.

Theorem 8. ***(Trace theorem)***

The trace mapping $\gamma : v \to v|_{\partial\Omega}$ defined from $H^1(\Omega) \bigcap C(\Omega)$ into $L^2(\partial\Omega) \bigcap C(\partial\Omega)$ can be extended by continuity to a continuous linear mapping of $H^1(\Omega)$ into $L^2(\partial\Omega)$, again called γ.

As a result, it exists a constant $C > 0$ such that:

$$\|v\|_{L^2(\partial\Omega)} \leq C \|v\|_{H^1(\Omega)} \tag{1.2.7}$$

Green formula in Sobolev spaces It can be proved that the Green formula (1.2.1) and the integration by parts (1.2.2)(1.2.3) remain valid for functions in $H^1(\Omega)$ and $H^2(\Omega)$ respectively:

$$\forall (u,v) \in \left(H^1(\Omega)\right)^2, \qquad \int_\Omega u(x)\partial_i v(x)\ dx = -\int_\Omega v(x)\partial_i u(x)\ dx + \int_{\partial\Omega} u(x)v(x)n_i(x)\ ds, \quad i=1,..,d \tag{1.2.8}$$

The proof is based on a density argument and the trace theorem, see e.g. [**?**] for more details.

A nonlinear problem ? Spaces $W^{m,p}(\Omega)$ To go further....

As already mentioned, the Sobolev spaces $H^m(\Omega)$ are the natural spaces to solve the variational formulations of *linear elliptic BVP;* it is the minimal functional space to get a *finite energy* too.

However for nonlinear problems these spaces are not the adequate ones: one generally needs to consider the spaces $W^{m,p}(\Omega)$ with p real, $1 \le p \le +\infty$ and m integer, $m \ge 1$.

These Sobolev spaces $W^{m,p}(\Omega)$ are constructed on the Banach space $L^p(\Omega)$ as follows:

$$W^{m,p}(\Omega) = \{v \in L^p(\Omega) \text{ s.t. } \forall \alpha \text{ with } |\alpha| \le m,\ \partial^\alpha v \in L^p(\Omega)\}$$

where the partial derivatives $\partial^\alpha v$ are taken in the weak sense.

Equipped with the norm $\|u\|_{m,p} = \left(\sum_{|\alpha|<m} \|\partial^\alpha u\|^p\right)^{1/p}$, these spaces are Banach spaces but *not* Hilbert spaces anymore ... (excepted if $p=2$ of course).

On the distributions To go further....

The concept of weak solutions and weak derivatives above are naturally derived if using the distributions theory. Distributions are object that generalize the notion of functions. The basic idea is to re-interpret the functions as linear functionals acting on the test functions space. This theory gives in particular a meaningful sense to the Dirac "function" δ.

Historically, the distributions theory derived by L. Schwarz (1915-2002) has been established after the Sobolev spaces (S. Sobolev, 1908-1989) and the variational approach for solving PDE.

The theory of distributions is the complete framework to write such analyses. However, the distribution theory is out of the scope of the present course; we refer to the present reference book [**?**] and references therein for more details.

1.2.4 Weak formulation in the Sobolev spaces and Lax-Milgram's theory

As already mentioned, the weak form of a BVP is more general since considering a greater set of potential solutions, and the natural spaces for analysing the *linear* elliptuc BVP are the Sobolev spaces $H^m(\Omega)$, $m \ge 1$. Indeed we prove by using the Lax-Milgram theory that the linear BVP problems are well-posed (that is they have a unique solution and it depends continuously on the data). To do so we follow the variational approach presented previously but in the Sobolev spaces.

1.2.4.1 The existence - uniqueness theorem

Let us denote by V a (real) Hilbert space with scalar product $<\cdot,\cdot>$ and norm $\|\cdot\|$.

In the present context V will be a Sobolev space $H^m(\Omega)$.

Then we consider the following weak formulation:

$$\text{Find } u \in V \text{ such that: } a(u,v) = b(v)\ \forall v \in V \tag{1.2.9}$$

with $a(\cdot,\cdot)$ a bilinear form on V and $b(\cdot)$ a linear on V (i.e. a linear application defined from V into $\mathbf{R}$).

Recall that (1.2.4) is the weak formulation version in the regular / classical space X.

Recalls: continuity and ellipticity of the forms

- The linear form $b(\cdot)$ is continuous from V into $\mathbf{R}$ if it exists $C > 0$ such that:

$$|L(v)| \le C\|v\|\ \forall v \in V$$

Lemma 4.2.4 (weak differentiation)	$u \in L^2(\Omega)$ is differentiable in the weak sense if, $\forall i$, $\left\lvert \int_\Omega u \frac{\partial \phi}{\partial x_i}\, dx \right\rvert \leq C\|\phi\|_{L^2(\Omega)} \quad \forall \phi \in C_c^\infty(\Omega)$
Proposition 4.3.2	$H^1(\Omega)$ is a Hilbert space for the scalar product $\langle u, v\rangle = \int_\Omega (\nabla u \cdot \nabla v + uv)\, dx$
Theorem 4.3.5 (density theorem)	$C_c^\infty(\overline{\Omega})$ is dense in $H^1(\Omega)$
Proposition 4.3.10 (Poincaré inequality)	$\forall u \in H_0^1(\Omega)$ (Ω bounded) $\|u\|_{L^2(\Omega)} \leq C\|\nabla u\|_{L^2(\Omega)}$
Theorem 4.3.13 (trace theorem)	$u \to u_{\lvert \partial\Omega}$ is a continuous mapping of $H^1(\Omega)$ into $L^2(\partial\Omega)$
Theorem 4.3.15 (Green's formula)	$\forall u, v \in H^1(\Omega)$ $\int_\Omega u \frac{\partial v}{\partial x_i}\, dx = -\int_\Omega v \frac{\partial u}{\partial x_i}\, dx + \int_{\partial\Omega} uv\, n_i\, ds$
Corollary 4.3.16 (characterization of $H_0^1(\Omega)$)	$H_0^1(\Omega)$ is the subspace of functions of $H^1(\Omega)$ which are zero on $\partial\Omega$
Theorem 4.3.21 (Rellich theorem)	The injection of $H^1(\Omega)$ in $L^2(\Omega)$ is compact (Ω bounded and regular)
Theorem 4.3.30 (Green's formula)	$\forall u \in H^2(\Omega), v \in H^1(\Omega)$ $\int_\Omega v\Delta u\, dx = -\int_\Omega \nabla u \cdot \nabla v\, dx + \int_{\partial\Omega} \frac{\partial u}{\partial n} v\, ds$

Table 4.1. Principal results on Sobolev spaces which must be known

Figure 1.2.3: To go further: results related to the Sobolev space $H^1(\Omega)$. Extracted from [?].

- The bilinear form $a(\cdot,\cdot)$ is continuous from $V \times V$ into $\mathbf{R}$ if it exists $c > 0$ such that:

$$|a(u,v)| \leq c \, \|u\|\|v\| \; \forall u, v \in V \tag{1.2.10}$$

- The form $a(\cdot,\cdot)$ is elliptic on V if it exists $\alpha > 0$ such that:

$$a(u,u) \geq \alpha \|u\|^2 \; \forall u \in V \tag{1.2.11}$$

The continuity properties are usually satisfied and easy to verify.

On the contrary the ellipticity is the crucial point and the key property of the existence and uniqueness result stated below.

Theorem 9. ***(Lax–Milgram)*** *Let V be real Hilbert space.*
If $a(\cdot,\cdot)$ is continuous coercive (elliptic) bilinear on V , if $b(\cdot)$ is continuous linear on V , then the variational formulation

$$a(u,v) = b(v) \; \forall v \in V \tag{1.2.12}$$

has a unique solution u in V.
Furthermore this solution depends continuously on the right hand side data (the source terms in b).

The proof can be found in [**?**].

P. Lax (1926 -), &A. N. Milgram (1912 - 1961).

This theorem (extended as the Lions-Lax-Milgram's theorem) is the basis to write the analysis of linear coercitive BVP.

However let us point out that generally the non linear BVP cannot be studied in the Sobolev spaces H^m (which are Hilbert spaces) but in the spaces $W^{m,p}$ for example (which are Banach spaces "only"), then the basis analysis theorems are not the present Lax-Milgram one.

In the case of non homogeneous Dirichlet boundary conditions, an extended version will have to be considered, see next section.

1.2.4.2 Energy expression, minimum of energy in the symmetric case

If we set $v = u$ in the weak form (1.2.9) of the BVP, we obtain the energy expression of the system.

The Left Hand Side (LHS) $a(u,u)$ is the stored energy in Ω while the RHS $b(u)$ represents the external force / source energy.

In the previous example, this would give:

$$a(u,u) = \|\nabla u\|^2_{L^2} \text{ and } b(u) = \int_\Omega f u \, dx - \int_{\Gamma_n} \varphi u \, ds \tag{1.2.13}$$

Moreover *if the bilinear form $a(\cdot,\cdot)$ is symmetric*, that is $a(u,v) = a(v,u) \; \forall (u,v)$, the (unique) solution of (1.2.12) minimizes the energy of the energy modeled by ((1.2.12).

Indeed, let us define the functional $J : V \to \mathbf{R}$ by:

$$J(v) = \frac{1}{2} a(v,v) - b(v) \tag{1.2.14}$$

Then u is solution of (1.2.12) if and only if u satisfies: $J'(u) \cdot v = 0 \; \forall v \in V$.

Indeed this equality is the Euler equation of the following minimization problem:

$$\min_{v \in V} \; J(v) \tag{1.2.15}$$

Moreover, it is easy to verify that $a(\cdot,\cdot)$ coercive in V implies that $J(.)$ is strictly convex.

Therefore this optimization problem (1.2.15) admits an unique solution u, which is the unique solution of (1.2.9).

In others words, at equilibrium the functional $J(.)$ is minimal: $J(.)$ defined by (1.2.13) is the energy functional.

It can be noticed that the Sobolev spaces $H^m(\Omega)$ are the minimal space to get a *finite energy:* it is the natural spaces to define and analyse these linear models.

1.3 The Laplace-Poisson equation: mathematical analysis

In this section, fundamental ingredients to analysis the Laplace-Poisson equation (that is the typical linear scalar elliptic model) are introduced.

1.3.1 The Dirichlet boundary conditions case

1.3.1.1 The model

Let us consider the following linear second order elliptic with variable coefficients:

$$\begin{cases} -\mathrm{div}(\lambda(x)\nabla u(x)) & = f(x) \text{ in } \Omega \\ u(x) & = 0 \text{ on } \partial\Omega \end{cases} \tag{1.3.1}$$

with $f \in L^2(\Omega)$ and $\lambda(x)$ given.

If $\lambda(.)$ is differentiable, we have: $\mathrm{div}(\lambda\nabla u) = \lambda\Delta u + \nabla\lambda \cdot \nabla u$.

Obviously if $\lambda(x) \equiv 1$, the Laplace operator is recovered.

A great interest of the variational approach is to consider non regular data (in the sense C^k) therefore "non regular" solutions.

In the present model, λ is often not differentiable and can even be not continuous e.g. the diffusion coefficients for two different materials, see the transmission problem in next section.

Therefore it is important to consider the model operator in the "div form", next in the weak form, since it contains more information than in the developed form.

Below the Lax-Milgram theorem enables to show the well-posedness of the BVP (1.3.1). To do so, the minimal required regularity of λ is the following.

Assumption. *The conductivity / diffusivity coefficient $\lambda(x)$ is a measurable function satisfying:*

$$0 < \lambda^- \leq \lambda(x) \leq \lambda^+ \ \textit{a.e. in } \Omega \tag{1.3.2}$$

1.3.1.2 From the classical to the variational form

Let us assume the existence and regularity of the solution u so that all the calculations below are valid (it is somehow *formal* calculations).

As already done in the classical space context X, we multiply the equation (1.3.1) by a test function v, then an integration by part gives:

$$\int_\Omega \lambda(x)\ \nabla u(x)\cdot\nabla v(x)dx - \int_{\partial\Omega} \lambda(x)\ \nabla u \cdot n v(x)\ ds = \int_\Omega f(x)v(x)dx$$

Since we have the homogeneous Dirichlet condition $u = 0$ on $\partial\Omega$, we choose V such that: $v = 0$ on $\partial\Omega \ \ \forall v \in V$.

Hence the equation:

$$\int_\Omega \lambda(x)\nabla u(x)\cdot\nabla v(x)dx = \int_\Omega f(x)v(x)dx$$

Next it becomes clear that the minimal regularity of the terms are the following : $(\nabla u, \nabla v, v)$ belong to $L^2(\Omega)$ with the data $\lambda \in L^\infty(\Omega), f \in L^2(\Omega)$.

Consequently, a good choice of functional space is the following Hilbert space:

$$V = \{v,\ v \in H^1(\Omega),\ v = 0 \text{ on } \partial\Omega\} \tag{1.3.3}$$

Therefore the proposed variational formulation of the BVP (1.3.1):

$$\text{Find } u \in V \text{ such that: } \int_\Omega \lambda(x)\nabla u(x)\cdot\nabla v(x)dx = \int_\Omega f(x)v(x)dx \ \ \forall v \in V \tag{1.3.4}$$

with $\lambda \in L^\infty(\Omega), f \in L^2(\Omega)$ given.

1.3.1.3 Well-posedness of the model

Definition 10. The model is well-posed in the sense of J. Hadamard (1923) if:

1. it has a unique solution;
2. the solution depends continuously on data.

The term "data" denotes here the source terms either in the volume or at the boundary i.e. f and φ.

A few remarks on this definition.

Mathematically, the existence of a solution can be enforced by enlarging the solution space.

If a problem has more than one solution, either it is physical or some information is missing in the model. In this case, additional properties (e.g. a sign condition) may be imposed in the model.

The second condition above is a *stability condition*. Indeed, data of a problem are always perturbed by "noise", uncertainties, and if the solution is not continuous with respect to the data then it may be "unstable" (therefore generally unphysical).

In the present example, the existence - uniqueness of the solution derives directly from the Lax-Milgram theorem.

Let us define: $a(u,v) = \int_\Omega \lambda(x)\nabla u(x)\cdot\nabla v(x)dx$ and $l(v) = \int_\Omega f(x)v(x)dx$.

Let us consider V defined by (1.3.3).

Then it is easy to verify that $a(u,v)$ is bilinear continuous from $V \times V$ into $\mathbf{R}$, and $l(v)$ is linear continuous from V into $\mathbf{R}$.

Moreover, from the Poincaré inequality it follows that $a(v,v)$ is coercive (elliptic) in V:

$$a(u,u) \geq \lambda^- \|\nabla v\|^2_{L^2} \geq \alpha \|v\|^2 \ \forall v \in V$$

with $\alpha > 0$.

Then in vertu of Lax-Milgram theorem, it exists a unique solution $u \in V$ of the variational formulation (1.3.4).

1.3.1.4 Equivalence between the variational (weak) form and the classical one

Finally let us demonstrate that if u is the (unique) solution of the weak formulation (1.3.4) then it is solution of the BVP (1.3.1) but *in a weak sense.*

If assuming that $u \in V \cap H^2(\Omega)$ (and not only in V) then the proof is the same as in the previous regular context (with (u,v) in $X \times X$).

To go further. If we do not assume this extra regularity on the solution u, then the proof becomes harder since we can no longer apply the Green's formula $-\int_\Omega \mathrm{div}(\lambda\nabla u)\ v\ dx$ =etc. Indeed this Green's formula requires u in $H^2(\Omega)$. However, if we denote $\sigma = \lambda\nabla u$, σ is a function in $(L^2(\Omega))^d$, and:

$$|\int_\Omega \sigma\cdot\nabla v\ dx| = |\int_\Omega fvdx| \leq c\|v\|_{L^2}\ \ \forall v \in V;\ V \supset C_c^\infty(\Omega)$$

This estimation proves that σ has a divergence in the weak sense, see [**?**] p83 and p112 for more details.

This weak divergence satisfies:

$$\int_\Omega \sigma\cdot\nabla v\ dx = -\int_\Omega \mathrm{div}\sigma v\ dx\ \ \forall v \in C_c^\infty(\Omega)$$

Then from the weak formulation it follows:

$$-\int_\Omega \mathrm{div}\sigma\ v\ dx = \int_\Omega fv\ dx\ \ \forall v \in V;\ V \supset C_c^\infty(\Omega)$$

Hence: $(\mathrm{div}\sigma + f) = 0$ almost everywhere in Ω, and $\mathrm{div}\sigma = -f$ in $L^2(\Omega)$.

Therefore the equation in Ω of the BVP (1.3.1) is recovered in the weak sense.

Next, the boundary conditions are recovered like in the regular case.

Finally we have proved

Theorem 11. *Let $\lambda \in L^\infty(\Omega), f \in L^2(\Omega)$ be given and let V be defined by (1.3.3).*
It exists a unique solution $u \in V$ of the variational formulation (1.3.4).
Furthermore this solution satisfies the BVP (1.3.1) almost everywhere:

$$-\mathrm{div}(\lambda(x)\nabla u(x)) = f(x) \text{ a.e. } \Omega \text{ and } u(x) = 0 \text{ a.e. } \partial\Omega.$$

Remark 12. It is will pointed out later in next section ("regularity result" discussion) that the weak solution can be a strong/classical solution if the BVP data f, λ, Ω are regular enough.

In such a case, the solution of the variational form satisfies the BVP in a classical sense, that is for all x in Ω and $\partial\Omega$.

On the continuity of the "model operator" $\mathcal{M}$

A straighforward consequence of the Lax–Milgram theorem is the continuity of the (unique) solution u with respect to f.

Indeed let us define the "model operator" $\mathcal{M}$ as follows:

$$\mathcal{M} : f \in L^2(\Omega) \mapsto u \in V \tag{1.3.5}$$

with u the unique solution of (1.3.4).

Then we have

Proposition 13. *The mapping $\mathcal{M}$ is linear and continuous from $L^2(\Omega)$ into $H^1(\Omega)$. In particular, we have:*

$$\|u\|_{H^1} \leq C\|f\|_{L^2} \tag{1.3.6}$$

with the constant $C > 0$.

Proof. The mapping $\mathcal{M}$ is clearly linear. Let us state the continuity inequality by setting $v = u$ in the variational formulation:

$$\int_\Omega \lambda(x)|\nabla u(x)|^2 \, dx = \int_\Omega f(x)u(x)dx$$

Next, using the coercivity inequality and Cauchy-Schwartz inequality we obtain:

$$\alpha\|u\|^2_{H^1} \leq \|f\|_{L^2}\|u\|_{L^2} \leq \|f\|_{L^2}\|u\|_{H^1}$$

Hence the result. □

The inequality (1.3.6) is an energy estimation since it shows that the energy of the solution is "controlled" by the source term norm (the data/source term energy).

This estimation is an *a-priori estimation* on the solution. It can be interpreted as a *stability inequality*; this point is more detailed in the Neumann boundary condition case.

This equality can be called an "energy inequality" too since established in the energy norm $\|\cdot\|_V$

This estimation ends to demonstrate that the variational formulation / BVP / model is well-posed in the sense of Hadamard.

Exercises.

Do the exercises proposed in the separated documents.

1.3.1.5 Symmetric case: equivalence with the minimun of energy

As already mentioned one has the

Proposition 14. *Let the bilinear form be symmetric that is: $a(u,v) = a(v,u)\ \forall u, v$.*
Then the (unique) solution of the variational formulation is the (unique) minimum of the energy functional:

$$\begin{aligned} J(v) &= \frac{1}{2}a(v,v) - b(v) \\ &= \frac{1}{2}\int_\Omega \lambda(x)|\nabla v(x)|^2\, dx - \int_\Omega f(x)v(x)\, dx \qquad \forall v \in V \end{aligned}$$

In other words, to solve the energy minimisation problem $\min_{v\in V} J(v)$ is equivalent to solve the variational formulation (1.3.4).

Exercise 15. Show this assertion.

Hint. Write the necessary condition / Euler inequation, actually the Euler equation by setting $v = -v$.

Moreover, since the symmetric bilinear form $a(\cdot,\cdot)$ is coercive (also called elliptic) in V, the functional J is (strongly) convex in V. As a consequence, the necessary condition is sufficient, see e.g. [?] p303.

Recall. The convexity (elliptic functions) inequality reads: $J(\theta u + (1-\theta)v) \leq \theta J(u) + (1-\theta)J(v)\ \forall (u,v) \in V,\ \forall \theta \in [0,1]$.

Another proof of this result can be found in [?] p76.

1.3.1.6 Non homogeneous Dirichlet condition case

Let us consider the case of non homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\mathrm{div}(\lambda(x)\nabla u(x)) &= f(x) \text{ in } \Omega \\ u(x) &= u_d(x) \text{ on } \partial\Omega \end{cases} \tag{1.3.7}$$

with u_d given on $\partial\Omega$.

In this case, the weak form of the BVP (1.3.9) reads as follows:

$$\begin{cases} \text{Find } u \in V_t & \text{such that:} \\ a(u,v) = l(v) & \forall v \in V_0 \end{cases} \tag{1.3.8}$$

with the affine subspace (translated space) V_t defined by:

$$V_t = \{v,\ v \in H^1(\Omega),\ v = u_d \text{ on } \partial\Omega\}$$

Observe that the test functions belong to the vectorial subspace $V_0 = H_0^1(\Omega)$.

Mathematical explanations *To go further...*

This function u_d is the trace of a function $H^1(\Omega)$, function still denoted u_d.

To *mathematically analyse* this BVP , we set the shifted function: $\tilde{u} = (u - u_d)$.

Formally this new unknown satisfies the following BVP with homogeneous Dirichlet condition:

$$\begin{cases} -\mathrm{div}(\lambda(x)\nabla \tilde{u}(x)) &= f(x) + \mathrm{div}(\lambda(x)\nabla u_d(x)) \text{ in } \Omega \\ \tilde{u}(x) &= 0 \text{ on } \partial\Omega \end{cases} \tag{1.3.9}$$

The RHS in its weak form reads: $\tilde{l}(v) = \int_\Omega [f(x) + \mathrm{div}(\lambda(x)\nabla u_d(x))]\ v(x)\, dx$; the term $\mathrm{div}(\lambda(x)\nabla u_d(x))$ does not belong to $L^2(\Omega)$.... However it belongs to the dual space of $H_0^1(\Omega)$; it is denoted $H^{-1}(\Omega)$. These "functions" (actually distributions) are not defined almost everywhere but it can be shown that all is fine in this context....

Next, the analysis of the weak formulation is similar to those in the homogeneous case.

Finally, the weak form of the BVP (1.3.9) can be written as previously indicated.

Remark 16. The existence and uniqueness of the weak solution of the BVP (1.3.7) (with non homogeneous Dirichlet boundary conditions) can be proved using the *Stampacchia's theorem*. The latter is an extension of the Lax-Milgram theorem. It is based on the same hypothesis but the solution space has to be a closed convex set of the Hilbert space V. This is the case with $u \in V_t$, V_t the affine subspace defined above.

1.3.2 The Neumann boundary conditions case

Let us consider the case of Neumann boundary conditions:

$$\begin{cases} -\mathrm{div}(\lambda(x)\nabla u(x)) + c(x)u(x) & = f(x) \text{ in } \Omega \\ -\lambda(x)\nabla u(x)\cdot n(x) & = \varphi(x) \text{ on } \partial\Omega \end{cases} \tag{1.3.10}$$

with n the unit outward norm and the flux φ given.

The considered BVP is based on the same second order operator as before plus a 0th order term, with the parameter c given, $c > 0$. This 0-th order term prevents to have "a solution up to a constant" only (see below and Lax-Milgram theory).

1.3.2.1 The weak formulation

Following the space approach as previously, we multiply the equation (1.3.10) by a test function v and we make an integration by part assuming that the solution u and all other functions are sufficiently regular so that all the calculations are valid. This gives:

$$\int_\Omega \lambda(x)\nabla u(x)\cdot\nabla v(x)dx - \int_{\partial\Omega} \lambda\nabla u\cdot nv\ ds + \int_\Omega c\,u\,v\,dx = \int_\Omega f\,v\,dx\ \forall v(x)$$

with $(\lambda\nabla u\cdot n)(x) = \varphi(x)$.

It is clear that choosing $V = H^1(\Omega)$ is an adequate choice; also the parameter $c(x)$ can be considered in $L^\infty(\Omega)$, $\varphi(x)$ can be considered in $L^2(\partial\Omega)$.

Hence the weak formulation of the BVP reads as follows.

Find $u \in V = H^1(\Omega)$ such that:

$$\int_\Omega \lambda\,\nabla u\cdot\nabla v\,dx + \int_\Omega c\,u\,v\,dx = -\int_{\partial\Omega} \varphi\,v\,ds + \int_\Omega f\,v\,dx \quad \forall v \in V \tag{1.3.11}$$

Remark 17. Observe that it is not necessary to include the Neumann boundary condition in the solution space V, $V = H^1(\Omega)$, on contrary to Dirichlet boundary conditions. Indeed, the Neumann condition naturally appears after the integration by parts. Therefore it would be useless to take into account the Neumann boundary condition in the definition of V.

1.3.2.2 Well posedness (existence - uniqueness)

Let us denote:

$$a(u,v) = \int_\Omega \lambda(x)\nabla u(x)\cdot\nabla v(x)dx + \int_\Omega c(x)u(x)v(x)dx$$

$$l(v) = -\int_{\partial\Omega} \varphi(x)v(x)\ dx + \int_\Omega f(x)v(x)dx$$

Then all the hypotheses of the Lax-Milgram theorem are satisfied, in particular the coercivity property since $a(v,v) \geq \alpha\|v\|_{H^1}$.

Note that the Poincaré's inequality is *not* required because of the 0-th order term (with $c(x) > 0$ a.e.).

Exercise. Prove these assertions.

1.3.2.3 Equivalence with the equations of the BVP

To go further...

Let us assume that the data (λ, c, f, φ) are regular enough i.e. C^k with k large enough, which make the unique solution u in $H^2(\Omega)$, hence in particular u in $C^0(\bar{\Omega})$.

(Such regularity result can be proved see **[?]** for more details; note that a regularity assumption on the domain Ω is required).

Next, we make the "reverse" integration by part on Eqn (1.3.11).

We obtain: $\forall v \in V$,

$$\begin{array}{c}\int_\Omega -\mathrm{div}(\lambda(x)\nabla u(x))v(x)\ dx + \int_{\partial\Omega} \nabla u\cdot n\ v(x)\ ds + \int_\Omega c(x)u(x)v(x)dx \\ = -\int_{\partial\Omega}\varphi(x)v(x)\ dx + \int_\Omega f(x)v(x)\ dx\end{array}$$

Hence for $v = 0$ on $\partial\Omega$ (v still belongs to V):

$$\int_\Omega [-\mathrm{div}(\lambda(x)\nabla u(x)) + c(x)u(x) - f(x)]\ v(x)\ dx = 0\ \forall v \in V$$

Next since $-\mathrm{div}(\lambda(x)\nabla u(x)) + c(x)u(x) - f(x)]$ is a continuous function, we obtain the original equation in Ω:

$$-\mathrm{div}(\lambda(x)\nabla u(x)) + c(x)u(x) = f(x)\ \forall x \in \Omega.$$

Next for all $v \in V$, one obtains: $\int_{\partial\Omega} \nabla u\cdot n\ v(x)\ ds = -\int_{\partial\Omega}\varphi(x)v(x)\ ds$.
Therefore : $-\nabla u\cdot n = \varphi(x)$ on $\partial\Omega$.

Therefore u is solution of the classical formulation (1.3.10).

1.3.2.4 Energy estimation and stability inequality

Energy estimation The unique solution of (1.3.10) satisfies the following *a-priori inequality* which is called the *"energy estimation"* too:

$$\|u\|_{H^1} \le C\,[\,\|f\|_{L^2} + \|\varphi\|_{L^2(\partial\Omega)}] \tag{1.3.12}$$

with C a constant strictly positive, C independent of u, f and φ; C depends on Ω.

Proof (to be completed in exercise).
We set $v = u$ in the weak formulation, we use the coercivity inequality and the continuity inequality. It follows:

$$\lambda^- \|\nabla u\|^2_{L^2} \le a(u,u) = (f,u)_{L^2(\Omega)} + (\varphi,u)_{L^2(\partial\Omega)}$$

Next we use the Poincaré's inequality, the Cauchy-Schwartz inequality (and the trace application continuity).

Then it exists a constant C (constant dependent on the geometry but independent of u) such that:

$$\|u\|_{H^1} \le C\,[\,\|f\|_{L^2} + \|\varphi\|_{L^2(\partial\Omega)}] \tag{1.3.13}$$

□

Stability concept

Let us consider the same BVP but with the perturbed source terms $(f + \delta f)$ and $(\varphi + \delta\varphi)$ (instead of f and φ). The BVP equations are *linear* so the weak formulation.

If we denote by $(u+\delta u)$ the solution corresponding to the perturbed source terms above, it follows the following "stability inequality":

$$\|\delta u\|_{H^1} \le c(\|\delta f\|_{L^2} + \|\delta\varphi\|_{L^2(\partial\Omega)}) \tag{1.3.14}$$

Let us interpret this inequality. The "perturbation" on the solution u due to the perturbations $(\delta f, \delta\varphi)$ on the source terms is bounded; it varies continuously with respect to the data perturbations.

This shows the *stability of the solution* with respect to data perturbations.

Observe that c is the ratio between the continuity constant and the coercivity constant. If the coercivity constant tends to 0 then c tends to $+\infty$ and the problem becomes unstable.

Remark 18. This inequality can be employed to prove the uniqueness of the solution too. To do so it is sufficient to apply the inequality to the difference of two potentials solutions $u_k, k = 1, 2$; next applying the inequality to the solution $u = (u_1 - u_2)$...

Exercise 19. Show the stability inequality (1.3.14).

1.3.3 On the regularity of the solution

1.3.3.1 Regular data - regular solution

To go further...

Previously, we have demonstrated the equivalence between the weak formulation and the classical form of the equations, under some hypothesis of regularity of the solution. For second order elliptic linear problems, u in $H^2(\Omega)$ is enough to obtain this equivalence.

Moreover more the data are regular, more the solution is regular. Let us illustrate this feature in the case of the typical order 2 (linear) BVP (1.3.1).

Proposition 20. *Let m be a integer. Let the data be regular in the sense: the geometrical domain Ω is an open bounded set of R^d of class C^{m+2}; the source term f is in $H^m(\Omega)$; the equation coefficient λ is of class C^{m+1} .*

Then, the unique solution $u \in V = H_0^1(\Omega)$ of the BVP with Dirichlet conditions (1.3.1) belongs to $H^{m+2}(\Omega)$.

The same result holds for the unique solution $u \in V = H^1(\Omega)$ of BVP with Neumann conditions (1.3.10); with the boundary source term φ regular enough and the coefficient c in C^m.

Furthermore the "model operator" $\mathcal{M} : f \mapsto u$ is linear continuous from $H^m(\Omega)$ into $H^{m+2}(\Omega)$ and there exits a constant C such that (in the homogenous boundary condition cases):

$$\|u\|_{H^{m+2}} \leq C \|f\|_{H^m} \tag{1.3.15}$$

As a consequence, under these regularity hypothesis on data, if $m > \frac{d}{2}$ then the (weak) solution $u \in V$ is a "strong/classical/regular" solution since it belongs to $C^2(\bar{\Omega})$ (recall Lemma 6).

The typical interesting case for second order elliptic linear problems in 3d is $m = 0$. That is if the RHS is $L^2(\Omega)$, if the coefficient λ in the divergence operator is C^1-regular, then far from any potential geometrical or boundary condition singularity, the solution u of the scalar linear elliptic model (1.3.1) is in $H^2(\Omega)$. Therefore it is is continuous in $\bar{\Omega}$.

This additional regularity of the solution ($H^2(\Omega)$ vs $H^1(\Omega)$) is very useful in the sequel in particular for the error estimations between u and u_h its Finite Element approximation.

1.3.3.2 Typical singularity origins

As shown above, the variational solution of the BVP (either Dirichlet or Neumann boundary conditions) can be infinitely regular if all data are C^∞ .

But what is a non-regular (i.e. a singular) solution ? What are the potential singularity origins ?

Let us consider the previous typical second order linear elliptic BVP. It admits an unique variational solution u in $H^1(\Omega)$.

Three reasons can limit "extra" regularity on its (unique) solution u. These three potential "singularity origins" are the following:

1. Non regular equations parameters or source terms (set either in Ω or on $\partial\Omega$),
2. Mixed boundary conditions (singularity in the change area of B.C. Dirichlet-Neumann),
3. A non regular geometrical domain Ω.

In Case 1, typical examples are the following.

If the source term f belongs to $L^2(\Omega)$ "only" (it may be less regular...), the unique solution u cannot be more regular than $H^2(\Omega)$. (Actually H^2 may considered as a regular solution compared to the minimal regularity $H^1(\Omega)$).

The diffusivity coefficient λ belongs to $L^\infty(\Omega)$ then the solution u cannot be "regular" in the sense C^1 or more. In particular the main operator of the BVP cannot be developed as follows: $div(\lambda\nabla u) = -\lambda\Delta u - \nabla\lambda\cdot\nabla u$ if λ is not regular enough (λ has to be differentiable).

This last example highlights the interest of the weak form if the parameter λ is discontinuous. Concerning this point we refer to the transmission problem below.

In Case 2, in a vicinity of the boundary condition change (make a figure), the solution is more than $H^1(\Omega)$ but less than $H^2(\Omega)$ (it is $H^{3/2}(\Omega)$...); in practice it presents a local singularity, see Fig. **??**.

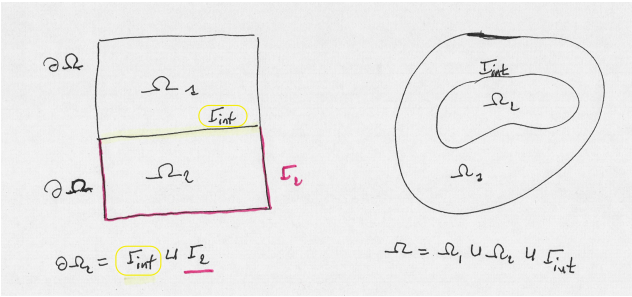


Figure 1.3.1: The domain Ω decomposed into two sub-domains.

In Case 3, any re-entring corner (worse, a crack !) generates a severe local singularity at the cone summit: the solution does not belong to $H^2(\Omega)$ in the vicinity of the cone, see , see Fig. **??** and e.g. Fig. 2.4.4.

Recall that from the modeling point of view (in physics, mechanics, biology etc), the gradient ∇u provides to the quantity flux $\nabla u \cdot n$ (e.g. heat, electric, stress field etc). Hence a singularity means that the physical flux may be not in L^∞ i.e. not bounded..., see e.g. Fig. 2.4.4.

It is important to notice that these local singularities may be the critical modeling feature (e.g. those at the corner summit). Singularities generates some difficulties for numerical methods; indeed if the modeled phenomena requires to "catch" accurately the singularity, some particular treatments have to be developed: adaptive mesh (based on a consistent a-posteriori error estimation, see later) or the introduction of additional FE basis function (xFEM, not studied here).

1.3.4 The transmission boundary condition

Let us consider the domain Ω splitted into two sub-domains Ω_k, $k = 1, 2$, see Fig.

We denote by Γ_{int} the interface between the two sub-domains. We have: $\Omega = \Omega_1 \cup \Omega_2 \cup \Gamma_{int}$.

We denote by $u_k = u|_{\Omega_k}$ (resp. λ_k and f_k) the restriction of the solution u (resp. λ and f) to Ω_k, $k = 1, 2$.

Such a geometry splitting is natural in the case of a composite material: the two subdomains are occupied by different materials with different conductivity λ_k.

Also such a splitting is required to apply Domain Decomposition Methods (DDM) (see the dedicated part of the course) enabling the computation of the solution in parallel on multiple processors (HPC) e.g. by employing the Schwarz method.

We have the following result.

Proposition 21. *The BVP:*

$$\begin{cases} -\mathrm{div}(\lambda(x)\nabla u(x)) & = f(x) \ in \ \Omega \\ u(x) & = 0 \ on \ \partial\Omega \end{cases}$$

is equivalent to the following BVP :

$$-\mathrm{div}(\lambda_k(x)\nabla u_k(x)) = f_k(x) \ in \ \Omega_k \tag{1.3.16}$$

accompanied with the original Dirichlet condition $u_k(x) = 0$ on $\partial\Omega \cap \partial\Omega_k$,
plus the "transmission condition" on the interface Γ_{int}:

$$\boxed{\begin{cases} u_1 = u_2 & on \ \Gamma_{int} \\ \lambda_1 \nabla u_1 \cdot n = \lambda_2 \nabla u_2 \cdot n & on \ \Gamma_{int} \end{cases}} \tag{1.3.17}$$

This transmission condition imposes the continuity of: a) the solution u, b) the flux, through the interface Γ_{int}.

To prove this result, one needs first the following result.

Lemma 22. *Let v be a fonction defined on Ω. It is assumed that:*
a) the restriction of v on Ω_k $v_k \equiv v|_{\Omega_k}$ belongs to $H^1(\Omega_k)$,
b) v is continuous on Γ_{int}.
Then we have: $v \in H^1(\Omega)$.

Proof. To be typed soon.

Proof of Proposition 21. This is the topic of the corresponding exercise session. Please consult the INSA Moodle page.

1.4 Appendix. The Laplace operator: basic properties

This section has to be studied by your own. (It should be translated in English next season).

The Laplace operator: an omnipresent operator in modeling

L'équation elliptique linéaire type (et aussi la plus simple) est l'équation de Poisson $-\Delta u(x) = f$, ou encore plus généralement l'équation suivante:

$$-div(\lambda \nabla u(x)) = f(x) \text{ dans } \Omega$$

où λ est donné, potentiellement dépendant de x. Rappelons que: $div(\nabla u) = \Delta u$.

Cette équation d'équilibre est omniprésente en modélisation, citons les exemples suivants.

1. En thermique, cette équation modélise la diffusion du champ de température $u(x)$ au sein d'un milieu ou matériau de géométrie Ω. Le coefficient λ est alors la diffusivité thermique du matériau et f un éventuel terme source extérieur. Le modèle doit être fermé en imposant des conditions aux limites. Pour cette équation, les conditions imposées sur le bord du domaine $\partial\Omega$, devront être une des conditions suivantes.

(a) La température est donnée: $u = u_d$. Il s'agit des conditions de Dirichlet.

(b) Le flux de température est donné: $-\lambda \nabla u \cdot n = \varphi$. Il s'agit des conditions de Neumann. Le vecteur n désignera toujours la normale sortante au bord.Dans le cas d'une paroi isolée, on aura: $\varphi = 0$, conditions de Neumann homogène. Cette condition de flux nul au bord peut également représenter une symétrie de la solution, ou encore une condition de sortie libre dans le cas d'une frontière ouverte.

(c) Une combinaision linéaire des deux conditions précédentes: $-\lambda \nabla u \cdot n = \alpha u + \varphi$. Il s'agit des conditions de Robin, aussi dites de Fourier. Ces conditions de Fourier représentent, modélisent, un flux dépendant linéairement de la valeur de u. En thermique, cela traduit un échange convectif sur une paroi. En propagation d'ondes, cela traduirait les effets d'un matériau absorbant.

(d) Les conditions aux limites sont dites mixtes lorsque l'on considère plusieurs types de C.L.; par exemple une condition de Dirichlet est imposée sur Γ_d et une condition de Neumann est imposée sur Γ_n. Attention on ne peut imposer qu'une seule CL par morceau du bord, c'est à dire que dans notre exemple on aurait nécessairement: $\Gamma_d \cup \Gamma_n = \partial\Omega$. Pour une équation elliptique une condition est requise sur l'intégralité du bord $\partial\Omega$.

2. En mécanique des structures, cette équation modélise le déplacement d'une membrane plane élastique fixée sur son bord $\partial\Omega$. u représente alors le déplacement de la membrane, déplacement perpendiculaire au plan défini par Ω, voir la figure précédente Fig. [fig:poisson].

3. En électrostatique, u représente le potentiel électrique associé à la distribution de charges f, voir la figure ci-dessous. Sur ces deux exemples, vous noterez notamment l'aspect régulier de la solution.

Qualitative properties

Principe du maximum

Considérons le problème dit de Dirichlet suivant:

$$\Delta u(x) = 0 \text{ dans } \Omega \text{ avec } u(x) = g(x) \text{ sur } \partial\Omega$$

Toute fonction solution u(x) est par définition une fonction harmonique (son laplacien est nul). On peut alors montrer que u atteint ses extrema (minima et maxima) sur le bord de $\partial\Omega$, et non à l'intérieur de Ω.

Cette propriété s'appelle le principe du maximum. Notons bien que le terme source f est nul. Par ailleurs, il est facile de démontrer que cette propriété du maximum implique l'unicité de la solution u.

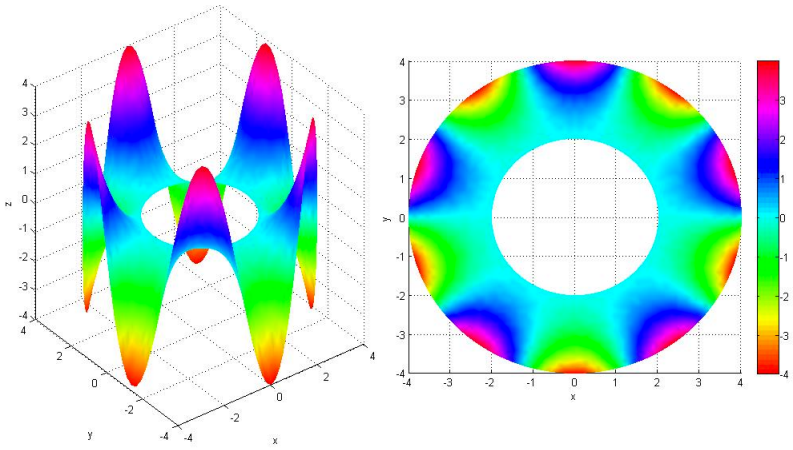


Figure 1.4.1: Solution of the Poisson equation: u is the electric potential in the ring; $u = 0$ on the inner boundary and $u = Rsin(5\theta)$ on the outer boundary. Image source: Wikipédia.

Propriété de la moyenne

La propriété de la moyenne s'énonce ainsi: la valeur de u, fonction harmonique, est en tout point égale à la moyenne de ses valeurs alentours.

Autrement formulé, on a:

$$u(x) = \frac{1}{|B_{1,n}|r^n} \int_{B_n(x,r)} u(x)dx$$

où$|B_{1,n}|$ est la mesure de la boule unité de $\mathbf{R}^n$.

Dans les exemples donnés précédemment, cette propriété se traduirait ainsi:

la valeur à l'équilibre de la quantité modélisée (température, déplacement, potentiel´electrique) est égale à la moyenne de ses valeurs environnantes.

(Rappelons que cela reste vrai en l'absence de terme source extérieur f).

Part II

Finite Element Methods

Chapter 2

Finite Element Methods: Fundamentals

Finite Element Methods (FEM) are the numerical methods of choice to solve elliptic models (e.g. those based on the Laplace equation or on the advection-diffusion equation) and parabolic models (e.g. the heat equation). They can be used for hyperbolic models too (e.g. the transport equation) by introducing stabilizing terms (e.g. artificial diffusion).

The principle of FEM directly derives from the variational approach studied in the previous chapter.

The basic principle consists is to replace the Hilbert space V of the -continuous- weak solutions by a subspace V_h of finite dimension, with V_h approximating V.

The obtained *discrete weak formulation posed in* V_h is equivalent to *an algebraic system*.

This algebraic system is linear if the original PDE is linear. The corresponding matrix is called the *stiffness matrix*.

Origins of FEM from [**?**] : "Historically, the first premises of the finite element method have been proposed by the mathematician R. Courant (without using this name) in the 1940s but it was mechanical engineers who have developed, popularized, and proved the efficiency of this method in the 1950s and 1960s (as well as giving it its actual name). After these first practical successes, mathematicians have then considerably developed the theoretical foundations of the method and proposed significant improvements".

In this chapter you will learn *how to*:

- derive a FE scheme,
- implement a FE code kernel (algorithm of assembly),
- validate a computational FE code.

On the utility of FE softwares, by Comsol company (Comsol Multiphyics software): "What Does Finite Element Analysis Software Bring ? The purpose of finite element analysis (FEA) software is to reduce the number of prototypes and experiments that have to be run when designing, optimizing, or controlling a device or process. This does not necessarily mean that companies and research institutes save money by adopting FEA. They do, however, get more

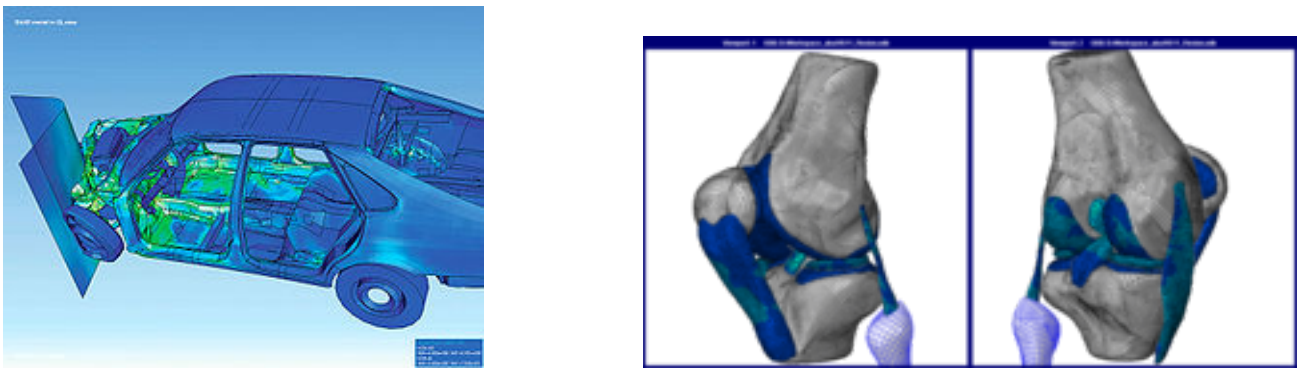


Figure 2.0.1: Examples of FE computations - simulations. (L) A car crash test. (R) A human knee joint. Images source: Wikipedia.

development for their dollars, which may result in gaining a competitive edge against the competition. For this reason, it may be reasonable to increase Research and Development resources for FEA. Once a FEA model is established and has been found useful in predicting real-life properties, it may generate the understanding and intuition to significantly improve a design and operation of a device or process. At this stage, optimization methods and automatic control may provide the last degree of improvements that can be difficult to obtain with intuition only. Most modern FEA software features methods for describing automatic control and incorporating such descriptions in mathematical and numerical models. Optimization methods are usually included in the solution process.

The introduction of high-fidelity models have also contributed to an accelerated understanding. This has sparked new ideas and completely new designs and operation schemes that would have been hidden or otherwise impossible without modeling. Therefore, FEA is an integral tool for R&D departments in companies and institutions that operate in highly competitive markets. Over time, the use of FEA software has expanded from larger companies and institutions that support educating engineers to smaller companies in many different industries and institutions with a wide variety of disciplines." Text extracted from the Comsol webpage.

This chapter follows in great part the presentation in the excellent book [**?**], see also e.g. [**?**].

Contents

2.1 Basic principles

2.1.1 Internal approximation and discrete weak formulation

Let us consider the following toy Boundary Value Problem (BVP) based on a linear elliptic PDE :

$$\begin{cases} -div(\lambda(x)\nabla u(x)) + c(x)\ u(x) & = f(x) \text{ in } \Omega \\ u(x) & = 0 \ \text{ on } \Gamma_d \\ -\lambda\, \partial_n u(x) & = \varphi(x) \ \text{ on } \Gamma_n \end{cases} \tag{2.1.1}$$

with $\partial\Omega = \Gamma_n \cup \Gamma_d$.

We assume that this BVP (2.1.1) satisfies the Lax-Milgram theory assumptions: it is assumed that $\lambda, c \in L^\infty(\Omega),\ \lambda, c > 0$ a.e.

Then it admits an unique (weak) solution u in $V_0 = \{v,\ v \in H^1(\Omega),\ v = 0 \text{ on } \Gamma_d\}$; this solution u satisfying the following weak formulation:

$$\int_\Omega \lambda(x)\ \nabla u(x)\cdot\nabla v(x)\quad dx + \int_\Omega c(x)u(x)v(x)\,dx = \int_{\Gamma_n} \varphi(x)\ v(x)\ ds + \int_\Omega f(x)v(x)\,dx \quad \forall v \in V_0 \tag{2.1.2}$$

The basic principle of FEM to compute an approximation of the exact solution u is the following.

The weak formulation (2.1.2) is not solved in the infinite dimension Hilbert space V_0 but *in a finite dimension space* V_{0h}; with V_{0h} approximating V_0 in a sense to be clarified.

The FEM consists to find $u_h \in V_{0h}$ satisfying the following (discrete) weak formulation:

$$\int_\Omega \lambda(x)\ \nabla u_h(x)\cdot\nabla v_h(x)\quad dx + \int_\Omega c(x)u_h(x)v_h(x)\,dx = \int_{\Gamma_n} \varphi(x)\ v_h(x)\ ds + \int_\Omega f(x)v_h(x)\,dx \quad \forall v_h \in V_{0h} \tag{2.1.3}$$

In terms of writing only, deriving the "discrete" weak formulation (2.1.2) from the "continuous" one (2.1.3) simply consists to add subscripts $_h$ on the function space(s), the function test and the unknown.

Of course this basic principle identically applies to the general weak formulation in a Hilbert space V:

$$\begin{cases} \text{Find } u \in V \text{ such that:} \\ a(u,v) = b(v) \qquad \forall v \in V \end{cases} \tag{2.1.4}$$

where the bilinear form $a(.,.)$ and the linear form $l(.)$ satisfy the conditions of the Lax-Milgram theory.

In this case, the corresponding discrete weak formulation reads:

$$\boxed{\begin{cases} \text{Find } u_h \in V_h \text{ such that:} \\ a(u_h,v_h) = b(v_h) \qquad \forall v_h \in V_h \end{cases}} \tag{2.1.5}$$

with V_h satisfying the properties of an internal approximation (see Definition 23).

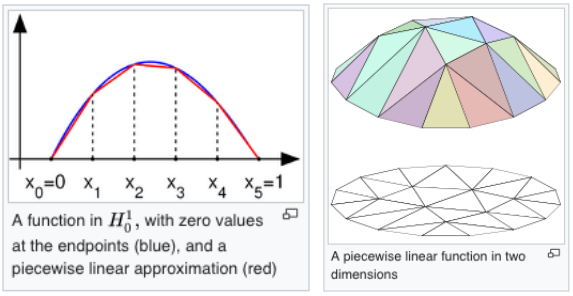


Figure 2.1.1: Approximation of functions on meshes. (L) In 1d: a (continuous) function $v \in V_0 = H_0^1(I)$ approximated by $v_h \in V_{0h}$, v_h a continuous piecewise linear function; $V_{0h} \subset V_0$. (R) Same but on a 2d triangular mesh. Images source: Wikipedia.

Definition 23. The discrete weak formulation (2.1.3) is an *internal approximation* of the (continuous) weak formulation (2.1.2) if:

1. For all h, $h > 0$ a characteristic mesh element size (e.g. in 2d it may be the maximum value of the mesh triangles edges), we have:
$$V_h \subset V \tag{2.1.6}$$
2. $\forall v \in V$, $\exists v_h \in V_h$ such that:
$$\|v - v_h\|_V \to_{h \to 0} 0 \tag{2.1.7}$$

In practice, $v_h(x)$ is an interpolation of $v(x)$ on the mesh, defined from an *interpolation operator* $r_h : V \longrightarrow V_h$, see figures 2.1.1 and 2.1.4.

Existence-uniqueness of the FEM solution u_h

Proposition 24. *Let us consider the weak formulation (2.1.4) with the assumptions of the Lax-Milgram theorem satisfied. Let V_h be an internal approximation. Then, the corresponding (discrete) weak formulation (2.1.5) in V_h is well-posed too.*

This proposition is straightforward to prove. Indeed since based on an internal approximation the assumptions of the Lax-Milgram theory are satisfied in V_h too. (V_h is equipped with the norm of V).

FEM solution & orthogonal projection If employing an internal approximation, $V_h \subset V$, it follows that:

$$a(u, v_h) = b(v_h) \; \forall v_h \in V_h, V_h \subset V \tag{2.1.8}$$

Indeed the continuous weak formulation remains satisfied with function tests v in V_h.

Next by subtracting (2.1.5) to (2.1.8) it follows:

$$a(u - u_h, v_h) = 0 \qquad \forall v_h \in V_h; \quad V_h \subset V \tag{2.1.9}$$

This equality (2.1.9) is called *the fundamental Galerkin orthogonality condition.*

If the bilinear form $a(.,.)$ is symmetric, $a(.,.)$ defines a scalar product. As a consequence (2.1.9) shows that :
If the bilinear form $a(.,.)$ is symmetric, if the FE solution $u_h(x)$ is defined from an internal approximation ($V_h \subset V$) then u_h is nothing else than the orthogonal projection of u onto V_h.
(Make a figure).

The great of majority of FE spaces V_h are internal approximation spaces of the Sobolev spaces $H^m(\Omega)$ (or $H_0^m(\Omega)$ if taking into account Dirichlet boundary conditions); a few of them are not (the latter are not studied here). When based on an internal approximation we call the method a *conforming FE method.*

2.1.2 On FE meshes

The construction of a finite dimensional space V_h is based on a *mesh of the domain* Ω. A mesh is a discrete representation of the geometry Ω; essentially it partitions Ω into simple elementary "volumes" called "elements", hence the terminology *Finite Elements*. (Note that these elementary "volumes" are generally called "cells" if employing a Finite Volume method).

These elementary "volumes" may be:

- triangles or rectangles (or even quadrilateral) in 2D;
- tetrahedra, prisms, hexahedra or parallelepipeds in 3D, see Fig. 2.1.2 and 2.1.3.

The parameter h of V_h represents a characteristic size of the elements e.g. the maximum size of the circumcircles radius of the triangles constituting the mesh.

Mesh generation is the practice of generating a polygonal or polyhedral mesh that approximates the geometric domain Ω. This known-how is not addressed in the present course. This task may be done by an adequate software e.g. by employing Gmsh software (Gnu license).

To be *admissible* a FE mesh has to satisfy a few properties. In particular, an admissible FE mesh (eg. in triangle or quadrangles) have to satisfy the geometrical properties indicated in Fig. 2.1.2(Down).

On triangle meshes. If considering a *triangulation* of the geometrical domain Ω that is in 2D a subdivision of Ω into triangles, and in higher dimension a subdivision into simplexes, tetrahedra in 3D. The triangles of a triangulation are required to meet edge-to-edge and vertex-to-vertex, see Fig. 2.1.2(Down).
An admissible triangulation $\mathcal{T}_h$ of the geometrical domain Ω has to satisfy the following criteria:

- $\Omega = \cup_{K\in\mathcal{T}_h} K$,
- The intersection between triangles is either void or equal to a single point or a complete edge,
- No element (triangle) K, $K \in \mathcal{T}_h$, can be flat.

Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation, with respect to a metric.
The metric may the Euclidian one (the Delaunay triangulation provides almost equilateral triangles only) or a metric derived from a-posteriori estimation of the FE solution; we refer to the sections "A-posteriori error estimations" and "Mesh refinement" at the end of this course.

On rectangular meshes. If the domain Ω is "rectangular" in the sense its faces are parallel to the axes, Ω can be meshed using rectangles if $n = 2$ and parallelepipeds if $n = 3$.

2.1.3 The (linear) algebraic system

Below we show that the weak FEM formulation (2.1.5) is equivalent to a system of algebraic equations. This algebraic system is linear if (and only if) the original PDE is linear (in its unknown $u(x)$).
In this case, the matrix A of the linear system is called the *matrix of rigidity* or s*tiffness matrix.*
Note that this terminology is usual even if the considered application is not related to structural mechanics. This terminology is due to the origin of the FEM.

The FE solution is a values vector at "nodes" of the mesh. It is obtained by solving ("inverting") the linear system.

Let us consider the problem (2.1.4) and its discrete version (2.1.5) in V_h.
The assumptions of the Lax-Milgram theory are supposed to be satisfied.

The definition of the FE space V_h sets the FE method.
We set $\{\varphi_i(x)\}_{i=1..NN}$ the function basis of V_h.

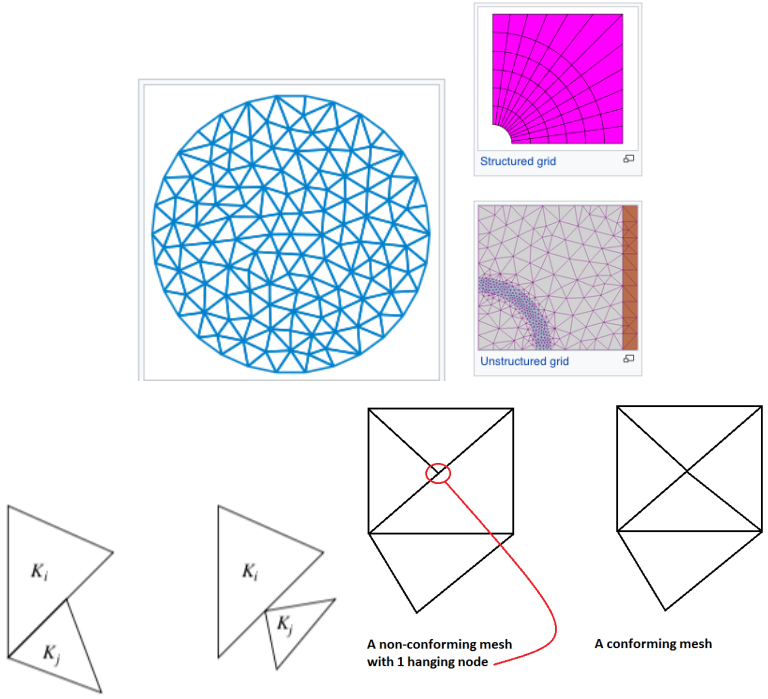


Figure 2.1.2: Mesh of a 2D domain. (Up)(L) Unstructured mesh (pseudo-uniform). (Up)(R) A structured mesh (up) and an unstructured one (containing local refinements) (down). (Images source: Wikipedia). (Down) Triangle meshes: non conform elements.

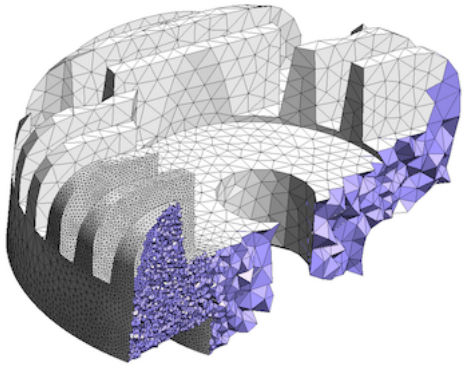


Figure 2.1.3: Tetrahedra mesh of a domain Ω obtained with Gmsh software. This mesh contains "refined" elements.

Classical FE spaces are studied in next section: the basis functions expressions are detailed.

The unique (discrete) solution $u_h(x)$ of (2.1.5) may be written in the basis of V_h as follows:

$$u_h(x) = \sum_{i=1}^{NN} u_i \, \varphi_i(x) \tag{2.1.10}$$

Where:

- u_i are the coefficients of $u_h(x)$ at the i-th node; they are called the *degrees of freedom (dof)*;
- NN is the total number of "points"; actually they are called *nodes* (hence the notation NN).

Proposition 25. Let us consider the weak formulations (2.1.4) and (2.1.5) above. We assume that the assumptions of Lax-Milgram theory are satisfied.
We consider the decomposition of $u_h(x)$ in V_h as in (2.1.10).
Then the discrete weak formulation (2.1.5) is equivalent to the following linear algebraic system:

$$A\,U_h = F \tag{2.1.11}$$

where:
U_h is the vector of degrees of freedom (dof), $U_h = (u_1, ..., u_{NN})$, $i = 1..N$; $U_h \in R^{NN}$.
$A = (a_{ij})_{i,j=1..NN}$ is the stiffness matrix defined by:

$$a_{ij} = a(\varphi_j, \varphi_i), \qquad 1 \leq i, j \leq NN \tag{2.1.12}$$

$F = (f_i)_{i=1..NN}$ is the RHS (also called source term) defined by: $f_i = b(\varphi_i(x))$, $1 \leq i \leq NN$.
Moreover since the bilinear form $a(.,.)$ is V_0-elliptic then the stiffness matrix A is positive definite.
Moreover if $a(.,.)$ is symmetric then A is symmetric too.

The FE solution is the dof vector U_h i.e. the vector of values at the mesh nodes.

Proof. *By combining (2.1.10) with (2.1.5), the discrete weak formulation is equivalent to:*

$$a\left(\sum_{i=1}^{NN} u_i \, \varphi_i(x), \varphi_j(x)\right) = l\left(\varphi_j(x)\right) \quad \forall j \in \{1, .., NN\}$$

Since the form $a(.,.)$ is bi-linear, in particular linear with respect to the unknown u (the original BVP being linear), it follows:

$$\sum_{i=1}^{NN} u_i \; a\left(\varphi_i(x), \varphi_j(x)\right) = l\left(\varphi_j(x)\right) \quad \forall j \in \{1, .., NN\}$$

Finally the equivalency with the linear system (2.1.11) follows.

The bilinear form $a(.,.)$ is supposed to V_0-elliptic, see (1.2.11), therefore: $\forall i,\ a(\varphi_i, \varphi_i) \geq \alpha \|\varphi_i\|_V^2$.
As a consequence, the stiffness matrix A is positive definite.

2.1.4 A-priori error estimation

The a-priori estimation derived in this paragraph shows that the FE error is bounded by the distance separating the continuous solution $u \in V$ to the discrete FE space V_h.

The result below is due to J. Céa; it is called Céa's lemma.

Lemma 26. *Let u be the solution of (2.1.4), let u_h be the solution of (2.1.5) and V_h be an internal approximation of V. We have:*

$$\|u-u_h\|_V \leq \tilde{c} \inf_{v_v \in V_h} \|u - v_h\|_V \tag{2.1.13}$$

with the constant $\tilde{c} = \frac{c}{\nu}$, c the continuity constant of the bilinear form $a(.,.)$, see (1.2.10), and ν the ellipticity constant, see (1.2.11).

Proof. We have:

$$a(u-u_h, u-u_h) = a(u-u_h, u-v_h) + a(u-u_h, v_h-u_h) = a(u-u_h, u-v_h)$$

since (2.1.9) holds.

Therefore for all v_h in V_h,

$$\alpha\|u-u_h\|_V^2 \leq a(u-u_h, u-u_h) = a(u-u_h, u-v_h) \leq c\|u-u_h\|_V\|u-v_h\|_V$$

It follows the inequality:

$$\|u-u_h\|_V \leq \frac{c}{\nu}\|u-v_h\|_V \quad \forall v_h \in V_h$$

Hence the result.

Corollary 27. *Let us assume that it exists :*
i) a subspace $\mathcal{V}$, $\mathcal{V} \subset V$, $\mathcal{V}$ dense in V;
ii) an interpolation operator r_h defined from $\mathcal{V}$ into V_h such that: $\forall v \in \mathcal{V}$, $\lim_{h\to 0} \|v - r_h(v)\|_V = 0$.
Then the approximation method in V_h as builded above (with V_h an internal approximation of V) converges:

$$\lim_{h\to 0} \|u-u_h\|_V = 0$$

Proof. We refer to [**?**].

In short, this proves that we have:

The FE error $\|u - u_h\|_V \leq cst \cdot$ the interpolation error $\|u - r_h(u)\|_V$

2.1.5 Building up a good FE space V_h

Considering Proposition 25, Lemma 26 and Corollary 27, the FE space V_h should be built following (at least) the two criteria below:

1. The linear system (2.1.11) is not too CPU-time consuming even for very large systems e.g. dozens of millions of dof for 3D models.
   As a consequence A must be *sparse* i.e. containing a very small percentage of non vanishing coefficients a_{ij} (2.1.12).
2. The distance between $u \in V$ and V_h tends to 0 with the mesh size h.
   This should be done by building an *interpolation operator* between V and V_h.

Defining a FE method consists to define the approximation space V_h.

2.1.5.1 On the Galerkin method

This a "to go further" paragraph.

Galerkin method denotes an approach to "discretize"-convert a continuous operator (e.g. a differential equation) to a discrete problem.

Let us assume that the Hilbert space V is separable (Sobolev spaces are). Then V admits an orthonormal (Hilbertian) basis $\{e_i\}_{i\geq 1}$, $(e_i, e_j)_V = \delta_{ij}$.

For all $v \in V$, $\exists \alpha_i$ such that $v(x) = \sum_{i\geq 1} \alpha_i e_i(x)$.

By setting $h = 1/n$, we may define V_h as the finite dimensional subspace generated by $\{e_1, ..., e_n\}$.

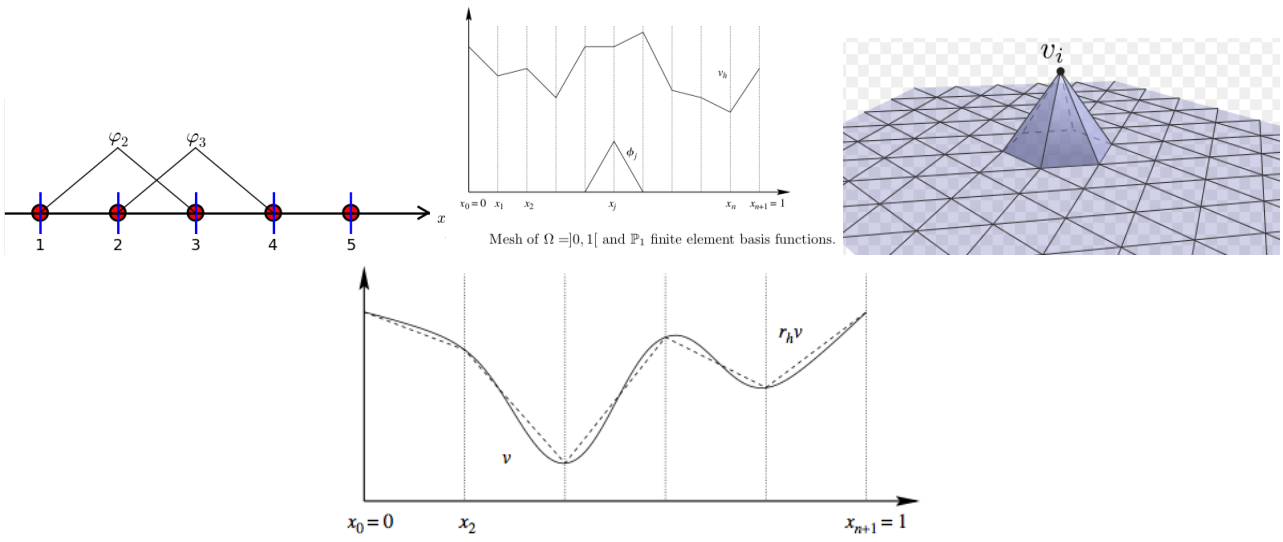


Figure 2.1.4: FE space V_h. Piecewise linear functions, globally continuous in $\bar{\Omega}$: the "hat" functions.
Up: (L&M) In 1D (R) In a 2D triangle mesh. Down: A continuous function approximated using a piecewise linear interpolation.

However with such a choice of V_h, the stiffness matrix A is a-priori full, at least not sparse. Moreover A would be *ill-conditioned* therefore providing a numerical solution very sensitive to any inherent error including rounding errors.

In brief, Galerkin's method presents a huge interest in a theoretical point of view (for analysis of non linear problems), but it is not relevant in a computational point of view. However Galerkin's method has been a fundamental step for FEM.

2.1.5.2 Required features of any FE space V_h

Let us recall the two fundamentals properties the space V_h has to satisfy:

- sparsity of A,
- easy-to-build from an interpolation operator r_h.

As a consequence, any FE space V_h should be composed of *basis functions whose support is small* i.e. supports localized in a few mesh elements only.

Indeed this feature has two crucial consequences:

- the stiffness matrix A of the linear system is sparse;
- when the mesh size h tends to 0, the (finite dimensional) space V_h is larger and larger. As a consequence V_h should approach better and better the (infinite dimension) space V.

A natural choice of basis functions are piecewise polynomials with small supports.

The simplest space V_h would be piecewise constant polynomials. However this choice would not lead to an internal approximation. Indeed in the most classical context $V = H_0^1(\Omega)$, partial derivatives of v_h would be Dirac measures and not L^2 functions...

The slightly more complex choice would be V_h defined as the space of linear functions (polynomials of degree 1), non continuous in Ω; that is discontinuous piecewise linear functions. However this choice would not lead to an internal approximation for the same reason as before.

The simplest conforming FE space The simplest FE space V_h with $V_h \subset V$ (V typically being a Sobolev space) seems to be the space of piecewise linear (affine) functions, globally continuous in $\bar{\Omega}$.

That is:

$$V_h = \{v_h, v_h \in C^0(\bar{\Omega}), \quad v_h|_K \text{ linear (affine) for all element } K \text{ of the mesh}\} \tag{2.1.14}$$

In vertu of Lemma 22, V_h defined as above is a subspace of $H^1(\Omega)$.

For V_h defined by (2.1.14), we have, see Fig. 2.1.4:

$$a(\varphi_j, \varphi_i) \neq 0 \text{ if and only if } \mathrm{Supp}(\varphi_j) \cap \mathrm{Supp}(\varphi_i) \neq \emptyset \tag{2.1.15}$$

Therefore the stiffness matrix A remains extremely sparse (a dozen of non vanishing coefficients per rows) even for very large numerical systems e.g. $NN \sim 10^6 - 10^9$.

Therefore linear algebra methods, e.g. the preconditioned Conjugate Gradient or GMRES algorithms, may apply to efficiently solve the FE linear system (2.1.11).

2.2 The P_k-Lagrange FE

The usual functional spaces to solve elliptic BVP are the spaces $W^{m,p}(\Omega)$, $m, p \geq 1$. This includes the most classical Sobolev spaces $H^1(\Omega), H^1_0(\Omega), H^2(\Omega)$. All of them are Hilbert spaces.

The basic principle of the mostly employed Finite Element Methods (FEM) are *internal approximations* as sketched in the previous section.

In the next sections we study the most classical FEM for scalar linear order 2 BVP problems, namely the P_k-Lagrange methods.

2.2.1 The P_1-Lagrange FE in 1D

In the present section, the linear case i.e. $k = 1$ is studied in 1d for a typical linear BVP.

First let us point out that meshing a 1D domain is straightforward.

Let us consider $\Omega =]0, L[$. If we mesh this domain with *an uniform* mesh of size h, we have: $h = \frac{L}{(NN-1)}$ *with* NN the total number of the mesh points ("vertices").

The mesh vertices x_i satisfy:

$$x_i = (i-1)h, \quad 1 \leq i \leq NN \tag{2.2.1}$$

Exercise

Let us consider the following model (2.1.1) with Neumann bc:

$$\begin{cases} -(\lambda(x)u'(x))' + c(x)\ u(x) & = f(x) \text{ in }]0, L[\\ -\lambda(0)\, u'(0) & = 0 \\ -\lambda(L)\, u'(L) & = \phi \end{cases} \tag{2.2.2}$$

with the following assumptions on its data: $\inf_x c(x) = c^- > 0$ and $\inf_x \lambda(x) = \lambda^- > 0$.

Q1) Show that under these assumptions and in vertu of the Lax-Milgram theorem, this BVP admits an unique weak solution u in V with $V = H^1(0, L)$
*

The P1 -Lagrange basis function in 1d The $\mathbf{P}_1$ FE scheme is defined from V_h builded up as the space of functions which are globally continuous and linear (actually affine) on each element.

In the 1D case this reads:

$$V_h = \{v_h, v_h \in C^0([0, L]), \quad v_h|_K \in \mathbf{P}_1 \quad \forall \mathrm{K} = [x_i, x_{i+1}], \quad \mathrm{i}=1,..,\mathrm{NN}\} \tag{2.2.3}$$

Q2) Show that V_h is a sub-space of $V = H^1(\Omega)$.
Correction hint: In the present 1d case, this result is trivial. Indeed, let us recall Lemma 6 and Lemma 22.
*

Let us define the "hat functions" as follows, see Fig. 2.1.4:

$$\varphi_i(x) = \varphi\left(\frac{x - x_i}{h}\right) \tag{2.2.4}$$

with $\varphi(.)$ the "normalized hat function" defined by: $\varphi(x) = (1 - |x|)$ for $|x| \leq 1$, $\varphi(x) = 0$ otherwise.

Q3) Let $v_h \in V_h$. Following (2.1.10), we write: $v_h(x) = \sum_{i=1}^{NN} v_i\, \varphi_i(x)$ with $\{\varphi_i(x)\}_i$ the hat functions defined above.

Show that any function $v_h \in V_h$ is uniquely defined by its values v_i at the mesh vertices x_i .
Deduce that the hat functions set $\{\varphi_i(x)\}_i$ defines a basis of V_h.

Correction. This result is proved in the 2d case in Proposition 29.
*

Q4) Show that:

$$u(x_j) = u_j \qquad \forall j, \quad 1 \leq j \leq NN \tag{2.2.5}$$

That is *the i-th degree of freedom (dof) u_j is nothing else than the value of the FE approximation at the i-th point x_i.*

Correction. For any function v_h in V_h, we write the decomposition: $v_h(x) = \sum_{i=1}^{NN} v_i \varphi_i(x)$.
One easily notice the following remarkable property of the $\mathbf{P}_1$ basis functions (the hat functions):

$$\varphi_i(x_j) = \delta_{ij} \quad \forall i, j, \quad 1 \leq i, j \leq NN \tag{2.2.6}$$

Then the results follows straighforwardly.
*

Q5) Show that the discrete weak formulation in V_h is equivalent to a linear system.
Detail the i-th equation of this system for x_i inside the domain ($i = 2..(NN-1)$) and for x_i at the boundaries ($i = 1$ and $i = NN$).

Correction. The discrete weak formulation of the BVP (2.2.2) reads as follows:

$$\text{Find } u_h \in V_h \text{ satisfying: } a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_h \tag{2.2.7}$$

With:

$$a(u_h, v_h) = \int_0^L \lambda(x)\ \ u_h'(x) v_h'(x)\ dx + \int_0^L c(x) u_h(x) v_h(x)\ dx$$

$$l(v_h) = \phi v_h(L) + \int_0^L f(x) v_h(x)\ dx$$

The discrete weak formulation is equivalent with: $a(u_h, \varphi_i) = l(\varphi_i) \quad \forall i,\ i = 1, ..NN$.

Moreover, we decompose u_h into the basis: $u_h(x) = \sum_j u_j \varphi_j(x)$.
It follows the i-th equation (those related to the i-th node of the basis):

$$\sum_{j=1}^{n} u_j \int_0^L \lambda(x)\ \ \varphi_j'(x) \cdot \varphi_i'(x) \quad dx + \sum_{j=1}^{n} u_j \int_0^L c(x) \varphi_j(x) \varphi_i(x)\ dx = \phi\ \varphi_i(L) + \int_0^L f(x) \varphi_i(x)\ dx \quad \forall i, j = 1..NN \tag{2.2.8}$$

Note that $\varphi_i(L) \neq 0$ only for $i = NN$.
Then the NN equations are equivalent to the linear system:

$$A\, U_h = F \tag{2.2.9}$$

with $U_h = (u_1, ..., u_{NN})$ the vector of degrees of freedom (dof), $A = (a_{ij})_{i,j=1..NN}$ the stiffness matrix with:

$$a_{ij} = a(\varphi_j, \varphi_i) = \int_0^L \lambda(x)\ \ \varphi_j'(x) \varphi_i'(x)\ dx + \int_0^L c(x) \varphi_j(x) \varphi_i(x)\ dx \qquad 1 \leq i, j \leq NN \tag{2.2.10}$$

and

$$F_i = \phi\ \varphi_i(L) + \int_0^1 f(x)\varphi_i(x)dx,\ 1 \leq i \leq NN$$

The basis functions φ_i have a small support, more precisely the two segments whose x_i is an extremity, see Fig. 2.1.4. Therefore the intersection of supports of φ_i and φ_j is empty excepted if x_i and x_j belong to the same segment (segments are the 1D elements).

As a consequence, the coefficients in the stiffness matrix *corresponding to the i-th equation* for internal nodes of the numerical model ($i = 2, .., NN-1$) read:

$$a_{ij} = \int_{x_{i-1}}^{x_{i+1}} \lambda(x)\ \ \varphi_j'(x)\varphi_i'(x)\ dx + \int_{x_{i-1}}^{x_{i+1}} c(x)\varphi_j(x)\varphi_i(x)\ dx \qquad \textit{for } j = \{(i-1), i, (i+1)\} \tag{2.2.11}$$

At the boundary $x = 0$, the first equation of the system ($i = 1$) simply reads*:*

$$\int_{x_1=0}^{x_2=h} \lambda(x)\ \ \varphi_j'(x)\varphi_i'(x)\ dx + \int_0^{x_2} c(x)\varphi_j(x)\varphi_i(x)\ dx\ \ =\ \ \int_0^{x_2} f(x)\varphi_i(x)dx \qquad \textit{for } j = \{1, 2\} \tag{2.2.12}$$

At the boundary $x = L$, the last equation ($i = NN$) reads:

$$\int_{x_{NN-1}}^{x_{NN}=L} \lambda(x)\ \ \varphi_j'(x)\varphi_i'(x)\ dx + \int_{x_{NN-1}}^{L} c(x)\varphi_j(x)\varphi_i(x)\ dx\ \ = \phi\ +\ \int_{x_{NN-1}}^{L} f(x)\varphi_i(x)dx \qquad \textit{for } j = \{NN-1, NN\} \tag{2.2.13}$$

*

Q6) Deduce from the previous question that the stiffness matrix A is tridiagonal (like it would be the case if using a Finite Difference method).

*

Q7) Let us consider the particular case $\lambda(x) = (\lambda_0 + x)$, λ_0 constant and $c = c_0$ constant.
Propose two methods to calculate the matrix coefficients a_{ij}.

***

To compute the integrals: quadrature formulas The exact values of the integrals in (2.2.11) and in the RHS may be difficult or even impossible to calculate, depending on the given functions $\lambda(x), c(x)$ and $f(x)$.

Moreover these calculations may be even more complex if the degree of the polynomials $\varphi_i(x)$ is higher.

To handle general cases, *numerical integration (quadrature formulas) is employed.* We set:

$$\int_\Omega F(x)\ dx \approx \sum_{k=1}^{NG} \omega_k F(x_k^G) \tag{2.2.14}$$

with ω_k the weigh coefficients and x_k^G the (Gauss) points.

The choice of (x_k^G, ω_k) determine the accuracy order of the quadrature formula.

The order of the employed quadrature Gauss formulas have to be in adequation with the degree of the integrand if polynomial.

If the integrand is not polynomial it has to be consistent with the "complexity" of the integrand that is in function of the expression of the equations parameters e.g. $(\lambda, c)(x)$.

On the CPU time consumption It is worth to point out that the most CPU time consuming part of a FE code is the resolution of the linear system $A\,U_h = F$ and the computation of the integrals...

On the FE terminology

- Points x_i are the mesh "vertices". Moreover these points "carry" a value of the unknown u_h; as a consequence they called *the node*s of the FEM too. Here the vertices set equals the nodes set.
  In general cases, the "vertices" (geometry information on the mesh) can be "nodes" or not.
- Since v_h is uniquely defined by its values at the nodes, the FE method above is called *Lagrange* FE.
  Later we will introduce FE spaces for which functions v_h will be uniquely defined by their value(s) at the node *and* their *derivative*(s). Such FE are called *Hermite* FE.

Elements of V_h cannot be regular solutions (regular in the sense C^k) Functions of V_h do not admit second derivatives; indeed they are Dirac measures therefore not regular functions. Therefore a solution of the classical form of the BVP cannot be a function of V_h. On the contrary, functions of V_h can be weak solutions (i.e. solutions of the weak formulation).

On the link between the FE method and the FD method It will be shown in the advection term section that in 1D the standard $\mathbf{P}_1$-Lagrange FEM (for the present equation) is nothing else than the standard FD scheme !
Moreover if considering in addition the linear advective term, the present FE scheme equals the centered FD scheme, see next section on the advection term.

2.2.2 The P_k-Lagrange FE in nD

From now the geometric domain Ω is a bounded open set of $\mathbf{R}^n$, $n \geq 2$. In practice $n = 2$ or 3.

2.2.2.1 Triangulation of Ω

It is assumed that Ω is a polyhedral (polygonal if $n = 2$); thus it is possible to mesh exactly the domain boundary $\partial\Omega$. For curves boundaries, we may use adequate Finite Element e.g. curved $\mathbf{P}_2$ elements, see the dedicated exercise session.

As a first step we consider here a mesh of Ω constituted by *triangles* if $n = 2$ or by *tetrahedra* if $n = 3$, see e.g. Fig. 2.1.2 and 2.1.3. (Following [?], triangles and tetrahedra may be grouped in the more general family of called *N-simplices*).

We consider an admissible triangulation $\mathcal{T}_h$ of Ω, see Section 2.1.2.

Remark 28. As already indicated in Section 2.1.2 if the domain Ω *is* "rectangular" in the sense its faces are parallel to the axes, Ω can be meshed using *rectangles* if $n = 2$ and *parallelepipeds* if $n = 3$. In this case the corresponding FE spaces V_h differ from the $\mathbf{P}_k$-Lagrange ones; they have to be adapted.

Rectangular elements lead to the so-called $\mathbf{Q}_k$*-Lagrange FE.* The $\mathbf{Q}_k$-Lagrange FE are studied as exercises (with $k = 1$ and 2).

2.2.2.2 The FE space V_h & basis functions

Lagrange type FE indicates that the degrees of freedom (dof) are *point values* of functions on the mesh.

On the contrary, the *Hermite* type FE are such dof are *point values and derivative(s) values* on the mesh.

Given an admissible triangulation $\mathcal{T}_h$ of Ω , the $\mathbf{P}_k-$Lagrange FE, $k \geq 1$, is defined by the following FE space:

$$V_h = \{v_h, v_h \in C^0(\Omega), \quad v_h|_{K_i} \in \mathbf{P}_k \quad \forall K_i \in \mathcal{T}_h\} \tag{2.2.15}$$

In practice $k = 1, 2$ and at maximum $k = 3$.

We define the subspace V_{oh} which includes the Dirichlet boundary conditions:

$$V_{0h} = \{v_h, v_h \in V_h, \quad v_h = 0 \text{ on } \Gamma_d\} \tag{2.2.16}$$

with Γ_d the part of the boundary where (homogeneous or not !) Dirichlet conditions are applied. We may have $\Gamma_d = \partial\Omega$.

Proposition 29. Let V_h be defined by (2.2.15) with $k \geq 1$. The following properties hold:

- V_h is an internal approximation of $V = H^1(\Omega)$.
- There exists a basis $\{\varphi_i(x)\}_{i=1..NN}$ of V_h such that:

$$\varphi_i(x_j) = \delta_{ij} \quad 1 \leq i, j \leq NN \tag{2.2.17}$$

- Any function v_h in V_h is uniquely defined by its values at the nodes x_i, with:

$$v_h(x) = \sum_{i=1}^{NN} v_i \ \varphi_i(x) \quad \forall x \in \Omega \tag{2.2.18}$$

Indeed for all $j \in \{1 \dots NN\}$, $v_h(x_j) = v_j$.

The set of values $\{v_1, ..., v_{NN}\}$ are called the *degree of freedom (dof)* vector, that is the values of v_h at the *nodes* x_i, $i = 1..NN$, see Fig. 2.2.1.

Proof. (* To go further *)
Let us prove these results in the case $k = 1$ (linear elements) and $n = 2$ (2D geometry).
i) Let us prove that V_h is an internal approximation of $V = H^1(\Omega)$.
Let T_k and $T_{k'}$ be two adjacent triangles with the common edge denoted $[A, B]$ (make a figure).
Let $w \in V_h$; we set: $v = w|_{T_k}, v' = w|_{T_{k'}}$. We have: $(v - v')(A) = 0$ and $(v - v')(B) = 0$.
Let us show that: $(v - v')(s) = 0, \forall s \in]A, B[$.
Let $M \in]A, B[$, $M = \lambda A + (1 - \lambda)B$ with $\lambda \in]0, 1[$.
The local functions v and v' are linear therefore:

$$(v - v')(M) = \lambda(v - v')(A) + (1 - \lambda)(v - v')(B) = 0$$

Therefore: $(v - v') = 0$ on $[A, B]$.
In vertu of Lemma 22, we have: $V_h \subset H^1(\Omega)$.

If considering the subspaces $V_0 = \{v; v \in H^1(\Omega), v|_{\Gamma_d} = 0\}$ and V_{0h}, $V_{0h} \subset V_0$, the same result holds.

ii) Next let us prove that any function v_h in V_h is uniquely defined by its values at the nodes N_i of coordinates (x_i, y_i).
Let T_k be the triangle (A_1, A_2, A_3). For $\mathbf{P}_1$-Lagrange elements, the nodes are the vertices of the element.
Then we show that the restriction of v_h to T_k is entirely determined by its values at the vertices A_l, $l = 1, 2, 3$.
We have $v_h|_{T_k} \in \mathbb{P}_1$ therefore $\exists(\alpha, \beta, \gamma)$ such that: $v_h(x) = \alpha x + \beta y + \gamma$.
Moreover:

$$v(A_l) = v_l, \ l = 1, 2, 3 \Longleftrightarrow \begin{bmatrix} \alpha & \alpha x_1 & \beta y_1 \\ \alpha & \alpha x_2 & \beta y_2 \\ \alpha & \alpha x_3 & \beta y_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{2.2.19}$$

We have:

$$det \left(\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \right) = (x_2 - y_3)(x_3 - y_2) - etc = \pm 2 \ area(T_k) \tag{2.2.20}$$

Therefore for an admissible triangulation $\mathcal{T}_h$ (no triangle is flat), the linear system above is invertible: it admits an unique solution (α, β, γ). This ends to show the result.

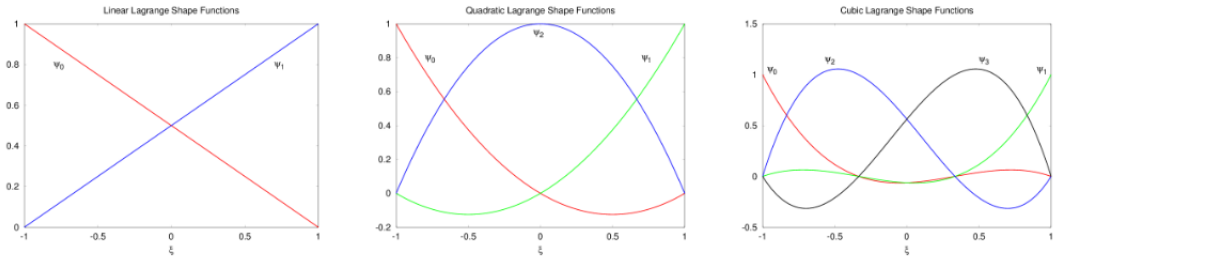
2.2.2.3 The classical higher order $\mathbf{P}_k$-Lagrange FE ($k = 2, 3$)

We illustrate on Fig. 2.5 the classical higher order $\mathbf{P}_k$-Lagrange FE ($k = 2$ and 3) in 1D.

Recall that since for $\mathbf{P}_1$-Lagrange elements the functions are affine, the *nodes* are the *vertices*.

For more details on $\mathbf{P}_k$-Lagrange FE, we refer to the exercises sessions.

It is worth to point out that a second order method ($k = 2$) is preferable to a first order one ($k = 1$) since much more accurate, see the convergence section.



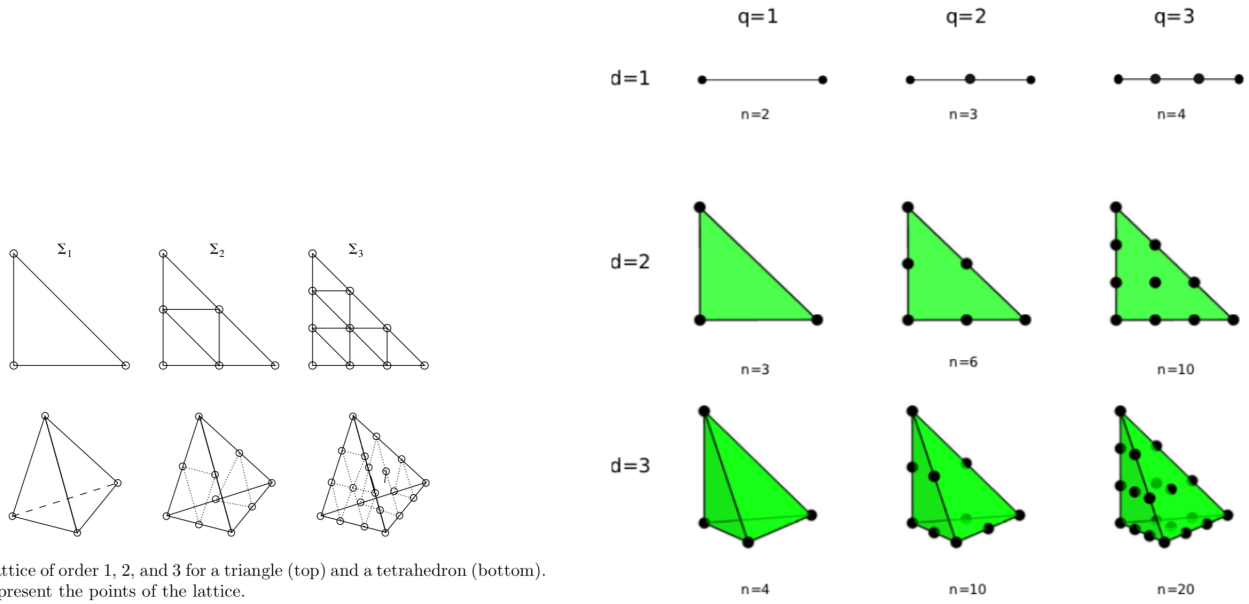


Figure 6.9. Lattice of order 1, 2, and 3 for a triangle (top) and a tetrahedron (bottom). The circles represent the points of the lattice.

Figure 2.2.1: (Up) Basis functions on the elementary segment [-1,+1] (element defined by 2 points): $\mathbf{P}_k$-Lagrange with 2, 3 and 4 nodes. (Down) $\mathbf{P}_k$-Lagrange FE in 2D and 3D *(d-simplices* elements): order $1, 2$ and 3; number of nodes n.
Images extracted from [**?**] and "Lectures on the FEM" A. Logg, K.-A. Mardal Eds.

2.3 FE code kernel: the assembly algorithm

The assembly algorithm constitutes the core of a FE code.

2.3.1 The assembly algorithm & elementary matrices

To obtain the FE solution, one has to solve the linear system $AU_h = b$ with for example:

$$a_{ij} = a(\varphi_j, \varphi_i) = \int_\Omega \lambda(x)\ \nabla\varphi_j \nabla\varphi_i(x)\ dx + \int_\Omega c(x)\varphi_j\varphi_i(x)\ dx$$
$$l_i = l(\varphi_i) = \int_\Omega f(x)\varphi_i(x)\ dx - \int_{\Gamma_n} \Phi(s)\varphi_i(s)\ ds$$

The FE solution is the dof vector U_h.

2.3.1.1 The linear system coefficients to be computed

Each integral of the discrete weak formulation is decomposed onto the elements K of the mesh, or on their edges - faces ∂K:

$$\int_\Omega \cdot\, dx = \sum_{K\in\mathcal{T}} \cdot\, dx \tag{2.3.1}$$

$$\int_\Gamma \cdot\, ds = \sum_{\partial K\in-} \cdot\, ds \tag{2.3.2}$$

Recall that each integral is non vanishing if and only if the considered nodes N_i and N_j belong to a same element, see Fig. 2.3.1.1.

Let us consider as an example the 0-th order term:

$$\int_\Omega c(x)\varphi_j\varphi_i(x)\ dx = \sum_{K\in\mathcal{T}} \int_K c(x)\varphi_j\varphi_i(x)\, dx \neq 0 \text{ if and only if } supp(\varphi_i \cap \varphi_j) \neq 0$$

Data structures required from the mesh

To make the computations above, one needs the following information from the mesh.

- Points & elements (geometry): NE, NP, $x_k = coord(IP = 1..NP, k = 1\cdots d)$
- Nodes (interpolation): $NN, ref(IN = 1..NN)$

For $\mathbb{P}_2$-Lagrange FE in 3d: 4 points (vertices) and 10 nodes per element.

2.3.1.2 The assembly algorithm

The assembly algorithm constitutes the core of a FE code. This algorithm consists to add, assembly the contribution of each element K.

The algorithm is as follows: see Algorithm 2.1.

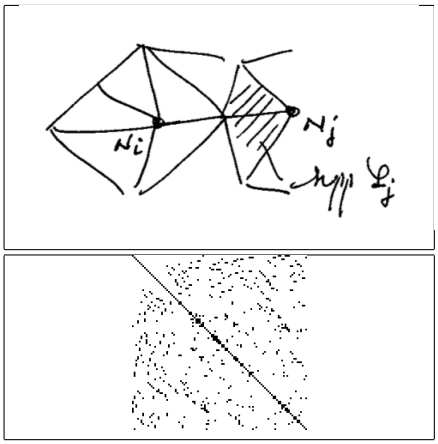
Remark 30. - A loop on the nodes instead on the elements (with the use of the connectivity table) would be much more CPU time consuming.

The complexity of the assembly algorithm above is relatively low compared to the linear solver one (linear algebra, typical complexity $\sim k\ NN^2$).
However a high-order quadrature formulae would greatly increase the assembming algorithm CPU time.

- If A is symmetrical then the loop on $JN2$ may be shorten as:

$$IN2 = modulo(IN1, NN(IE) + 1)$$

- Coding the assembly algorithm may be complicated.
Indeed e.g. for a coupled 3D model on tetrahedra, order 2, with *~10 dof per node*, one get: *~ 100 dof per element).*



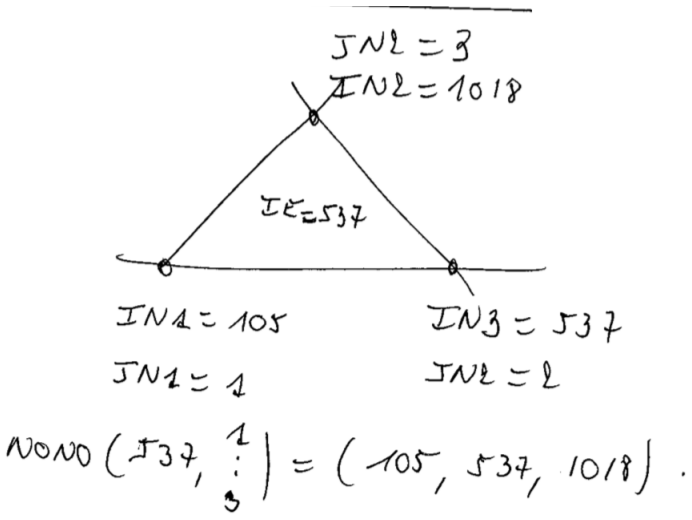


Figure 2.3.1: (L)(Up) $Supp(\varphi_i \cap \varphi_j) = \emptyset$: the FE matrices are sparse.
(L)(Down) A sparse FE matrix (of relatively small size and weird numbering algorithm).
(R) Local and global node numbering: example for a P_1 Lagrange FE.
$NONO(IE, JN)$ is the connectivity table built up by the meshing software (and provided by file).

Exercise: make a very simple mesh by hand and write the information required to describe it.

On the elementary matrices.

We may define the elementary matrices E of sizes $NNE \times NNE$, $NNE = NN(IE)$.
Then the assembly algorithm consists to add, "super-impose" all the elementary matrices.

2.3.1.3 Data structures required from the mesh (resumed)

In a FE code on need the following tabs and data; the latter are provided by the mesh generator.

- Points & elements (geometry):
  - NE, $ref(IE = 1..NE)$ if presence of few materials or areas within the domain
  - NP, $x_k = coord(IP = 1..NP, k = 1 \cdots d)$
- Nodes (interpolation):
  - $NN, NNE(IE = ..NE)$ (if presence of a mix of elements),
  - $ref(IN = 1..NN)$ for boundary conditions.
  - $CONNEC(IE, JN)$(called "$NONO(IE, JN)$" above): the connectivity table.

2.3.2 How to introduce the Dirichlet boundary conditions ?

Either a) by considering the equation as a weak constraint; or b) by imposing the node value in the stiffness matrix.

Solution a) implies to introduce a Lagrangian multiplier, see the dedicated section at the end of the course.

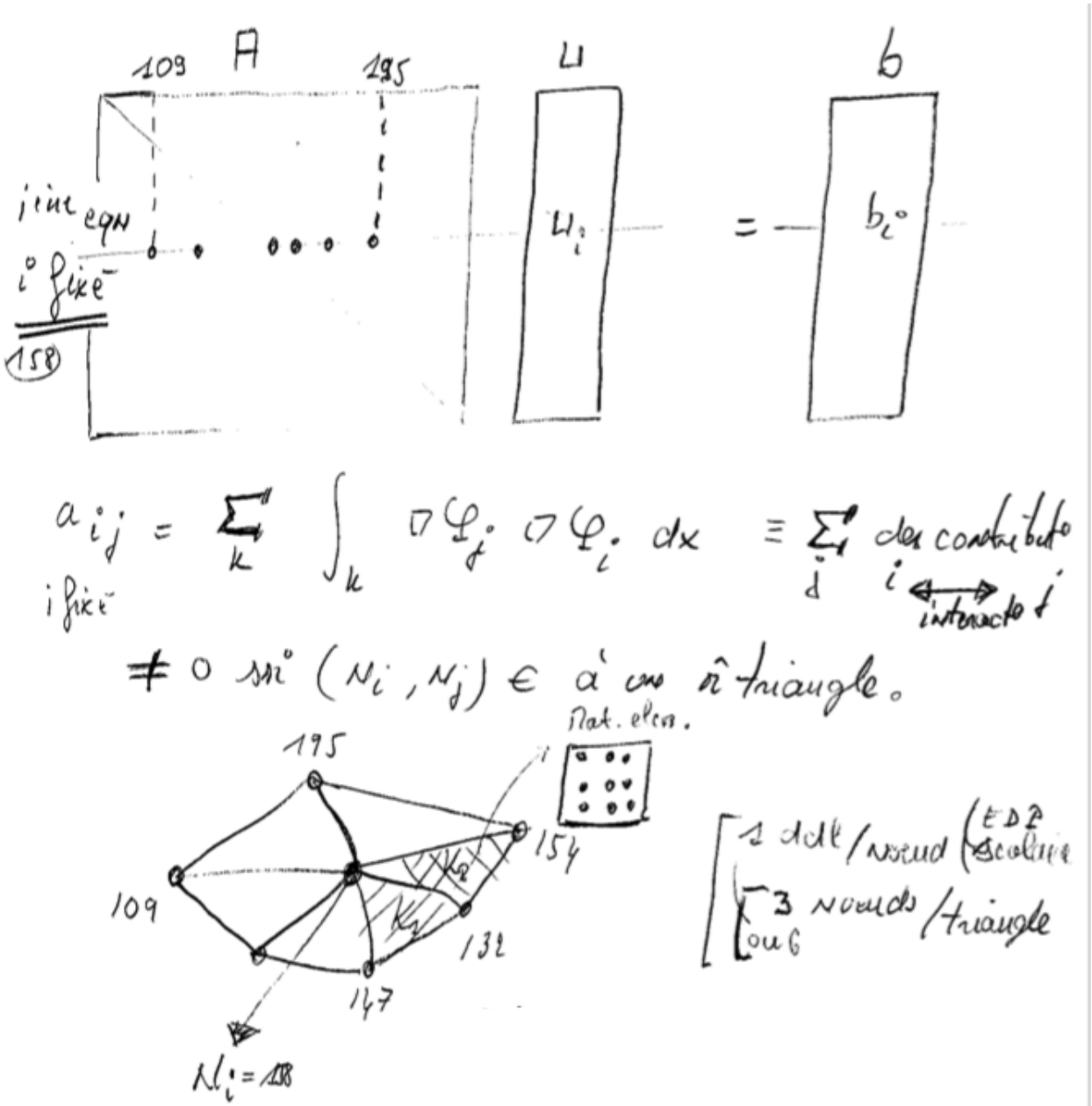


Figure 2.3.2: Assembly algorithm illustration

Algorithm 2.1 Assembly algorithm

- Loop on the elements $IE = 1 \dots NE$
  - Loop on the local node numbers $JN1 = 1 \dots NN(IE)$ *(e.g. 3 nodes for a P_1 element in 2d).*
    - $IN1 = NONO(IE, JN1)$: global number $IN1$ of node $JN1$ of element IE.
      $IN = NONO(IE, JN)$: *connectivity table provided by the mesh generator.*
    - Loop on the local node numbers $JN2 = 1 \dots NN(IE)$:
      - $IN2 = NONO(IE, JN2)$: global number of $JN2$.
      - Compute the integrals e.g. $\int\limits_{IE} c(x)\ \varphi_{IN1}(x)\varphi_{IN2}(x)\ dx$,
        in fact the integrals $\int\limits_{IE} c(x)\ \varphi_{JN1}(x)\varphi_{JN2}(x)\ dx$, see details later when using the change of variables.
        This integral may be viewed as of the coefficient of the "elementary matrix" E of size $NN(IE)^2$ *(e.g.* 3×3*)*
        $E_{JN1,JN2} = \int\limits_{IE} c(x)\ \varphi_{JN1}(x)\varphi_{JN2}(x)\ dx.$
      - Assembly the rigidity matrix A of size NN^2 *(e.g.* $10^4 \times 10^4$ *in 2d)*:
        $A_{IN1,IN2} = A_{IN1,IN2} + E_{JN1,JN2}$
    - End loop on $JN2$
    - Assembly the RHS e.g. $L_{JN1} = \int\limits_{IE} f(x)\ \varphi_{JN1}(x)\ dx$
      $b_{IN1} = b_{IN1} + L_{JN1}$
  - End loop on $JN1$
- End loop on IE

Solution b) is as follows.
The dof vector $U_h = (u_1 \cdots u_{NN})$ is augmented with the Dirichlet nodes.

Next, the corresponding (discrete) equation

$$u_i = g(x_i)\ i = 1, ..., nnd$$

is coded into the stiffness matrix as follows:

$$i^{th} \text{ equation } \rightarrow \begin{pmatrix} a_{11} & & & 0 & & & \\ & \ddots & & \vdots & & & \\ & & \ddots & 0 & & & \\ 0 & & 0 & a_{ii} & 0 & & 0 \\ & & & 0 & \ddots & & \\ & & & \vdots & & \ddots & \\ & & & 0 & & & a_{(NN+nnd)} \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_{(NN+nnd)} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_{i-1} \\ \times \\ b_{i+1} \\ \vdots \\ b_{(NN+nnd)} \end{pmatrix} \rightarrow \begin{matrix} -a_{1i}g_i \\ \\ a_{ii}g_i \\ \\ -a_{(NN+nnd)i}g_i \end{matrix} \tag{2.3.3}$$

The RHS b is modified as required:

- the i-th value b_i is set to $a_{ii}g_i$,
- For $j = 1, \cdots, (NN + nnd), j \neq i : b_j = b_j - a_{ji}g_i$.

As a consequence the i-th equation of the linear system reads as required: $a_{ii}u_i = a_{ii}g_i$ with i corresponding to a Dirichlet node.

The algorithm to impose the Dirichlet condition $u_i = g(x_i)$ with $i \in \{1, .., nnd\}$ is summarised in Algo. 2.2.

Algorithm 2.2 Imposing the Dirichlet condition $u_i = g_i$ in the linear system

- For $j = 1, \cdots, (NN + nnd), j \neq i$; j not a Dirichlet node number:
  - $a_{ji} = a_{ij} = 0$
  - $b_j = b_j - a_{ji} g_i$
- $b_i = a_{ii} g_i$

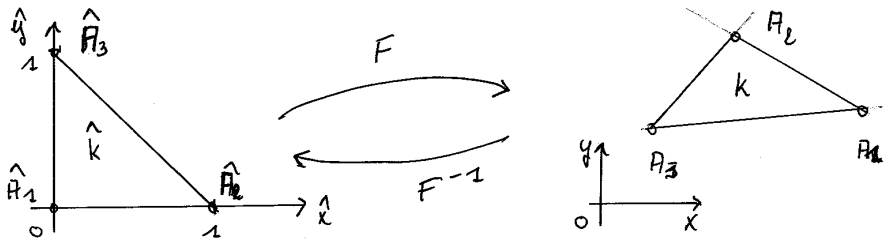


Figure 2.3.3: Change of variables onto the reference element $\hat{K}$. Here in the case of a triangle.

How to recognize a node on a given boundary ? By introducing a reference number for each node (or vertex). This may be as a tab provided by the mesh generator.

Such a tab necessary to handle the boundary conditions.

By convention, the value $ref = 0$ corresponds to nodes in the interior of Ω.

2.3.3 Change of variables onto the reference element $\hat{K}$

To compute the integral terms of the linear system coefficients it will be much more simple (and elegant) to make a change of variables to a *reference element* $\hat{K}$, see Fig. 2.3.3.1.

2.3.3.1 The geometric change of variable onto $\hat{K}$

To do so we define the change of variable (geometric transformation) F such that:

$$K = F(\hat{K}) \tag{2.3.4}$$

with $\hat{K}$ the "reference element" defined in $[0,1]^n$.

F is a change of variables if and only F is of class $C^1(\bar{\hat{K}})$, F bijective, with F^{-1} of class C^1.

In the case of a triangle $K = (A_1 A_2 A_3)$, vertex A_i has coordinates(x_i, y_i), the reference triangle $\hat{K}$ is defined as $\hat{K} = (\hat{A}_1 \hat{A}_2 \hat{A}_3)$ with, see Fig. 2.3.3.1: $\hat{A}_i = F^{-1}(A_i)$ $i = 1, \cdots, 3$.

The following property holds: n- simplexes (triangles, tetrahedra in 2d and 3d) are preserved by affine transformations i.e. $F(x)$ is affine.

For the triangle $K = (A_1 A_2 A_3)$, we have:

$$F : (\hat{x}, \hat{y}) \mapsto (x, y) = F(\hat{x}, \hat{y}) = \begin{cases} (x_2 - x_1)\hat{x} & +(x_3 - x_1)\hat{y} + x_1 \\ (y_2 - y_1)\hat{x} & +(y_3 - y_1)\hat{y} + y_1 \end{cases} \tag{2.3.5}$$

Since F is affine, $|det(DF)|$ is constant and:

$$\mathbf{x} = DF \cdot \hat{\mathbf{x}} + A_1 \tag{2.3.6}$$

with:

$$DF = \begin{pmatrix} (x_2 - x_1) & (x_3 - x_1) \\ (y_2 - y_1) & (y_3 - y_1) \end{pmatrix} \tag{2.3.7}$$

We have: $|det(DF)| = 2|K|$ i.e. twice the element area.

It is worth to notice that the elements areas $|K|$ should be computed from the mesh generator output files (i.e. out of the computational code core).

Moreover these values can be computed by the following simple formula: $|K| = \frac{1}{4}\left|\overrightarrow{A_1A_2} \wedge \overrightarrow{A_1A_3}\right|$.

Let us remark that the basis functions in $\hat{K}$ are the image by F^{-1} of the basis functions defined in K. Indeed we have:

Lemma 31. *Let $\{\varphi_i\}_{i=1..NNE}$ be the basis functions defined on K with $\varphi_i(x) \in P$, $\forall i$.*
We assume that: $\varphi_i(N_j) = \delta_{ij}$ $\forall i, j,\ 1 \leq i, j \leq NNE$.
Let F be the geometric transformation (change of variables) defined above.
Then $\{\hat{\varphi}_i = \varphi_i \circ F\}_{i=1..NNE}$ is the basis functions defined on K.

Proof. We set: $\hat{N}_i = F^{-1}(N_i)$. We have: $\hat{\varphi}_i(\hat{N}_j) = \delta_{ij}$ $\forall i, j$. This proves the result.

Change of variable for 0-th order terms

As an example, let us consider the following term: $\int_K c(x)\varphi_j(x)\varphi_i(x)\, d\mathbf{x}$. We have:

$$\int_K c(x)\varphi_j(x)\varphi_i(x)\, d\mathbf{x} = \int_{\hat{K}} c \circ F(\hat{x})\ \varphi_j \circ F(\hat{x})\varphi_i \circ F(\hat{x})\ |det((DF)|\ d\hat{\mathbf{x}} \tag{2.3.8}$$

$$= \int_{\hat{K}} \hat{c}(\hat{x})\ \hat{\varphi}_j(\hat{x})\hat{\varphi}_i(\hat{x})\ |det((DF)|\ d\hat{\mathbf{x}} \tag{2.3.9}$$

in vertu of the lemma above.

Change of variable for 1-st order terms

Let us consider the following term: $\int_K \lambda(x)\ \triangledown\varphi_j(x)\triangledown\varphi_i(x)\ dx$.

The ∇ operator is implicitly with respect to the current variable x. Thus, to make a change of variable in such integrals we need the change the gradient too.

We have

Lemma 32. *Let F be the geometric transformation (change of variables) defined above. We have:*

$$\triangledown_x\varphi_i \circ F(\hat{x}) = {}^TDF^{-1} \circ \widehat{\nabla}_{\hat{x}}\hat{\varphi}_i(\hat{x}) \qquad \forall i,\ 1 \leq i \leq NN \tag{2.3.10}$$

Proof. We have: $\varphi_i(x_1, \cdots, x_n) = \hat{\varphi}_i(\hat{x}_1, \cdots, \hat{x}_n)$. Therefore: $\forall k \in \{1, .., n\}$,

$$\partial_k\varphi_i(x_1, \cdots, x_n) = \sum_{l=1}^{n} \partial_l\hat{\varphi}_i(\hat{x}_1, \cdots, \hat{x}_n).\frac{\partial\hat{x}_l}{\partial x_k} \tag{2.3.11}$$

Recall that:

$$DF(\hat{x}, \hat{y}) = \begin{pmatrix} \frac{\partial x_1}{\partial\hat{x}_1} & \cdots & \frac{\partial x_1}{\partial\hat{x}_n} \\ \vdots & & \vdots \\ \frac{\partial x_n}{\partial\hat{x}_n} & \cdots & \frac{\partial x_n}{\partial\hat{x}_n} \end{pmatrix}\ ;\quad {}^TDF^{-1}(x, y) = \begin{pmatrix} \frac{\partial\hat{x}_1}{\partial x_1} & \cdots & \frac{\partial\hat{x}_n}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial\hat{x}_1}{\partial x_n} & \cdots & \frac{\partial\hat{x}_n}{\partial x_n} \end{pmatrix} \tag{2.3.12}$$

The end of the proof is quite straightforward (calculations to be finished in exercise).

Let us point out that these change of variables are valid even if F is not affine.
In the case F affine, of course the Jacobian $|det(DF)|$ is constant.

In the example above, we obtain:

$$\int_K \lambda(x)\ \triangledown\varphi_j(x)\triangledown\varphi_i(x)\ dx = \int_{\hat{K}} \hat{\lambda}(\hat{x})\ \left({}^TDF^{-1} \circ \widehat{\nabla}\hat{\varphi}_j\right)(\hat{x})\ \left({}^TDF^{-1} \circ \widehat{\nabla}\hat{\varphi}_i\right)(\hat{x})\ |det(DF)|\ d\hat{x} \tag{2.3.13}$$

Change of variable for boundary terms

The principle is exactly the same as above. If we consider for example the following 1d term $\int_\Gamma \Phi(s)\varphi_i(s)\ ds$, we have:

$$\int_\Gamma \Phi(s)\varphi_i(s)\ ds = \int_0^{+1} \hat{\Phi}(\hat{s})\hat{\varphi}_i(\hat{s})\ |\Gamma|\ d\hat{s} \tag{2.3.14}$$

2.3.3.2 Isoparametric FE

Isoparametric FE are FE with the geometric transformation (change of variables) F belonging to the interpolation space P.

For example for $\mathbf{P}_2$-Lagrange FE, F is not simply an affine function, it is $\mathbf{P}_2$ too.
The typical classical Iso-FE are:

- ($\mathbf{P}_2-$iso FE): they enable to generate "triangles" with curved edges. These elements are particularly interesting to properly approximate curved domain boundaries.
- ($\mathbf{Q}_1$-iso FE): they enable to generate quadrilaterals (also called quadrangles) and not simply parallelograms.

These two classical Iso-FE are studied in exercise(s).

2.3.4 On triangles & tetrahedra (n-simplexes): barycentric coordinates

In a n-simplex (triangles in $2d$, tetrahedra in $3d$), it is much more convenient to use the barycentric coordinates instead of the Cartesian coordinates.

2.3.4.1 The barycentric coordinates

Definition 33. Let K be a (non-degenerated) n-simplex with vertices $(A_i)_{1\leq i\leq n+1}$; $A_i = (a_{i,j})_{1\leq j\leq n}$.
The barycentric coordinates $(\lambda_j)_{1\leq j\leq n+1}$ of a point $x \in R^n$ are defined as follows:

$$\sum_{j=1}^{(n+1)} \lambda_j = 1, \quad x_i = \sum_{j=1}^{(n+1)} \lambda_j a_{j,i} \quad \textbf{\textit{for}}\ \ i = 1, \cdots, n. \tag{2.3.15}$$

Proposition 34. Let K be a (non-degenerated) n-simplex and $(\lambda_j)_{1\leq j\leq n+1}$ its barycentric coordinates. We have the following properties:

1. The barycentric coordinates $(\lambda_j)_{1\leq j\leq n+1}$ are affine functions of x.
2. K is characterized as:
$$K = \{x \in \mathbf{R}^n \text{ s.t. } \lambda_j(x) \geq 0,\ j = 1, \cdots, n\} \tag{2.3.16}$$
3. The $(n+1)$ faces of K are the intersections of K and the n hyperplanes $\lambda_j(x) = 0$, $j = 1, \cdots, n$.

Make a figure.

2.3.4.2 Lattices

To go further.

From these barycentric coordinates, one can define particular points sets of the n-simplex K: the so-called *lattices of order k*.

A lattice of order k denoted by Σ_k is defined by:

$$\Sigma_k = \{x \in K \text{ s.t. } \lambda_j(x) \in \{0, \frac{1}{k}, \cdots, \frac{(k-1)}{k}, 1\},\ \text{for } j = 1, \cdots, n\} \tag{2.3.17}$$

For $k = 1$, Σ_1 is the set of vertices of K.
For $k = 2$, Σ_2 equals Σ_1 plus the midpoints of the edges, see Fig. 2.3.4; etc.

Let us denote by $\{M_1, \cdots, M_{nk}\}$ the nk points contained in the lattice Σ_k.
For example, $\Sigma_1 = \{A_1, A_2, A_3\}$; $n1 = 3$.
In an n-simplex K, the lattice Σ_k enables to characterize the polynomials of P_k as follows.

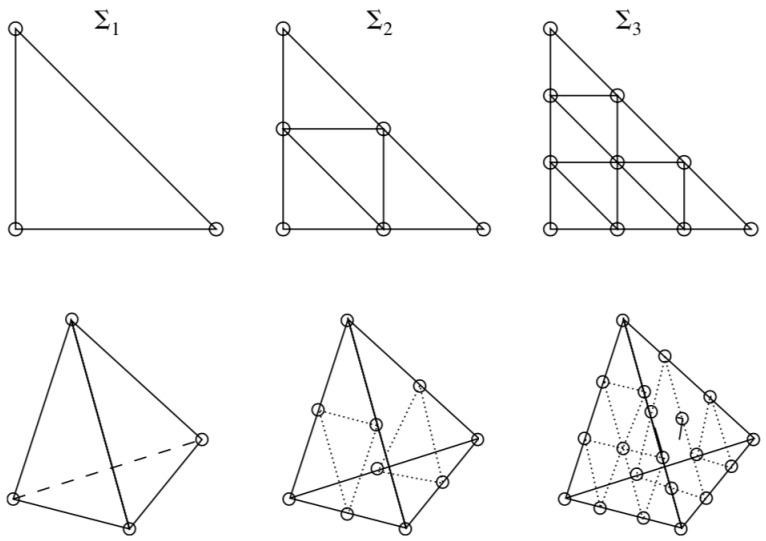


Figure 6.9. Lattice of order 1, 2, and 3 for a triangle (top) and a tetrahedron (bottom). The circles represent the points of the lattice.

Figure 2.3.4: Lattice of order 1, 2, and 3 for a triangle (top) and a tetrahedron (bottom). The circles represent the points of the lattice. Image extracted from [**?**].

Lemma 35. *Let K be an n-simplex.*
For $k \geq 1$, any polynomial of $\mathbf{P}_k$ is uniquely determined by its values at the lattice points $\{M_1, \cdots, M_{nk}\}$.
In other words, it exists a basis $\{\psi_1, \cdots, \psi_{nk}\}$ of $\mathbf{P}_k$ such that: $\psi_i(M_j) = \delta_{ij} \qquad 1 \leq i, j \leq nk$.

Proof. Please consult [**?**].

2.4 Convergence and error estimation

Céa's lemma (Lemma [**?**]) shows that the (a-priori) FE error $\|u - u_h\|_V$ in the "energy space" V is upper bounded by the distance separating the continuous solution $u \in V$ to the discrete FE space V_h.

Below we define the *interpolation operator* π_h in V_h and we present the resulting interpolation error.

2.4.1 Interpolation operator & error

We define the $\mathbf{P}_k$-*Lagrange interpolation operator* π_h as follows:

$$\text{For all } v \in V, \quad \pi_h(v)(x) = \sum_{i=1}^{NN} v(x_i)\varphi_i(x) \quad \forall x \in \Omega \tag{2.4.1}$$

with $\{\varphi_i(x)\}_{i=1\ldots NN}$ the $\mathbf{P}_k$-Lagrange FE basis.

If considering $\mathbf{P}_1$-Lagrange FE basis ($k = 1$) then the interpolation function $\pi_h(v)(x)$ is simply the piecewise linear function which coincides with the values of v at nodes $x_i,\ i = 1, .., NN$ ($\{x_i\}_i$ are the mesh vertices too), see Fig.2.1.4 (Down).

In 1D, $\pi_h(v)$ is well defined for any function in V since $V = H^1(I) \subset C^0(\bar{I})$, see e.g. Fig. 2.1.4 (Down).

On the contrary in higher dimensions ($n = 2,\ 3$), the functions v of $H^1(\Omega)$ are generally not continuous at all points... Therefore, the interpolated functions v will need to be more regular than $H^1(\Omega)$ to be in $H^1(\Omega) \cap C^0(\bar{\Omega})$.

Note that we have in 2D and 3D the following inclusion: $H^2(\Omega) \subset C^0(\bar{\Omega})$.

The following estimation of the interpolation error is technical but crucial to establish the forthcoming FE error estimation.

Proposition 36. Let V_h be the FE space defined from the $\mathbf{P}_k$-Lagrange FE basis, see (2.2.15).

The following properties hold.

i) For all $v \in H^1(\Omega) \cap C^0(\Omega)$, the interpolation function $\pi_h(v)$ defined by (2.4.1) is well defined and the interpolation error is convergent:

$$\lim_{h\to 0} \|v - \pi_h(v)\|_{H^1} = 0 \quad \forall v \in H^1(\Omega) \tag{2.4.2}$$

ii) For all $v \in H^{k+1}(\Omega)$, the interpolation function $\pi_h(v)$ is well defined and it exists a constant $c > 0$ (c independent of v and h) such that:

$$\|v - \pi_h v\|_{H^1} \leq c \;\; h^k \; \|v\|_{H^{k+1}} \tag{2.4.3}$$

Proof. Note that in 2D and 3D cases ($n = 2, 3$), we have $(k+1) > \frac{n}{2}$, therefore $H^{k+1}(\Omega) \subset C^0(\Omega)$, see Lemma 6. Therefore, here by assumption the interpolation function $\pi_h(v)$ is well defined.

We refer to [**?**] for the proof of the proposition: Section 6.2.2 in 1d and Section 6.3.2 (Proposition 6.3.16) in nD.

2.4.2 FE error estimation in the energy space V

2.4.2.1 A-priori error estimation: general case

As an immediate consequence of the interpolation error estimations above and Céa's lemma, it follows: a) the convergence of the FE scheme follows; b) an estimation of the FE error in the "energy space" $V = H^1(\Omega)$.

We have

Theorem 37. *Let u be the (unique) solution of the general weak formulation (2.1.2), let V_h be the $\mathbf{P}_k$-Lagrange FE space defined by (2.2.15) and let u_h be the solution of (2.1.5) (i.e. the FE solution). The Lax-Milgram theory assumptions are supposed to be satisfied.*
Then the $\mathbf{P}_k$-Lagrange FE scheme is convergent:

$$\lim_{h\to 0} \|u - u_h\|_{H^1} = 0 \tag{2.4.4}$$

Moreover if $u \in H^{k+1}(\Omega)$, it exists a constant $c > 0$ (c is independent of u and h) such that:

$$\|u - u_h\|_{H^1} \leq c \; h^k \; \|u\|_{H^{k+1}} \tag{2.4.5}$$

Proof. It is straightforward; the proof relies on the (relatively difficult) interpolation error estimation (2.4.3).

Indeed Lemma 26 states that:

$$\|u - u_h\|_{H^1} \leq c \inf_{v_h \in V_h} \|u - v_h\|_V$$

Using (2.4.3) we obtain (2.4.5).

2.4.2.2 Typical cases

The BVP is (linear) second order model (e.g. based on the Laplace operator) then the "energy space" is a subspace of $H^1(\Omega)$. Ω may be a 2d or 3d geometry, therefore the condition $(k+1) > \frac{n}{2}$ is satisfied.

Typical cases are the following:

- Linear elements ($k = 1$) and a "regular" (H^2) solution.
  If the exact solution $u^{ex} \in V \cap H^2(\Omega)$ then:

$$\|u^{ex} - u_h\|_{H^1} \leq c \; h \; \|u^{ex}\|_{H^2} \tag{2.4.6}$$

  Then the FE scheme is linear (order 1) in the space energy V.

- Quadratic elements, $k = 2$ and an "extra regular" (H^3) solution.
  If the exact solution $u^{ex} \in V \cap H^3(\Omega)$ (i.e. the exact solution needs to be more regular than before...) then:

$$\|u^{ex} - u_h\|_{H^1} \leq c\ h^2\ \|u^{ex}\|_{H^3} \tag{2.4.7}$$

  Then the FE scheme is quadratic (order 2) in the space energy V (if the solution is regular enough).

Remark 38. The actual FE scheme order of a computational code may be numerically measured, see next paragraph for the method description.

When numerically measuring the FE scheme order, it turns out that the FE error estimation (2.4.5) is optimal if.... the solution is regular enough.

However this holds if the integrals (of the stiffness matrix coefficients and the RHS) are numerically evaluated without errors: the numerical integration have to be consistent with the target accuracy.

2.4.2.3 On the numerical integration errors

In practice, these integrals are not exactly evaluated since they are computed by numerical integration. Nevertheless, if the employed quadrature formulas are "high-order enough", the order k of the FE scheme should be recovered.

Indeed the discrete weak form $a(u_h, v_h) = l(v_h)\ \forall v_h \in V_h$ is equivalent to the linear system $AU_h = b$ with the coefficients:

$$a_{ij} = a(\varphi_j, \varphi_i) \text{and } l_i = l(\varphi_i)$$

For example: $a(\varphi_j, \varphi_i) = \int_\Omega \lambda(x)\ \nabla\varphi_j \nabla\varphi_i(x)\ dx$ and $l(\varphi_i) = \int_\Omega f(x)\varphi_i(x)\ dx$.

However what is actually solved is the formulation:

$$a_h(u_h, v_h) = l_h(v_h) \qquad \forall v_h \in V_h \tag{2.4.8}$$

That is the coefficients of the linear system are (in the example above):

$$a_{ij} = a_h(\varphi_j, \varphi_i) = \sum_{p=1}^{NGP} \omega_p\ \lambda(x_p)\ \nabla\varphi_j \nabla\varphi_i(x_p) \sim a(\varphi_j, \varphi_i)$$

$$l_i = l_h(\varphi_i) = \sum_p \omega_p\, f(x_p)\varphi_i(x_p) \sim l(\varphi_i)$$

Therefore the quadrature formulas have to be accurate enough to preserve the order k of the FE scheme. It is worth to notice that this feature depends on the BVP data regularity $\lambda(x), f(x)$.

On the quadrature formula $\int \Psi(x)dx \sim \sum_p \omega_p \psi(x_p)$. WE onvite the reader to consult the complementary documents available on the Moodle page.

A typical example in 2d for a triangle $T = (a_1 a_2 a_3)$ with middle edges denoted by a_4, a_5, a_6: $\int_T \psi(x)dx \sim \frac{1}{3}|T| \sum_{i=4}^{6} \psi(a_i)$.

This formula is exact for $\psi \in \mathbf{P}_2$. fi

2.4.3 Measuring the convergence order: code validation

The best way to assess a FE computational code is to perform convergence curves following the technique described below.

The method

The principle is as follows.

1) We set the sough solution $u^{ex}(x)$ e.g. $u^{ex}(x_1, x_2) = \cos(\omega_1 x_1)\sin(\omega_2 x_2)$.

2) We calculate the corresponding RHS: $f(x) = A(u^{ex}(x))$ with $A(.)$ the differential operator e.g. $A(u^{ex}) = -div(\lambda\nabla u^{ex}) + cu^{ex} = ... \to f(x)$.

3) We compute the numerical solution u_h with the code and we plot the FE error value $\|u^{ex} - u_h\|_V$ vs a characteristic mesh size h.

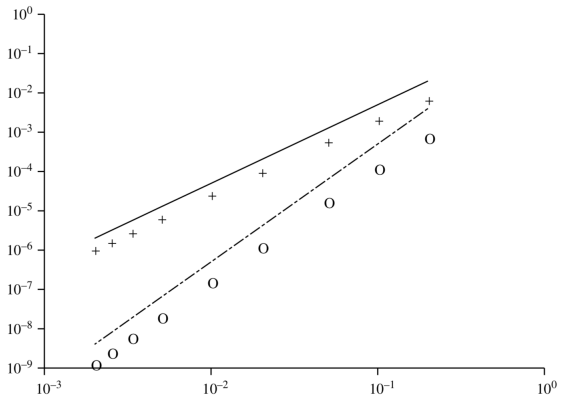


Figure 6.4. Case of a regular solution: example (6.29). Discrete H^1 norm of the error as a function of the size h of the mesh (the crosses correspond to $\mathbb{P}_1$ finite elements, the circles to $\mathbb{P}_2$ finite elements, the lines are the graphs of $h \to h^2$ and $h \to h^3$).

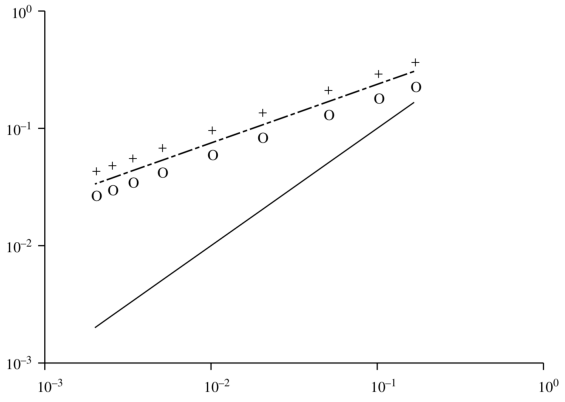


Figure 6.5. Case of a nonregular solution: example (6.30). Discrete H^1 norm of the error as a function of the mesh size h (the crosses correspond to $\mathbb{P}_1$ finite elements, the circles to $\mathbb{P}_2$ finite elements, the lines are the graphs of $h \to \sqrt{h}$ and $h \to h$).

Convergence curves from explicit solutions (1d solutions). (L) The optimal rate is obtained (the solution is regular); the FE code is fully assessed. (R) The obtained rate of convergence is not optimal since the present exact solution is not regular (here it is not in $H^2(\Omega)$). Curves extracted from [?].

We do so for a few values of h; typically for 4 mesh (therefore 4 values of h).
It turns out that the estimation (2.4.5) is optimal in the sense that if the chosen exact solution u^{ex} is regular then: $\|u^{ex} - u_h\|_{H^1} \sim c\ h^k\ \|u^{ex}\|_{H^{k+1}}$.

Therefore we have:

$$\ln(\|u^{ex} - u_h\|_{H^1}) \sim \text{cste} + k\ \ln h \tag{2.4.9}$$

Hence in $\ln - \ln$ scale the convergence curve is linear, with a slope equal to the order of convergence k, see Fig. 2.4.3(L).

It is possible that a super-convergence phenomena appears (the observed converge rate is higher than the general theoretical one). This super-convergence phenomena is generally due to the uniformity of the mesh.

Such convergence curves constitute the most robust validation method of the computational code.

However, this method is possible for simple models only since based on an exact solution. Indeed, if the model is complex (eg multi-physics system, potentially coupled) then the calculations to obtain u^{ex} may be too difficult to obtain.

In such complex cases, the reference solution (considered as almost exact) may be a solution computed on an extremely fine mesh. Next, the differences (considered as being the "errors") are evaluated between this reference solution and the others obtained on the few (eg 4) coarser meshes with $h_{coarse}, h_{coarse}/2, h_{coarse}/4$ etc.

An illustration of mesh refinement is shown in Fig. 2.4.1 (L): h is locally decreased therefore a (local) decreasing of the error. This is a called "h-adaptivity".

Moreover, to validate a computational code, one generally consider benchmarks too.

On $h-p$ adaptivity

The accuracy of the FE scheme can be improved too by increasing the order p of the polynomials in areas where the exact solution u is regular enough, see Theorem 37.
This is the "p-adaptivity", see Fig. 2.4.1(R).

Combinations of these two types of adaptivity is called hp-adaptivity.

So-called "hp-FEM" combines adaptively elements with variable size h and polynomial degree p in order to achieve higher accuracy and/or convergence rates.

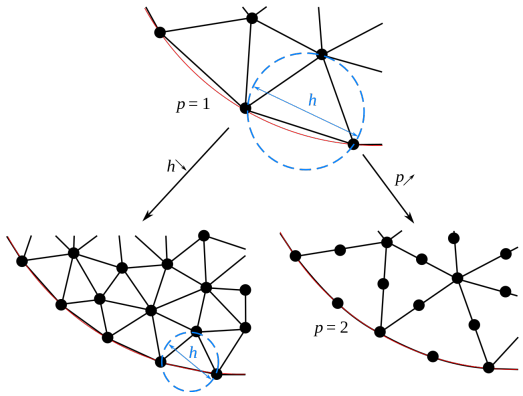


Figure 2.4.1: "h or p" refinements?
(L) Mesh refinement : h decreases, the accuracy locally increases.
h-adaptivity (refining and un-refining elements) is well adapted for local singularities.
(R) Accuracy improvement by increasing the FE order : order p of polynomials is increased.
p-adaptivity is well adapted for smooth solutions. Image source: Wikipedia.

‘

Figure 2.4.2: "h or p" refinements? Convergence curves *error vs #dofs* in a particular case. Image source: agros2d.org.

In practice,

- in areas where the solution is regular, decreasing the FE error is achieved by increasing the polynomials order p.
- on the contrary, in areas where the solution lacks of regularity (e.g. a local singularity being between H^1 and H^2), it is useless to increase the polynomials order p (see e.g. (2.4.5)).
  In this case, decreasing the FE error may be achieved by decreasing the element size h (h adaptivity).

2.4.4 On non optimal FE scheme order: presence of singularity

The FE error estimation (2.4.5) is true if the exact solution u is regular enough. This may be not the case... See Paragraph 1.3.3.2.
If u is not regular, we say u is singular. Singularities are generally local only. Numerically the global convergence rate is observed to be weaker. Typically the obtained order equals $\frac{1}{2}$ (resp. $\frac{3}{2}$) instead of 1 (resp. 2), see e.g. Fig. 2.4.3(R).

As an example let us consider the following BVP: the classical Laplace equation with constant Dirichlet conditions solved with a RHS $f(x) = 1 \quad \forall x$. This BVP is regular, data are all C^∞, but it is solved in a domain Ω presenting a ‘re-entrant corner’. This ‘re-entrant corner’ generates a local singularity on the solution u : the solution may be C^∞ far enough from the corner vertex but is $H^{3/2}(\Omega)$ in a vicinity of this re-entrant corner. (This can be mathematically demonstrated for the Laplace operator).

Therefore u is not (globally) $H^2(\Omega)$. This implies that the gradient of u_h is not $L^\infty(\Omega)$.

Then it is observed that more h tends to zero more $\|\nabla u\|_{L^\infty}$ increases: it diverges, see Fig. 2.4.4.

This example highlights a typical singularity for 2nd order BVP (see Paragraph 1.3.3.2 too).

Remark 39. One may enrich the approximation space V_h with well-chosen local singular functions enabling to better approximate the targeted singularity; this is called *extended finite element method (xFEM)*. The (great) difficulty consists to find the adequate singular function(s)...

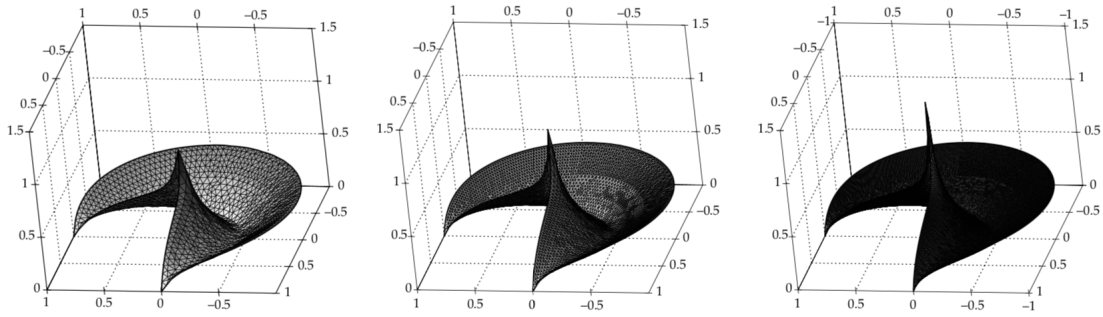


Figure 2.4.3: (L) $\|\nabla u\|_{L^\infty} \approx 0.92$ for the "coarse" mesh (1187 vertices). (M) $\|\nabla u\|_{L^\infty} \approx 1.18$ for the intermediate mesh (4606 vertices). (R) $\|\nabla u\|_{L^\infty} \approx 1.50$ for the "fine" mesh (18 572 vertices). Example and image extracted from [?].

2.4.5 Error estimation in norm $L^2(\Omega)$

Theorem 37 shows that the $\mathbf{P}_k$-Lagrange FE *error in* $H^1(\Omega)$ i.e. in the energy norm, behaves as h^k (if the exact solution is regular enough).

Under the same assumptions as Theorem 37, it can be proved (see e.g. [1]) that the $\mathbf{P}_k$-Lagrange FE *error in* $L^2(\Omega)$ (gradients are not "measured") satisfies:

$$\|u - u_h\|_0 \leq c\ h^{k+1}\ \|u\|_{H^{k+1}} \tag{2.4.10}$$

That is that the $\mathbf{P}_k$-Lagrange FE *error in* $L^2(\Omega)$ behaves as h^{k+1} (if the exact solution is regular enough): one order is gained compared to the error in the energy norm $H^1(\Omega)$.

In practice convergence curves in norm L^2 provide a convergence rate equal to $(k+1)$; again if the exact solution is regular enough.

2.5 Hermite FE: a brief presentation

In this section, we briefly present the Hermite FE principles, mainly in 1D.

For high-order BVP, typically order 4 such as the bi-laplacian Δ^2 (this operator models linear plate deformations), we may be interested to consider more regular solution: discrete solutions which are globally C^1 and not C^0 only (like those obtained if using P_k-Lagrange FE).

To do so, in 1D the considered FE space V_h reads:

$$V_h \equiv V_h^{H,1D} = \{v_h, v_h \in C^1([0,1]), \quad v_h|_{[x_i,x_{i+1}]} \in P_3 \quad \forall i,\ i = 1, \dots NN\} \tag{2.5.1}$$

In 1D the minimal regularity of polynomials to C^1 -connect at triangle edges / tetrahedra faces is P_3.

A 1D Hermite element is plotted on Fig. 2.5.

Every function v_v of V_h is (uniquely) defined by its values and its derivatives values at the nodes.

In 1D, this reads:

$$v_h(x) = \sum_{i=1}^{NN} v(x_i)\varphi_i(x) + \sum_{i=1}^{NN} v'(x_i)\psi_i(x) \quad \forall x \in \Omega \tag{2.5.2}$$

In 2D the Hermite FE space reads:

$$V_h \equiv V_h^{H,2D} = \{v_h, v_h \in C^1(\bar{\Omega}), \quad v_h|_{K_i} \in P_5 \quad \forall K_i \in \mathcal{T}_h\} \tag{2.5.3}$$

In 2D the minimal regularity of polynomials to C^1 -connect at triangle edges is P_5.

Any polynomial $p \in P_5$ is uniquely defined on a triangle by the following 21 values:

$$\{p(a_i),\ \partial_1 p(a_i),\ \partial_2 p(a_i),\ \partial^2_{11} p(a_i),\ \partial^2_{22} p(a_i),\ \partial^2_{12} p(a_i),\ \partial_n p(a_{ij})\}_{1 \leq i < j \leq 3} \tag{2.5.4}$$

with $\{a_i\}_{1 \leq i \leq 3}$ the triangles vertices and $\{a_{ij}\}_{1 \leq i < j \leq 3}$ its edges midpoints.

[1] Quarteroni, Alfio, and Alberto Valli. "Numerical approximation of partial differential equations". Springer Science, 2008.

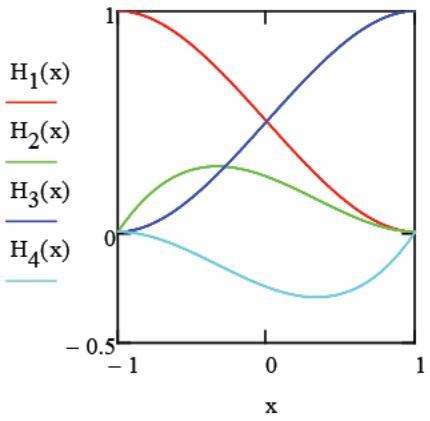


Figure 2.5.1: Hermite FE. Basis functions on the elementary segment $[-1,+1]$ (hence defined by 2 nodes = the 2 element vertices): 2 nodes, 4 dof. Function basis are polynomials in P_3.

This Hermite FE method is suitable to solve the bi-Laplacian equation (e.g. modeling a plate deformation); it is called the Argyris element. Its analysis is proposed in exercise (see Moodle page).

For all $v_h \in V_h$, $\partial_{x_i} v_h(x)$ is continuous and piecewise C^1; therefore $\partial_{x_i} v_h$ belongs to $H^1(\Omega)$ and $v_h \in H^2(\Omega)$.
For more details on Hermite FE we refer to the dedicated exercises session.

2.6 Appendix: Formalization of what is a FE

To go further...

2.6.1 A definition of FE & "P-unisolving property"

A Finite Element (FE) may be defined as follows.

Definition 40. We call a Finite Element (FE) a triplet (K, P_K, Σ_K) with:

- K an elementary geometry (e.g. triangle, tetrahedra, quadrangle, parallelepiped, prism),
- P_K a space function (the basis functions space),
- $\Sigma_K = \{N_1 \cdots N_{NNE}\}$ a set of points (the nodes) satisfying the "P_K-unisolving property" below.
  The P_K-unisolving property characterizes the consistency between P_K and Σ_K.

Definition 41. The nodes set Σ_K is P_K-unisolving if and only if for any values set $\{\alpha_i\}_{i=1..NN}$ corresponds an unique $p \in P_K$.

In the Lagrange FE case (vs Hermite FE case), this means that it exists an unique $p \in P_K$ interpolating the nodes values on K:

$$p(N_i) = \alpha_i \quad i = 1, \cdots, NNE \tag{2.6.1}$$

Given a triplet (K, P_K, Σ_K), how to verify that Σ_K is P_K-unisolving ? A necessary condition of P_K-unisolving property is:

$$dim(P_K) = Card(\Sigma) = NNE \tag{2.6.2}$$

Next verifying that Σ_K is P_K-unisolving may done by checking one of the following criteria:

1. Show that:
$$p(N_i) = 0 \Longrightarrow p \equiv 0 \quad \forall i,\ 1 \leq i \leq NNE \tag{2.6.3}$$
2. Write the expression of the basis functions of P_k.

Let us prove that Criteria 1. above implies that Σ_K is P_K-unisolving.
Let us define the application $\mathcal{L}_{\kappa}$ by:

$$\mathcal{L}_{\kappa} : p \in P_K \mapsto \{p(N_1), \cdots, p(N_{NNE})\} \in R^{NNE} \tag{2.6.4}$$

$\mathcal{L}_{\kappa}$ is linear. We have the following property:

$$\text{"}\Sigma_K \text{ is } P_K\text{-unisolving"} \Leftrightarrow \mathcal{L}_{\kappa} \text{is bijective} \tag{2.6.5}$$

Therefore (2.6.3) is equivalent to $\mathcal{L}_{\kappa}$ is injective (since $\mathcal{L}_{\kappa}$ is linear). Moreover since acting in finite dimensional spaces ($\dim(P_K) = NNE$), this is equivalent to $\mathcal{L}_{\kappa}$ bijective; and the result follows from (2.6.5).

Criteria 2. above implies that Σ_K is P_K-unisolving too. Indeed basis functions $\{\varphi_i(x)\}_{i=1,..,NNE}$ of P_k enables to prove that $\mathcal{L}_{\kappa}$ is surjective therefore bijective.

In the (usual) case where $\varphi_i(N_j) = \delta_{ij},\ 1 \leq i, j \leq NNE$, we have:

$$p(x) = \sum_{k=1}^{NNE} \alpha_i \varphi_i(x) \Rightarrow p(N_j) = \alpha_i,\ 1 \leq i \leq NNE \tag{2.6.6}$$

i.e. the coefficients of p in the functions basis are the nodes values.

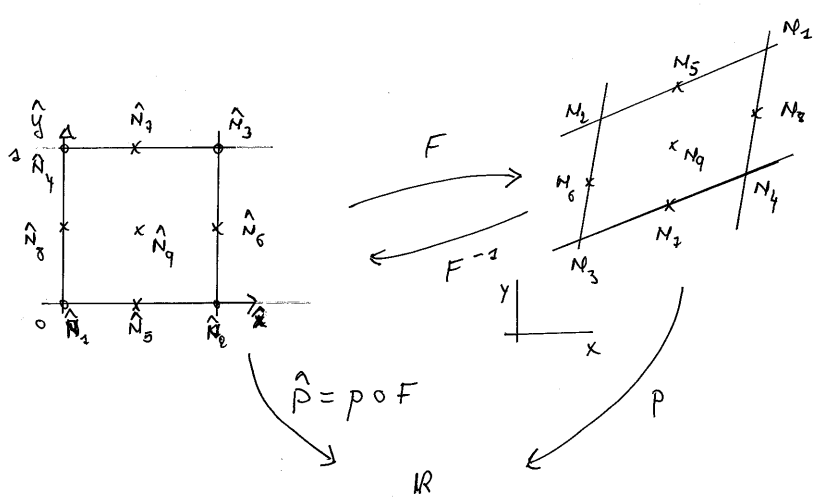


Figure 2.6.1: Generation of finite elements (K, P, Σ) from the reference finite element $(\hat{K}, \hat{P}, \hat{\Sigma})$. Here in the case of a $\mathbf{Q}_2$-Lagrange FE with an affine change of variable (geometric transformation) F.

2.6.2 Generating finite elements from the reference element

Let us show how to generate P_k and Q_k Lagrange FE from their corresponding reference element.

Proposition 42. *Let F be the C^1 bijective function (a change of variables) transforming the reference element (triangle, tetrahedra, square, cube) $\hat{K}$ into K (resp. triangle, tetrahedra, parallelogram, parallelepiped), see Fig 2.3.3.1.*
Let $(\hat{K}, \hat{P}, \hat{\Sigma})$ be a $\mathbf{P}_k$ (resp. $\mathbf{Q}_k$) Lagrange FE . Then the triplet (K, P, Σ) defined by:

$$K = F(\hat{K});\ \Sigma = F(\hat{\Sigma}) \text{ and } P = \{p : K \to \mathbf{R};\ (p \circ F) \in \hat{P}\} \tag{2.6.7}$$

is a $\mathbf{P}_k$ (resp. Q_k) Lagrange FE too (in the sense of Definition 40).

We say that the FE $(\hat{K}, \hat{P}, \hat{\Sigma})$ and the FE (K, P, Σ) are equivalent.
Moreover if the geometric transformation F is affine, we say that $(\hat{K}, \hat{P}, \hat{\Sigma})$ and (K, P, Σ) are affine-equivalent.
Proof. We set: $\hat{\Sigma} = \{\hat{N}_i\}_{i=1..NNE}$ and $N_i = F(\hat{N}_i)$; then: $\Sigma = \{N_i\}_{i=1..NNE}$.
We have F bijective from $\hat{K}$onto K then: $dim(\hat{P}) = Card(\hat{\Sigma}) = NNE = dim(P) = Card(\Sigma)$.
It remains to check that Σ is P-unisolving.
We have: $\forall \hat{p} \in \hat{P}$, $p = (\hat{p} \circ F^{-1}) \in P$.
Let $\{\hat{\varphi}_i(x)\}_{i=1,..,NNE}$ be the basis functions of $\hat{P}$. We set:

$$\varphi_i = \hat{\varphi}_i \circ F^{-1}$$

We have: $\varphi_i = \hat{\varphi}_i \circ F^{-1}(N_j) = \hat{\varphi}_i(\hat{N}_j) = \delta_{ij}$ for all i, j. $\{\varphi_i(x)\}_{i=1,..,NNE}$ are the basis functions of P.
The triplet (K, P, Σ) is a FE (see Definition 40).

2.7 Computational freewares

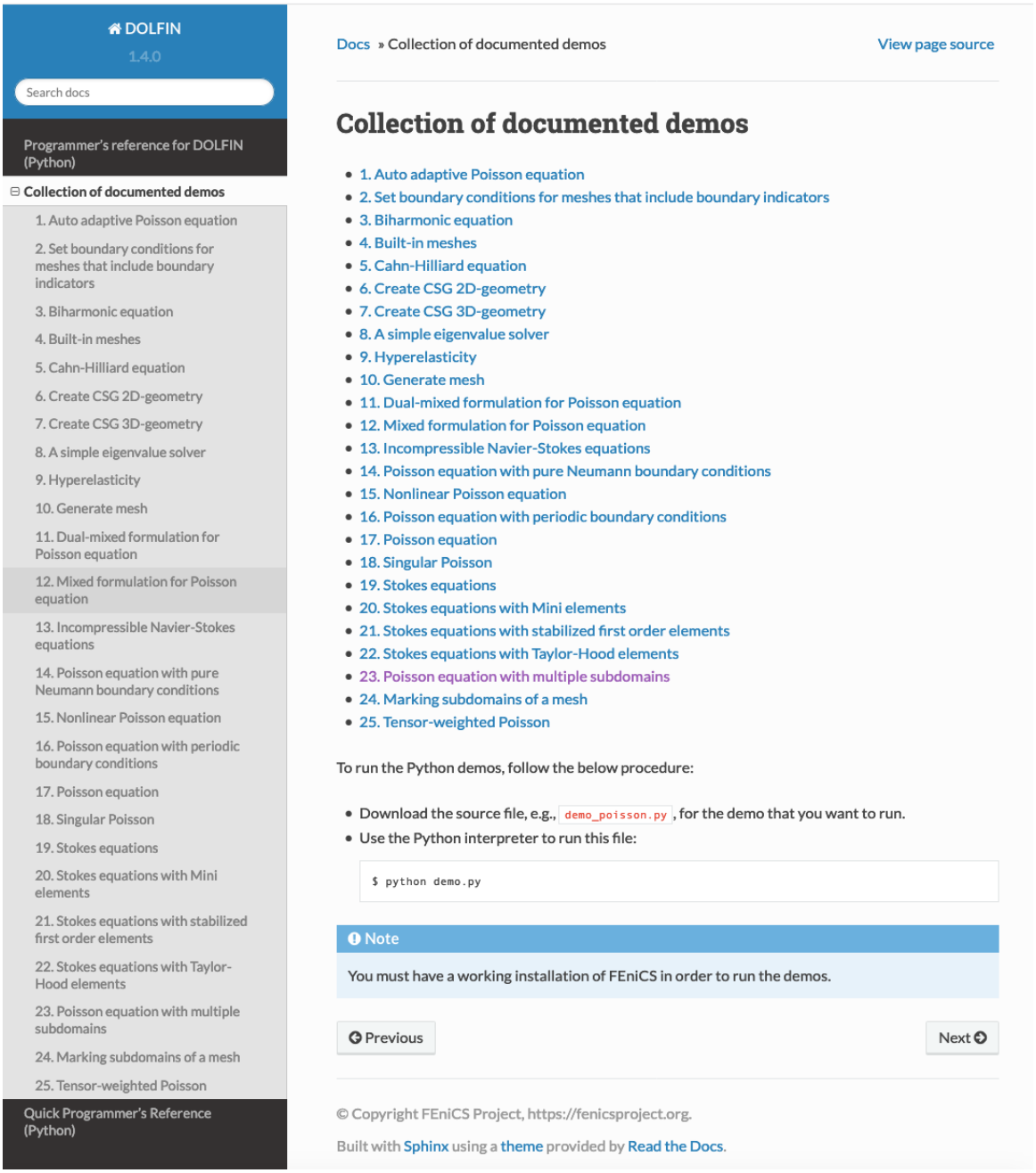


Figure 2.7.1: FEniCS Project (https://fenicsproject.org) in Python: standard PDE models and FE schemes are available. (Snapshot made in february 2019).

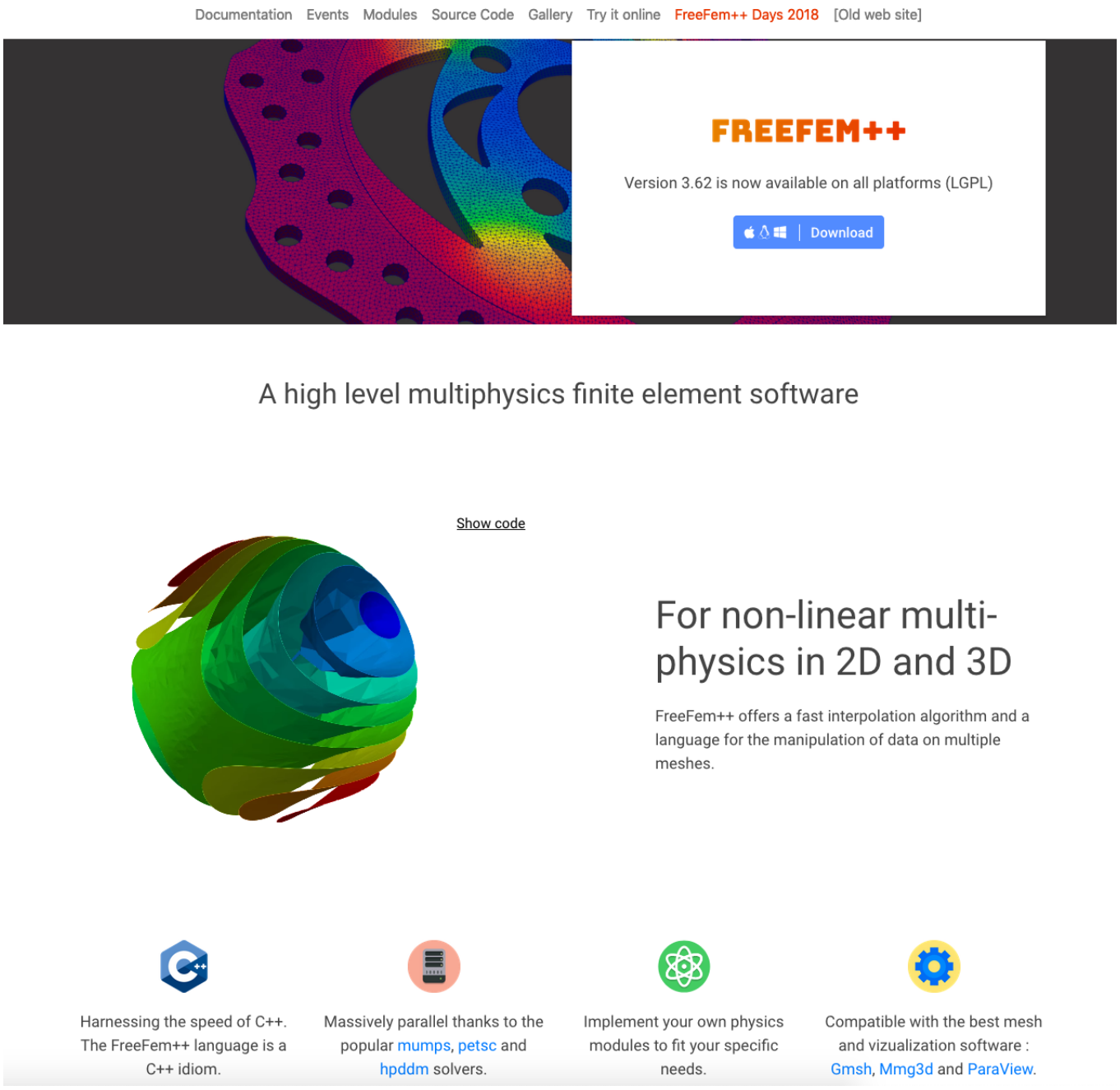


Figure 2.7.2: FreeFem++: A high level multi-physics FE software for non-linear multi-physics in 2D and 3D. (Snapshot made in february 2019).

Chapter 3

Finite Element Methods: Complements

Finite Element Methods (FEM) are the numerical methods of choice to solve elliptic models (e.g. those based on the Laplace equation or on the advection-diffusion equation) and parabolic models (e.g. the heat equation). They can be used for hyperbolic models too (e.g. the transport equation) by introducing stabilizing terms (e.g. artificial diffusion).

The basic principles of FEM are presented in Part 1 of the course manuscript entitled "Finite Element Methods: Fundamentals".

In this chapter you will learn how:

- to solve a non-linear model by FEM using the Newton-Raphson algorithm,
- to write FE schemes for time-dependent PDEs,
- to stabilize a FE scheme in presence of an advective term (transport term),
- an automatic mesh refinement works.

On all these topics, the reader may refer to e.g. [**?**, **?**].

3.1 Non-linear stationary PDEs: linearization

3.1.1 The (scalar) non-linear BVP

Let us consider a non-linear BVP:the PDE or one of its boundary condition is non linear with respect to the unknown function $u(x)$. More precisely, we consider the following variational problem.

Find $u \in V$satisfying:

$$a_{nl}(u, v) = l(v) \quad \forall v \in V$$

Where:

- $u \mapsto a_{nl}(u, \cdot)$ is continuous in V but *non-linear*,
- $v \mapsto a_{nl}(\cdot, v)$ and $v \mapsto l(v)$ are linear continuous in V.

Simple examples are the following scalar PDEs:

- $-\mu \triangle u + u^3 = f$
- $-div(\mu(u)\nabla u) + u = f$

accompanied by boundary conditions on $\partial\Omega$.

Remark 43. It is convenient and generally possible to consider the non-linear form $a_{nl}(u, \cdot)$ as follows: $a_{nl}(u, \cdot) = a(u; u, \cdot)$, where

- $u \mapsto a(u;\cdot,\cdot)$ *is non-linear:* this partial map represents the non-linear term(s) of the form;
- $u \mapsto a(\cdot;u,\cdot)$ is *linear:* this partial map represents the linear terms of the form;

By construction, $v \mapsto a(\cdot, v)$ and $v \mapsto l(v)$ are always linear continuous in V.

Then, the variational problem reads as follows:

$$(\mathcal{P}) \quad \begin{cases} \text{Find } u \in V \text{ satisfying:} \\ a(u;u,v) = l(v) \qquad\qquad \forall v \in V \end{cases} \tag{3.1.1}$$

On the mathematical analysis It is assumed that $(\mathcal{P})$ is well-posed in a Banach space V. Observe that since the PDE is non linear, the Lax-Milgram theory does not apply anymore. Moreover, the well-posedness of $(\mathcal{P})$ is a-priori in a Banach space and a-priori not in a Hilbert space.
For second-order problems, one typically has $V = W^{1,p}(\Omega)$ with $p \neq 2$ i.e. not the Sobolev space $H^1(\Omega)$....
However, the basic principle of weak formulations (and resulting weak solutions) remain the same.
A quite general theorem to address the existence of solutions for such elliptic non linear PDEs is the Leray-Schauder fixed point theorem.

3.1.2 Linearized discrete system

Let us employ an internal approximation as previously.
Since the equation is non linear (in u), the discrete variational formulation is *not* equivalent to a linear system.
Indeed:

$$a_{nl}(\sum_i u_i \varphi_i(x), \cdot) \neq \sum_i u_i a_{nl}(\varphi_i(x), \cdot) \tag{3.1.2}$$

The discrete variational formulation may be written as a non-linear system of the form:

$$A(U)U = b \tag{3.1.3}$$

with $U \in R^{NN}$ the dof vector, $A(U)$ a stiffness matrix depending on U and b the RHS.

Exercise. Write the variational (weak) formulation of a scalar diffusion equation with the diffusivity parameter μ depending on the solution u.

Let us set $F : U \in \mathbb{R}^{NN} \mapsto F(U) \in R^{NN}$ with

$$F(U) = A(U)U - b \text{ in } R^{NN} \tag{3.1.4}$$

The most classical methods to solve the non-linear $(NN \times NN)$-system (3.1.4) are:

- A fixed point method as $A(U^{n-1})U^n = b$.
- The Newton-Raphson method based on the linearized PDE therefore the differential $DF(U)$.

Recall that the fixed point method is trivial to implement; it converges if $F(\cdot)$ is L-Lipschitz with $L < 1$. It is a 1st order method only (if converging).

On the contrary, the Newton-Raphson method may be more complex to implement since it requires the differential $DF(\cdot)$. However it is a 2nd order method hence much faster, if converging. The attractor(s) basins of these two methods are different.

Exercise. Let us consider the (discrete) non-linear system $F(U) = 0$ with $F : U \in \mathbb{R}^{NN} \mapsto F(U) \in R^{NN}$.
Derive the Newton-Raphson algorithm.
Show that, in the end this consists to compute at each iteration the increment ΔU satisfying the following linear system:

$$DF(U^{(k)}).\Delta U = -F(U^{(k)}) \tag{3.1.5}$$

3.1.3 Linearized PDE (continuous form)

An elegant and concise approach consists to write the non linear variational problem and its corresponding linearized equation in continuous form (in the function space V), instead of manipulating potentially "heavy" discrete equations.

The map $v \mapsto (a(\cdot;\cdot,v) - l(v))$ is linear. Then, we set the map $F(u)$ such that: $F(u)(v) = a(u;u,v) - l(v)$.

Given u, $F(u)$ is a linear map acting on V. Therefore, by definition $F(u)$ is an element of V': $F(u) \in V'$.

Thus we have:

$$F : V \to V^{'} \text{ with } < F(u), v >_{V'\times V} \equiv F(u)(v) = a(u;u,v) - l(v) \tag{3.1.6}$$

where $< .,. >_{V'\times V}$ denotes the so-called *duality product* $V' \times V$.

Then the equation of problem $(\mathcal{P})$, see (3.1.1), re-reads as:

$$F(u) = 0 \text{ in } V' \tag{3.1.7}$$

Equivalently: $< F(u), v >_{V'\times V} = 0 \quad \forall v \in V$.

Recall that V is an infinite dimensional space. Moreover, given $u \in V$, the differential $DF(u) \in \mathcal{L}(V, V')$.

On the dual space of a Hilbert space H Concerning the connection between a Hilbert space H and its dual H', the Riez-Fréchet theorem constitutes a very nice and useful result. The reader may study the Riez-Fréchet representation theorem presented in a supplementary note.

Moreover, a few exercises consisting of applying the Riez-Fréchet theorem in the present context are proposed.

The Newton-Raphson algorithm in continuous form

Based on the formalism above, the Newton-Raphson algorithm reads as follows.

Algorithm 3.1 Newton-Raphson

- Given $u^{(0)}$ (the best "first guess" as possible),
- $k \to (k+1)$:
  - Calculate or compute the differential $DF(u^{(k)})$.
  - Solve the linearized equation which reads:

$$\begin{cases} \text{Find } \delta u \in V \text{ such that:} \\ < DF(u^{(k)}) \cdot \delta u, v >_{V'\times V} = - < F(u^{(k)}), v >_{V''\times V} \qquad \forall v \in V \end{cases} \tag{3.1.8}$$

  Using the notation $a(w;u,v)$, this equation re-reads:

$$\partial_w a(u^{(k)};u^{(k)},v) \cdot \delta u + a(u^{(k)};\delta u, v) = -a(u^{(k)};u^{(k)},v) + l(v) \qquad \forall v \in V \tag{3.1.9}$$

- Update the solution: $u^{(k+1)} = u^{(k)} + \delta u$
- Test of convergence.
  After FE discretization the convergence criteria may be: $\frac{\|u^{(k+1)} - u^{(k)}\|}{\|u^{(k)}\|} < \varepsilon$.
  With $\varepsilon \approx 10^{-10}$ *since it is an order 2 method.*

Remark 44. Recall that the Newton algorithm is convergent at order 2 (quadratic convergence) if the differential $DF(u^{(k)})$ is locally Lipschitz and if the first "point" ("first guess") $u^{(0)}$ is close enough to the solution.

As a consequence, the choice of $u^{(0)}$ is crucial to make converge the algorithm.

Exercises. See the supplementary material.

3.2 FEM for unsteady PDEs (parabolic models)

In this section, we present how the FEM is adapted to unsteady BVP.

In short, FE are employed for the spatial discretization, while time schemes for ODEs (Euler, RKn etc) are employed for the temporal discretization.

3.2.1 The general model

Let us consider the general time-dependent BVP (first order in time):

$$\begin{cases} \partial_t u(x,t) + A(u(x,t)) & = f(x,t) \text{ in } \Omega\times]0,T[\\ u(x,0) & = u_0(x) \text{ in } \Omega \\ + \text{ B.C.} & \forall t \in]0,T[\end{cases} \tag{3.2.1}$$

where the Initial Condition (IC) $u_0(x)$ is given.

By default we assume in this section the same hypothesis as in the steady-case case in the following sense: at time t given, the differential operator satisfies the Lax-Milgram theory, see Section 2.1.

The typical example of linear parabolic equation is the heat equation. An extended version of the reference model includes a 0-th order term; it is the following unsteady *linear diffusion reaction model*:

$$\begin{cases} \partial_t u(x,t) - div(\mu\nabla u)(x,t) + c\,u(x,t) & = f(x,t) \text{ in } \Omega\times]0,T[\\ u(x,0) & = u_0(x) \text{ in } \Omega \\ u(x,t) & = u_d(x,t) \text{ in } \partial\Omega\times]0,T[\end{cases} \tag{3.2.2}$$

Of course, mixed boundary conditions could be considered to close the equation posed in Ω.

If the operator $A(u)$ is an elliptic operator e.g. like those adressed in the previous sections, this general BVP (3.2.1) is a parabolic model.

Recall that parabolic models have regularizing effects on the I.C.: even if u_0 is not regular e.g. not continuous, then the solution $u(x,t)$ immediately (that is for any $t > 0$) becomes regular. This feature is not true with hyperbolic models.

The basic principles of time discretization presented below formaly apply to non-linear parabolic PDEs, or even to hyperbolic PDEs, including those of second order in time (e.g. the wave equation $\left(\partial^2_{tt} u - \Delta u\right)(x,t) = 0$).

Of course, the B.C. have to be adequate with the differential operator $A(u)$.

3.2.2 Weak formulation

Weak formulations of time-dependent equations are built like in the stationary case:

the equation is multiplied by a test function $v(x)$ depending on the spatial variable x only i.e. not depending on the time variable t.

Remark 45. We could build a FEM in space *and* time. However, the resulting formulation is not interesting excepted if in a very few cases where Ω is time-dependent.

For the typical example (3.2.2), this reads:

$$\int_\Omega \partial_t u(x,t)v(x)\,dx + \int_\Omega \mu(x)\,\nabla u(x,t)\cdot\nabla v(x)\,dx + \int_\Omega c(x)u(x,t)v(x)\,dx = \int_\Omega f(x,t)v(x)\,dx \quad \forall v(x) \in V_0 \tag{3.2.3}$$

with $V_0 = \{v,\ v \in H^1(\Omega),\ v = 0 \text{ on } \partial\Omega\} = H^1_0(\Omega)$.

The temporal term satisfies:

$$\int_\Omega \partial_t u(x,t)v(x)\,dx = \frac{d}{dt}\int_\Omega u(x,t)v(x)\,dx = \frac{d}{dt}\left(u(t),v\right)_{L^2(\Omega)} \tag{3.2.4}$$

where $(\cdot,\cdot)_{L^2(\Omega)}$ denotes the L^2-scalar product in Ω.

Like in steady-state cases, we set: $V_t = \{v,\ v \in H^1(\Omega),\ v = u_d \text{ on } \partial\Omega\}$.

Then, by adopting the same notations as previously, we obtain the weak formulation:

$$\begin{cases} \text{Find } u(t) \in V_t \text{ such that:} \\ \frac{d}{dt}\left(u(t), v\right)_{L^2(\Omega)} + a(u(t), v) = b(v) \qquad \forall v \in V_0, \text{ for } 0 < t < T \end{cases} \tag{3.2.5}$$

where the solution $u(t,x)$ is here considered as a function of time t with values in V_0:

$$u : t \in]0,T[\mapsto u(t) \in V_t \tag{3.2.6}$$

Recall that $a(\cdot,\cdot)$ is here a bilinear form, continuous, and coercive in V_0.

To mathematically analyse (3.2.5), one has to clarify the regularity in time of the functions $u(t)$ and $f(t)$. Classical theorems establish the well-posedness of (3.2.5) in some functional spaces built from V_0.

Exercise. Write the weak formulation for the typical (linear) example.

Energy estimation

Exercise. Considering the BVP (3.2.2), show the following estimation:

$$\int_\Omega u^2(x,t)\ dx + \int_0^t \int_\Omega \left(\mu(x)\left|\nabla u\right|^2(x,s) + c(x)u^2(x,s)\right)\ dxds = \int_\Omega u_0^2(x)dx + \int_0^t \int_\Omega f(x,s)\ u(x,s)\ dxds \tag{3.2.7}$$

The resulting energy space is $L^2(]0,T[;V_0) \cap C^0(]0,T[;L^2(\Omega))$.

For details the reader may refer e.g. to [**?**] Section 8.2.

3.2.3 Semi-discretization in space: the mass matrix

The semi-discrete formulation consists to consider the weak formulation above and to apply a FEM like those studied in the steady-state cases.

We adopt here the same notations and the same FEMs as previously: *internal approximations based on P_k-Lagrange FE.*

The solution $u_h(t)$ is decomposed in the FE basis $\{\varphi_i(x)\}_{i=1..NN}$:

$$u_h(t)(x) \equiv u_h(x,t) = \sum_{j=1}^{NN} u_j(t)\,\varphi_j(x) \quad \text{for all } t,\ 0 < t < T \tag{3.2.8}$$

The dof $u_i(t)$ are here time-dependent.

We obtain the following *semi-discrete weak formulation* (discrete in space, continuous in time):

$$\begin{cases} \text{Find } u_h(t) \in V_{th} \text{ such that:} \\ \left(\frac{du_h}{dt}(t), v_h\right)_{L^2(\Omega)} + a(u_h(t), v_h) = b(v_h) \qquad \forall v_h \in V_{0h}, 0 < t < T \end{cases} \tag{3.2.9}$$

(3.2.9) is equivalent to:

$$\begin{cases} \text{Find } U_h(t) = (u_1(t), ..., u_{NN}(t)) \in R^{NN} \text{ such that:} \\ \sum_{j=1}^{NN} \left(\varphi_j(x), \varphi_i(x)\right)_{L^2(\Omega)} \frac{du_j}{dt}(t) + \sum_{j=1}^{NN} a(\varphi_j(x), \varphi_i(x))u_j(t) = b(t, \varphi_i(x)) \qquad \forall i,\ i = 1, \ldots, NN,\ 0 < t < T \end{cases} \tag{3.2.10}$$

with the I.C.: $u_i(0) = u_{0,i}$, $1 \le i \le NN$.

We recognize here the *stiffness matrix* A with, see Prop. 25: $A = (a_{ij})_{i,j=1..NN}$, $a_{ij} = a(\varphi_j, \varphi_i)$.

Moreover, a new matrix naturally appears in (3.2.10): this is the *mass matrix* M, $M = (m_{ij})_{i,j=1..NN}$,

$$m_{ij} = (\varphi_j, \varphi_i)_{L^2(\Omega)} = \int_\Omega \varphi_j(x)\varphi_i(x)\ dx \tag{3.2.11}$$

The mass matrix is sparse, symmetric, positive definite.
Using the matrix notations, (3.2.10) re-writes as:

$$\begin{cases} \text{Find } U_h(t) = (u_1(t), ..., u_{NN}(t)) \in R^{NN} \text{ such that:} \\ M\frac{dU_h}{dt}(t) + AU_h(t) = F(t) \quad \text{for } t \in]0,T[\end{cases} \tag{3.2.12}$$

with: $F = (f_i)_{i=1..NN}$, $f_i(t) = b(t, \varphi_i(x))$, $1 \leq i \leq NN$ and $U_h(t) = (u_1(t), ..., u_{NN}(t)) \in R^{NN}$.

Eqn (3.2.12) can be viewed as an ODE system.

The existence and the uniqueness of the discrete (finite-dimensional) solution U_h of (3.2.10) can be classically shown by diagonalization of the matrices M and A.

Exercise. Detail the matrix coefficients for the typical (linear) example solved by using the RKn time scheme, with $n = 2$.

3.2.4 Complete space-time discretisation

Recall the discrete system to be solved:

$$M\frac{dU_h}{dt}(t) + AU_h(t) = F(t) \quad \text{for } t \in]0,T[\tag{3.2.13}$$

To numerically solve this ODE system, the classical time schemes of ODEs can be employed: forward/backward Euler schemes, θ-scheme, Runge-Kutta (RKn with $n = 2$ or 4 in practice), or the more sophisticated IMplicit EXplicit (IMEX) time schemes.

In what follows, we discretize the time intervall $]0,T[$ into constant time step Δt .
We denote by t_n the n-the time step: $t_n = n\Delta t$, $n = 0, \ldots, N_T$. $T = N_T\Delta t$.
We denote by U_n the approximation of $U_h(t_n)$.

3.2.4.1 Using a Runge-Kutta scheme

Using a Runge-Kutta scheme is very likely the first good option to consider. In practice, RKn with $n = 2$ or 4 is often enough.

Considering the system (3.2.12) the RK2 scheme reads as follows.
Given $U_0 \in \mathbf{R}^{NN}$, compute $U_{n+1} \in \mathbf{R}^{NN}$ solution of:
TO BE DETAILED

3.2.4.2 Using the θ-scheme

Mainly for educational purpose, we here discretize the system (3.2.12) by *using the θ-scheme.* This reads as follows.

Given $U_0 \in \mathbf{R}^{NN}$, compute $U_{n+1} \in \mathbf{R}^{NN}$ solution of:

$$M\frac{U_{n+1} - U_n}{\Delta t} + A\left(\theta U_{n+1} + (1-\theta)U_n\right) = \theta F_{n+1} + (1-\theta)F_n, \quad n = 0, \ldots, N_T. \tag{3.2.14}$$

Recall that the case:

- $\theta = 0$ corresponds to the forward Euler scheme (explicit scheme, order 1 in Δt);
- $\theta = 1$ corresponds to the backward Euler scheme (implicit scheme, order 1 in Δt);
- $\theta = 1/2$ corresponds to the Crank-Nicholson scheme (implicit scheme, order 2 in Δt).

The θ-scheme above can be re-written as:

$$(M + \theta\Delta tA)U_{n+1} = \left(M - (1-\theta)\Delta tA\right)U_n + \Delta t\left(\theta F_{n+1} + (1-\theta)F_n\right), \quad n = 0, \ldots, N_T. \tag{3.2.15}$$

Remark 46. In the non linear PDE case, the nonlinear discrete (finite dimensional) system reads:

$$M\frac{dU_h}{dt}(t) + A(U_h(t)) = F(t) \quad \text{for } t \in]0,T[\tag{3.2.16}$$

with $U \in \mathbf{R}^{NN} \mapsto A(U) \in \mathbf{R}^{NN}$ non linear.
Next, the θ-schema formally reads:

$$MU_{n+1} + \theta\Delta t A(U_{n+1}) = G_n, \quad n = 0, \ldots, N_T. \tag{3.2.17}$$

with $A(U_{n+1})$ to be somehow linearized and $G_n = MU_n - (1-\theta)\Delta t A(U_n) + \Delta t\,(\theta F_{n+1} + (1-\theta)F_n)$.

3.2.4.3 On the stability condition & choice of the time scheme

Let us recall the stability condition of the θ-scheme.

For $1/2 \leq \theta \leq 1$, the θ-scheme is unconditionnaly stable.

The value $\theta = 1/2$ (Crank-Nicholson scheme) is an interesting case since providing a second order scheme.

However, the implicit Euler scheme ($\theta = 1$) may be prefered for 'stiff' problems since it is more robust even though it is less accurate than the Crank–Nicolson scheme.

For $0 \leq \theta < 1/2$, the θ-scheme is stable under the condition: $\max_i \lambda_i \Delta t \leq \frac{2}{1-2\theta}$, with $\{\lambda_i\}_i$ the eigenvalues of the system: $AU_h = \lambda\, MU_h$.

3.2.4.4 Explicit schemes & non linear PDEs

Let us consider the general equation ($\partial_t u + A(u) = f$) based on a non linear differential operator $A(u)$.
Then the form $a(\cdot,\cdot)$ in non linear: $u \mapsto a(u,\cdot)$ non linear.
If using an explicit time scheme, say using the forward Euler time scheme for sake of clarity, on has to solve at each time step:

$$Mu_j^{n+1} = Mu_j^n - \Delta t\; a(u_j^n, \varphi_j) + l^n(\varphi_j) \tag{3.2.18}$$

Consequently, at each time step, one do *not* have to linearize the PDE. Indeed, in this case the non linearity is in the RHS.
However as all explicit schemes, a stability condition must be respected, which is very often (but not always) not tractable.
For more information on the time scheme properties, the reader is invited to consult his favorite course on time-schemes for ODEs and their mathematical analysis (order and stability conditions in particular).

3.2.4.5 Explicit schemes: mass lumping (condensation of mass)

Actually, Eq. (3.2.18) does not enable to *explictly* compute u_j^{n+1} !
Indeed, the mass matrix M has to be inverted.
Fortunatly, a trick to diagonalize M exists. Thus an actual *explicit* scheme is recovered.

Indeed, the classical second order quadrature formulae below to compute the integrals $\int_\Omega \varphi_j(x)\varphi_i(x)\,dx$ is equivalent to diagnolize the matrix M following the so-called mass lumping.

In the case of n-simplex (triangle or tetrahedra) T, by applying the quadrature formula below, the mass matrix becomes diagonal.

$$\int_T \Psi(x)dx \approx \frac{1}{3}|T| \sum_{j=1}^{3(or\ 4)} \Psi(S_{j,T}) \tag{3.2.19}$$

with $S_{j,T}$ the vertices of T.

Indeed, this quadrature formulae consists to lump all the extra diagonal coefficients to the diagonal term. Then, it is called the *mass condensation* or *mass lumping* method.

Resulting temporal accuracy This quadrature formulae is exact for quadratic integrands $\Psi(x)$, with here $\Psi(x) = \varphi_i(x)\varphi_j(x)$. Therefore, this approximation is exact for affine functions $\varphi_i(x)$ thats is P_1-Lagrange FE.

As a consequence, if applying this method with higher-order FE (e.g. $k = 2$), the resulting scheme won't be at the expected maximum order since the temporal term remains order 1...

3.3 Advection term: FE schemes stabilization

Let $u(x)$ be a scalar field, the unknown of the PDE, and $\mathbf{w}(x)$ be a given velocity field.
The advection term $div(\mathbf{w}u)$ models the transport of the quantity u by the vector field $\mathbf{w}$.

If the fluid flow is incompressible, that is $div(\mathbf{w}) = 0$ then $div(\mathbf{w}u) = (\mathbf{w} \cdot \nabla u)$.

Transport terms are of course extremely frequent in fluid flows modeling, but also in wave propagation etc.
The advection terms above are *first order terms leading to numerical instabilities if no particular treatment is applied.*
The stabilization of the advection terms in FE schemes is the topic addressed in the present section.

3.3.1 Equations with an advection term, Peclet number

The (pure) advection equation If modeling the transport of quantity u by the fluid of velocity w (u may be for example the temperature or a chemical specie concentration in the fluid), one has the equation:

$$\partial_t u(x,t) + div(\mathbf{w}\ u)(x,t) = f(x,t) \text{ in } \Omega \times (0,T) \tag{3.3.1}$$

It is a first order PDE. It is a hyperbolic equation, in conservative form.
This equation has to be closed with an I.C. ($u(x,0)$ given) and an adequate B.C. (the quantity must be known at inflow characteristics).

Remark 47. FE methods are naturally suitable for elliptic equations and less naturally for hyperbolic equations. However, they can be employed for hyperbolic equations with stabilization procedures.

Remark 48. Recall that if the flow is *incompressible* then $div(\mathbf{w}) = 0$ and Eqn (3.3.1) simplifies as:

$$\partial_t u(x,t) + \mathbf{w} \cdot \nabla u(x,t) = f(x,t) \tag{3.3.2}$$

The (linear) unsteady advection-diffusion equation If modeling in addition the diffusion of the same quantity u in the media represented by Ω, the equation reads:

$$\left(\partial_t u - div(\mu \nabla u) + div(\mathbf{w}\ u)\right)(x,t) = f(x,t) \text{ in } \Omega \times (0,T) \tag{3.3.3}$$

where μ denotes the diffusivity of u in the media.
The equation above is a linear parabolic equation.

The steady-state advection-diffusion equation In its steady-state version, Eqn (3.3.3) reads:

$$\left(-div(\mu \nabla u) + div(\mathbf{w}\ u)\right)(x) = f(x) \textbf{ in } \boldsymbol{\Omega} \tag{3.3.4}$$

The equation above is a linear elliptic equation.

Existence, uniqueness of the solution of Eqn. (3.3.5).
Please refer to the exercise available on the Moodle course page.

Dimensionless form.
If μ is constant, if the velocity field is incompressible, then the dimensionless form of the equation reads:

$$-\frac{1}{Pe}\ \bar{\Delta}\bar{u}(\bar{x}) + (\bar{\mathbf{w}}\cdot\bar{\nabla}\bar{u})(\bar{x}) = F^* f(\bar{x}) \text{ in } \bar{\Omega} \tag{3.3.5}$$

where the $\bar{}$ denotes the dimensionless quantities and Pe denotes the Peclet number defined by $Pe = \frac{L^* W^*}{\mu}$. The superscript * denotes orders of magnitude.

Exercise. *Show this dimensionless form of the equation. (Moreover, you will clarify the value of F^*).*

The Peclet number Pe measures the ratio of the rate of advection by the flow to the diffusion rate:

$$Pe = \frac{|advection|}{|diffusion|} \tag{3.3.6}$$

In the dimensionless form of the advective-diffusive equation (3.3.5), Pe^{-1} plays the role of the diffusion coefficient.

If $Pe \to 0$ than the model tends to be purely diffusive one.

On the contrary, if $Pe \to +\infty$ then the model tends to be purely advective one: tending to the transport equation $(\mathbf{w}\cdot\nabla u) = f$.

Numerically, one observes that if $\frac{1}{Pe}$ is small (the advection term is dominating) than a standard FE scheme e.g. $\mathbf{P}_k$-Lagrange is unstable: the FE solution $u_h(x)$ blows up !

Indeed, one shows below that standard $\mathbf{P}_k$-Lagrange *FE schemes are centered.*

Stabilizing FE schemes for the advection-diffusion equation is the topic addressed in the next paragraph.

3.3.2 Standard $\mathbf{P}_k$-Lagrange FE schemes = centered schemes

We show below that *on a regular grid, the $\mathbf{P}_1$-Lagrange FE scheme of the advection term* $(\mathbf{w}\cdot\nabla u)$, like in Eq. (3.3.5), *is nothing else than the centered Finite Difference (FD) scheme.*

As a consequence, the FE solution of Eq. (3.3.5) blows up as soon as the Peclet number is large enough.

3.3.2.1 Standard $\mathbf{P}_1$-Lagrange FE scheme of the advection term

Let us consider the advection term $(\mathbf{w}\cdot\nabla u)$, like in Eq. (3.3.5), in 1D. This term simply reads here: $(wu')(x)$.

We set $\Omega =]0,1[$ and we consider a regular grid (mesh): $h = (x_{i+1} - x_i)\ \forall i$.

Let us recall that the i-th $\mathbf{P}_1$-Lagrange basis functions satisfies: $\varphi_i(x_{i-1}) = 0$, $\varphi_i(x_i) = 1$, $\varphi_i(x_{i+1}) = 0$, and reads:

$$\varphi_i(x) = \begin{cases} h^{-1}(x - x_{i-1}) & \text{in } [x_{i-1}, x_i] \\ h^{-1}(x_{i+1} - x) & \text{in } [x_i, x_{i+1}] \\ 0 \quad \text{elsewhere} \end{cases} \tag{3.3.7}$$

Exercise. Show that in 1D the $\mathbf{P}_1$-Lagrange FE scheme of the advective term $(\mathbf{w}\cdot\nabla u)$ in Eq. (3.3.5) equals the centered Finite Difference formula.

Correction. The weak form of this advective term reads: $\int_\Omega w\ \partial_x u(x) v(x)\ dx$.

Let us consider w constant for a sake of simplicity. The corresponding i-th equation term reads as:

$$\begin{aligned}
\int_\Omega w\ \partial_x u(x)\varphi_i(x)\ dx &= w\int_\Omega \varphi(x)\partial_x(\sum_j u_j\varphi_j(x))\ dx \\
&= w\sum_j u_j \int_\Omega \varphi_i\partial_x\varphi_j\ dx \\
&= wu_{i-1}\int_\Omega \varphi_i\partial_x(\varphi_{i-1})\ dx + wu_i\int_\Omega \varphi_i\partial_x(\varphi_i)\ dx + wu_{i+1}\int_\Omega \varphi_i\partial_x(\varphi_{i+1})\ dx
\end{aligned}$$

We have:

$$\partial_x \varphi_i(x) \equiv \varphi_i'(x) = \begin{cases} h^{-1} & \text{in } [x_{i-1}, x_i] \\ -h^{-1} & \text{in } [x_i, x_{i+1}] \\ 0 & \text{elsewhere} \end{cases} \tag{3.3.8}$$

A short calculation shows that:

$$\int_\Omega \varphi_i \partial_x(\varphi_{i-1}) dx = -\frac{1}{2h}; \quad \int_\Omega \varphi_i \partial_x(\varphi_i) dx = 0; \quad \int_\Omega \varphi_i \partial_x(\varphi_{i+1}) dx = \frac{1}{2h} \tag{3.3.9}$$

Therefore:

$$\int_\Omega w\ \partial_x u(x) \varphi_i(x)\ dx = w \left(\frac{u_{i+1} - u_{i-1}}{2} \right) \tag{3.3.10}$$

Hence the result.

This result remains true in nD and for non constant velocity fields.

3.3.2.2 Explicit solutions in the 1D case & unstabilities

Let us consider now the following 1D steady-state advection-diffusion BVP:

$$\begin{cases} -\epsilon\ u''(x) + w\ u'(x) = 0 & x \in]0,1[\\ u(0) = 0 & u(1) = 1 \end{cases} \tag{3.3.11}$$

In the equation above, for $|w| \approx 1$, $Pe^{-1} \approx \varepsilon$, see (3.3.5).
The parameter ε is supposed to be small, equivalently Pe is supposed to be large.

Exercice. The velocity field w is supposed to be constant, $w > 0$.
1) Verify that the exact solution of the BVP above reads:

$$u(x) = c \left[exp(\frac{wx}{\epsilon}) - 1 \right] \text{ with } c = \left[exp(\frac{w}{\epsilon}) - 1 \right]^{-1}. \tag{3.3.12}$$

2) Deduce that close to the outflow boundary at $x = 1$, the solution u presents a *boundary layer* i.e. a stiff gradient.

Exercice.
1) Verify that the solution of the *centered* FD scheme, with $w = 1$, reads:

$$u_i = \frac{\left(\frac{1 - Pe_h}{1 + Pe_h} \right)^i - 1}{\left(\frac{1 - Pe_h}{1 + Pe_h} \right)^{N+1} - 1} \qquad i = 1, \ldots, (N+1) \tag{3.3.13}$$

with Pe_h the *numerical Peclet* number defined by:

$$Pe_h = \frac{1}{2\epsilon} w\ h \tag{3.3.14}$$

2) Deduce the scheme behavior if $Pe_h > 1$.

Remedy: to refine or to stabilize ? The natural solution to this unstability issue is to refine the mesh i.e. to reduce the grid size h. However, for numerous real-world problems, this is not possible. Indeed, ...

Numerical example. Let us consider an air flow around a vehicle engine at 100 km/h, then $Pe = \frac{L^* W^*}{\mu} \sim 10^5$.
Stabilizing the numerical scheme by refining the mesh would mean to set: $h \approx 10^{-5}$ m....
In such a context, the remedy consists to stabilize the FE scheme.

On the upwinding technique.

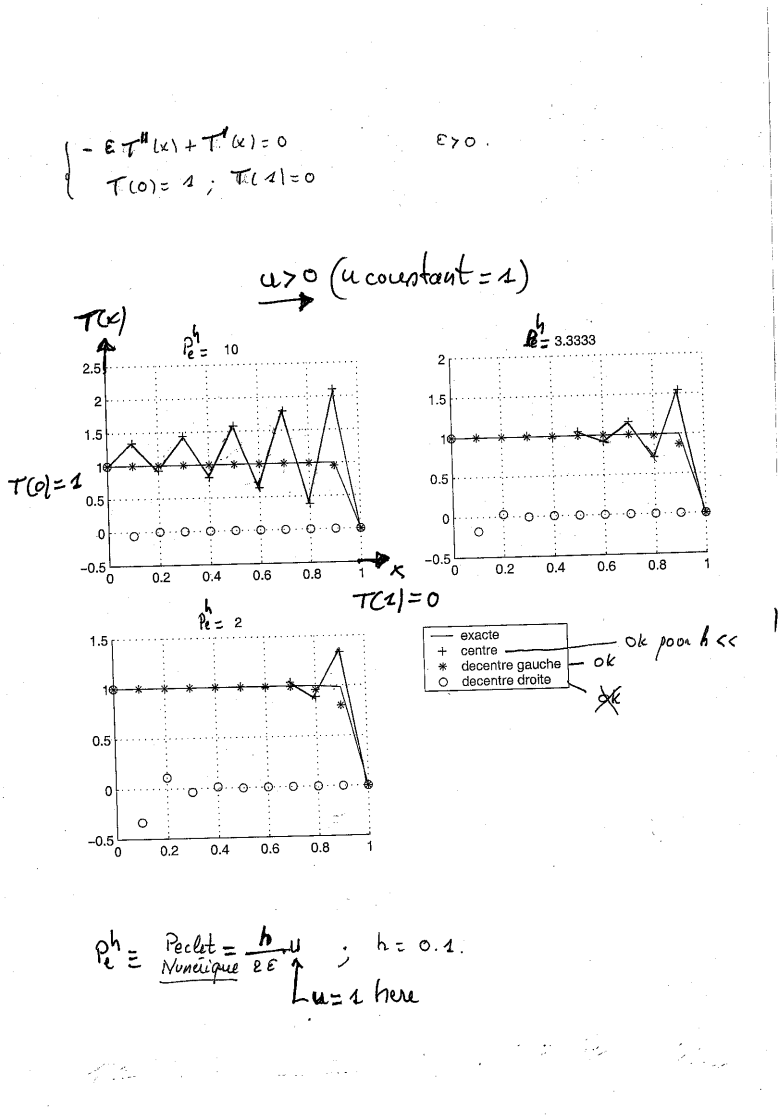


Figure 3.3.1: The centered FD solution oscillates if the *numerical Peclet* Pe^h is greater than 1.

The stabilization of the numerical solution u_h can be simply done by *upwinding the advective term* as: $(w \frac{u_i - u_{i-1}}{h})$ for $w > 0$ and $(w \frac{u_{i+1} - u_i}{h})$ for $w < 0$.

However, as it will be shown in next section, such a simple upwinding of the gradient term ∇u introduces non-expected numerical diffusion...

3.3.2.3 Equivalent equations & numerical diffusion (2D illustration)

Let us consider the (pure) advection equation (3.3.1) in 2D. We denote the velocity components as $\mathbf{w} = (w_1, w_2)$.

The $\mathbf{P}_1$-Lagrange FE discretization on a regular grid reads as the following (potentially unstable) centered scheme:

$$\frac{(u_{ij}^{n+1} - u_{ij}^n)}{\Delta t} + w_{1,ij}^n \frac{(u_{i+1j}^n - u_{i-1j}^n)}{2\Delta x} + w_{2,ij}^n \frac{(u_{ij+1}^n - u_{ij-1}^n)}{2\Delta y} = a \qquad \forall (i,j) \tag{3.3.15}$$

In the following and for a sake of simplicity, w_k are assumed to be strictly positive.

Exercise.

1) Show that the centered FD formula of the advective term $(\mathbf{w} \cdot \nabla u)$ equals the (adequate) uncentered one minus a

diffusive term in $\mathcal{O}(\Delta x)$.
2) Deduce that the $\mathbf{P}_1$-Lagrange FE discretization (on a regular grid) (3.3.15) is equivalent to an uncentered scheme with a diffusive term in $\mathcal{O}(\Delta x)$.

Correction.
1) It is easy to verify that:

$$w^n_{1,ij}\frac{(u^n_{i+1j}-u^n_{i-1j})}{2\Delta x} = w^n_{1,ij}\frac{(u^n_{i+1j}-u^n_{ij})}{\Delta x} - \left(\frac{\Delta x}{2}w^n_{1,ij}\right)\frac{(u^n_{i+1j}-2u^n_{ij}+u^n_{i-1j})}{\Delta x^2} \tag{3.3.16}$$

2) As a consequence, the $\mathbf{P}_1$-Lagrange FE discretization (on a regular grid) (3.3.15) reads:

$$\frac{(u^{n+1}_{ij}-u^n_{ij})}{\Delta t} + w^n_{1,ij}\frac{(u^n_{i+1j}-u^n_{ij})}{\Delta x} + w^n_{2,ij}\frac{(u^n_{ij+1}-u^n_{ij})}{\Delta y}$$
$$-\left(w^n_{1,ij}\frac{\Delta x}{2}\right)\frac{(u^n_{i+1j}-2u^n_{ij}+u^n_{i-1j})}{\Delta x^2} - \left(w^n_{2,ij}\frac{\Delta y}{2}\right)\frac{(u^n_{ij+1}-2u^n_{ij}+u^n_{ij-1})}{\Delta y^2} = a \tag{3.3.17}$$

As a consequence, on a regular grid, the $\mathbf{P}_1$-Lagrange FE scheme of the advection equation equals the upwinded FD scheme with non-physical ("artificial") diffusion.
The diffusion coefficient reads as: [velocity $\times$ $(\Delta x, \Delta y)/2$].

Therefore, *the centered schemes (which are potentially unstable) are equivalent to the corresponding upwinded schemes with artificial diffusion, whose the diffusivity coefficient* $\mu_h \sim \frac{1}{2}|\mathbf{w}|h$.

Recall that these upwinded schemes are unconditionally stable, on contrary to the centered schemes.

3.3.3 Stabilization techniques: SD, SUPG, GLS

Let us consider here the following steady-state linear advection-diffusion equation:

$$-\varepsilon\ \triangle u(x) + \mathbf{w}\cdot\nabla u(x) = f(x) \tag{3.3.18}$$

with adequate boundary conditions.
The diffusion coefficient ε is supposed to be small, $\varepsilon > 0$. In the dimensionless equation, $\varepsilon = Pe^{-1}$ therefore Pe large.

(Dissatisfaying) isotropic diffusion

A straightforward extension of the artificial diffusion term derived in the previous section is to add the term:

$$-h\,\|\mathbf{w}\|_\infty \triangle u \tag{3.3.19}$$

The latter stabilizes the FE scheme as expected. However, it does it in all directions. Therefore it introduces unnecessary (and unphysical) cross-wind diffusion.

The stabilization of FE schemes has to be done consistantly with respect to the physics.

Streamline Diffusion (SD) method

The principle of SD is to introduce artificial diffusion along the streamlines only.
This is done by adding the following anisotropic diffusion matrix $\left(\frac{h}{2\epsilon^2}\mathbf{w}\mathbf{w}^T\right)$.
Thus the diffusion matrix to be considered in the equation becomes:

$$\varepsilon I \leftarrow \left(\varepsilon I + \alpha\frac{h}{2\epsilon^2}\mathbf{w}\mathbf{w}^T\right) \tag{3.3.20}$$

with α a weight coefficient to be set, $\alpha \lesssim 1$.
Recall that: $\epsilon\ \triangle u = div(\varepsilon I\ \nabla u)$.

Lemma 49. *Let us denote by D the Streamline Diffusion (SD) matrix: $D = \frac{h}{2\epsilon^2}\mathbf{w}\mathbf{w}^T$.*

For any vector $d \in \mathbf{R}^n$ (n the dimension geometric space), the product $(D \cdot d)$ is co-linear to the streamline flow.

Proof. Let us show the result in 2d ($n = 2$). We set: $\mathbf{w} = (w_1, w_2)$. We have $\mathbf{w}^T = (-w_2, +w_1)$ and:

$$D = \mathbf{w}.\mathbf{w}^T = \begin{pmatrix} w_1^2 & w_1 w_2 \\ w_1 w_2 & w_2^2 \end{pmatrix}$$

In the coordinate system $(\mathbf{w}, \mathbf{w}^\perp)$, D has to be of the following form: $\begin{bmatrix} \bullet & 0 \\ 0 & 0 \end{bmatrix} \equiv D_{ww^T}$.

We denote by P the change of variable matrix. We have: $D_{ww^T} = P^{-1}DP$. P has to satisfy:

$$Pe_1 = \mathbf{w} \text{ and } Pe_2 = \mathbf{w}^T \tag{3.3.21}$$

We set: $P = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$. From (3.3.21) it follows: $P = \begin{vmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{vmatrix}$.

We have: $PD_{ww^T} = \begin{vmatrix} \tilde{a}w_1 & 0 \\ \tilde{a}w_2 & 0 \end{vmatrix}$. And: $DP = \begin{vmatrix} w_1^3 + w_1 w_2^2 & -w_1^2 w_2 + w_1^2 w_2^2 \\ w_1^2 w_2 + w_2^3 & -w_1 w_2^2 + w_2^2 w_1 \end{vmatrix} = \begin{vmatrix} w_1(w_1^2 + w_2^2) & 0 \\ w_2(w_1^2 + w_2^2) & 0 \end{vmatrix}$.

Therefore: $PD_{ww^T} = DP$ with $\tilde{a} = (w_1^2 + w_2^2) = \|\mathbf{w}\|_2^2$. And: $D_{ww^T} = P^{-1}DP = \begin{vmatrix} \|\mathbf{w}\|_2^2 & 0 \\ 0 & 0 \end{vmatrix}$.

In the local coordinate system $(\mathbf{w}, \mathbf{w}^\perp)$, the diffusion matrix has the following form: $\begin{vmatrix} \|\mathbf{w}\|_2^2 & 0 \\ 0 & 0 \end{vmatrix}$.

It is a Streamline Diffusion.

SD method is efficient since it stabilizes the FE scheme by introducing artificial diffusion along the streamlines only. However the correction made to the (discrete) variational form is in $\mathcal{O}(h)$, see (3.3.20). Therefore the resulting FE scheme is order 1 at most independently on the original FE order k, e.g. $k = 2$...

This is the drawback of SD method. This drawback may be circumvented by employing the SUPG or GLS methods below.

Streamline Upwind Petrov-Galerkin (SUPG) and Galerkin Least-Square (GLS) methods*

** This is a to go further paragraph*

The standard FE method (conforming FE i.e. an internal approximation) to solve (3.3.18) consists to find $u_h \in V_{0h}$ such that:

$$a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_{0h} \tag{3.3.22}$$

That is the function tests belongs to the same function space than the solution (here V_{0h}).

The *SD method* consists to introduce the additional term, see (3.3.20): $\tau_{v_h}(T_h) = \int_\Omega \left(\alpha \frac{h}{2\epsilon^2}\mathbf{w}\mathbf{w}^T\right) \nabla u_h \cdot \nabla v_h \, dx$.

This correction provides a consistent scheme in the sense: $\lim_{h\to 0} \tau(u_h) = 0$; but not in the classical (strong) sense $\tau(u_h) = 0 \quad \forall h > 0$. Typically, the orthogonality property (2.1.9) does not hold anymore.

The *Streamline Upwind Petrov-Galerkin (SUPG)* method and the *Galerkin-Least-Square (GLS)* method are different. They consist to consider *modified test functions*. As a consequence, the test functions belong to a different space than the solution space V_{0h}: it is so-called Petrov-Galerkin approximation.

Considering the equation (3.3.18), the modified weak form reads as:

$$a(u_h, v_h) + \; c(u_h, v_h) = l(v_h) + \; k(v_h) \quad \forall v_h \in V_{0h} \tag{3.3.23}$$

with:

$$c(u_h, v_h) = \sum_K \int_K \alpha_K(h) \; (-\varepsilon \Delta u_h + \mathbf{w} \cdot \nabla u_h) \; z_h(v_h) \, dx \tag{3.3.24}$$

$$k(v_h) = \sum_K \int_K \alpha_K(h) \; f \quad z_h(v_h) \, dx \tag{3.3.25}$$

$\alpha_K(h)$ is the local weight coefficient detailed below.
The test function $z_h(v_h)$ is defined as:

$$\begin{cases} z_h(v_h) = (\mathbf{w}\cdot\nabla v_h) & \text{in SUPG method} \\ z_h(v_h) = (\mathbf{w}\cdot\nabla v_h) - \varepsilon\Delta v_h & \text{in LS method} \end{cases} \tag{3.3.26}$$

The additional terms $c(\cdot,\cdot)$ and $k(\cdot)$ in (3.3.23) consist to introduce diffusion along the streamlines.
Let us define the *residual* function of (3.3.18):

$$r_h(u_h) = (-\varepsilon\Delta u_h + \mathbf{w}\cdot\nabla u_h - f) \tag{3.3.27}$$

The additional terms introduced in (3.3.23) satisfy:

$$c(u_h, v_h) - k(v_h) = \sum_K \int_K \alpha_K(h)\ r_h(u_h)\ z_h(v_h)\ dx \tag{3.3.28}$$

The stability parameter $\alpha_K(h)$ is locally defined depending on the flow regime as:

$$\alpha_K(h) = \delta\frac{h_K}{2\|\mathbf{w}\|_2} \times \begin{cases} Pe_K & \text{if } 0 \le Pe_K < 1 \\ 1 & \text{if } Pe_K \ge 1 \end{cases} \tag{3.3.29}$$

with δ a coefficient to be set, $\delta \sim 1$, and Pe_K the local numerical Peclet number:

$$Pe_K = h_K\frac{\|\mathbf{w}\|_2}{2\varepsilon} \tag{3.3.30}$$

For more details, the reader may refer to[1]or e.g. to the short note review [2]and references therein.

The resulting modified FE schemes above are well-posed; moreover they are *strongly consistent.*
Indeed we have:

$$a(u, v_h) + \ c(u, v_h) = l(v_h) + \ k(v_h) \ \ \forall v_h \in V_{0h}$$

with $u(x) \in V_0$ the (unique) exact solution.
Moreover this approach enables to *preserve higher order interpolations.*

We have

Theorem 50. *Let us assume that: a) the (unique) exact solution $u(x)$ of the advection-diffusion equation (3.3.18) satisfies: $u \in V_0 \cap H^{k+1}(\Omega)$; b) $\mathbf{P}_k$-Lagrange FE are employed.*
Then the solution of the stabilized FE scheme (3.3.23)(3.3.26) converges to the exact solution u of (3.3.18) as follows:

$$\varepsilon\|\nabla(u_h - u)\|_0^2 + \|\alpha^{1/2}\ \mathbf{w}\cdot\nabla(u_h - u)\|_0^2 \tag{3.3.31}$$

$$\le C\ \sum_K h^{2k}\ |u|_{k+1,K}^2\ [H(Pe_K - 1)\ h\|\mathbf{w}\| - H(1 - Pe_K)\ \varepsilon] \tag{3.3.32}$$

with $H(\cdot)$ the Heaviside function.

Proof. See [L. P. Franca, S. L. Frey, and T. J. R. Hughes ' 1992].

Observe that the upper-bound error depends on the regime: advection dominated ($Pe_K > 1$) or diffusion dominated ($Pe_K < 1$).

Remark 51. A few other stabilizing technique exist in particular those consisting to enrich the FE space V_h by adding a *cubic bubble function* in each element K. For details the reader may refer to the numerical analysis book [3]

[1] L. P. Franca, S. L. Frey, and T. J. R. Hughes, Stabilized finite element methods: I. application to the advective-diffusive model, Comput. Methods Appl. Mech. Engrg., 95, 253–276 (1992).

[2] Franca, Leopoldo P., G. Hauke, and A. Masud. "Stabilized finite element methods." International Center for Numerical Methods in Engineering (CIMNE), Barcelona, 2004.

[3] Quarteroni, Alfio, and Alberto Valli. Numerical approximation of partial differential equations. Vol. 23. Springer, 2008.

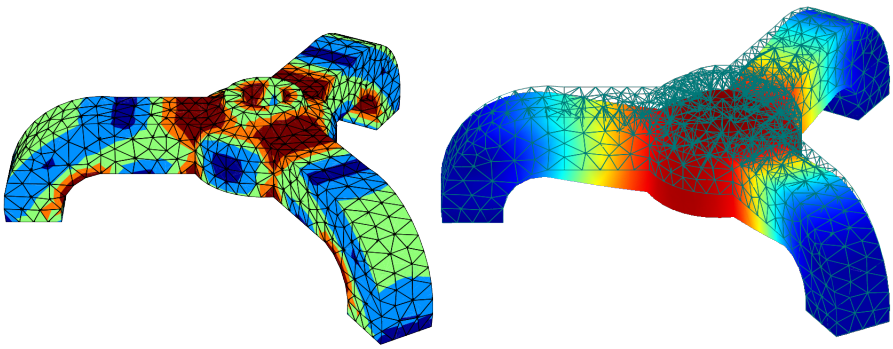


Figure 3.4.1: FE methods are particularly well adapted and popular in structural mechanics. Here a computation of the von Mises Yield criterion. Image source:...

3.3.4 On conservative numerical methods: Finite Volume (FV) and Discontinuous Galerkin (DG)

Finite Volume (FV) methods are the most adequate methods for conservative systems (or equations).

Recall that a conservative equation is an equation which reads as:

$$div(G(u))(x) = f(x) \text{ in } \Omega \tag{3.3.33}$$

where $G(.)$ is the (physical) flux (e.g. a mass flux). This flux may be linear (in u) or not.
A naturally *conservative* numerical method is the FV method.
Let us mention the *Discontinuous Galerkin (DG)* methods too. DG schemes combine the features of the FE and the FV schemes. DG methods are particularly interesting for dominant first-order terms. They enable to built up $\mathbf{P}_k$-discontinuous approximations.

3.4 The linear elasticity system

The FE method has been presented up to now for scalar PDEs. Its extension to the *system of linear elasticity* does not introduce big issues.

Section to be completed.

You may consult the short supplementary notes available on the INSA Moodle platform or your structural mechanics course, or the aforementionned books.

3.5 A-posteriori error estimations and mesh refinement*

**This is a to go further section.*

3.5.1 Introduction

3.5.1.1 The BVP context

Let us consider a scalar linear second order BVP. The weak formulation (3.5.1) reads:

$$\begin{cases} \text{Find } u \in V \text{ such that:} \\ a(u,v) = b(v) \qquad \forall v \in V \end{cases} \tag{3.5.1}$$

where the bilinear form $a(.,.)$ and the linear form $l(.)$ satisfy the conditions of the Lax-Milgram theory.
Let us supposed that the unique solution is regular in the sense $u \in H^{1+k}(\Omega)$ for a given k, $k \geq 1$.
Let us consider a P_k-Lagrange FE approximation in V_h, $k \geq 1$.
V_h is a finite dimensional internal approximation of V (see Definition 23).

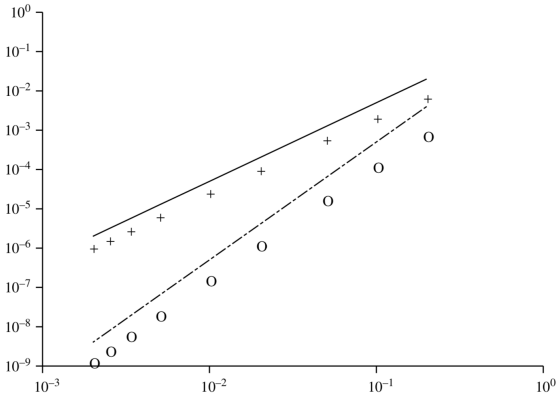


Figure 6.4. Case of a regular solution: example (6.29). Discrete H^1 norm of the error as a function of the size h of the mesh (the crosses correspond to $\mathbb{P}_1$ finite elements, the circles to $\mathbb{P}_2$ finite elements, the lines are the graphs of $h \to h^2$ and $h \to h^3$).

Convergence curve obtained from an explicit solution: the theoritical rate is recovered. Image source: [**?**].

3.5.1.2 The general a-priori estimation

We then have the *a-priori error estimation*, see Theorem 37:

$$\|u - u_h\|_{H^1} \leq c\ h^k\ \|u\|_{H^{k+1}} \tag{3.5.2}$$

with the constant c independent of u and h, $c > 0$.

Note that these are asymptotic estimations where one do not knows the value of the constant c...

However given a mesh resolution h, equivalently given a number of nodes NN (the polynomial order k is fixed), this shows that one can reach some accuracy of the FE solution.
Typically, one has the following convergence behavior, see Fig. 2.4.3 or the Fig. below.

Mesh adaptation can yield more accurate results with similar computational resources, on one hand by refining the mesh where the solution sharply varies, on the other hand by relaxing the mesh where the solution is gently varies, see e.g. Fig. 3.5.2.4.

3.5.2 A-posteriori estimators: basic properties

3.5.2.1 Desired properties of an a-posteriori error estimator

In next sections, methods to compute *a-posteriori error estimations* enabling to built refined meshes are presented.

- A-posterior error estimations differ from a-priori error estimations in that the upper bound does not depend on $\|u\|_{H^{k+1}}$.
- A-posteriori error estimators provide local information on the error of the computed FE solution, from the data of the PDE.

The perfect FE error estimator would guarantee a maximum error value at every node. However such estimator do not exist yet !

Given the exact BVP solution u, given the FE approximlation u_h, we denote the error estimator by $e(h, u_h, f)$. The exact solution u is here represented by the RHS f.

Algorithm 3.2 Mesh adaptation algorithm based on a local error estimator $e_K(u_h, f)$.

- Given a mesh, solve the BVP $A(u_h) = f$: the FE solution u_h is obtained in Ω, $\Omega = \cup_{K\in\mathcal{T}_h} K$.
- Compute the local error indicator $e_K(u_h, f)$ for each mesh element K, $K \in \mathcal{T}_h$.
  - If $e_K(u_h, f)$ is greater than a prescribed tolerance, then the mesh is locally refined, e.g. the element K is split.
  - If $e_K(u_h, f)$ is lower than another tolerance, then the mesh can be locally unrefined: the element K is recombined with its neighbors.

Definition 52. A function $e(h, u_h, f)$. is said to be an *a-posteriori error estimator* if:

$$\|u - u_h\|_V \leq e(h, u_h, f) \tag{3.5.3}$$

This is the *reliability property*: the estimator controls the error in the energy norm.
Moreover, the estimator has to be locally computable as:

$$e(h, u_h, f) = \left(\sum_{K\in\mathcal{T}_h} \left(e_K(u_h, f)\right)^2 \right)^{1/2} \tag{3.5.4}$$

The term $e_K(u_h, f)$ denotes the *local error indicator*.

3.5.2.2 On the control of the local errors

Let us assume that one can "localize" the norm computation, that is:

$$\|\cdot\|_V^2 = \sum_{K\in\mathcal{T}_h} \|\cdot\|_{V,K}^2 \tag{3.5.5}$$

This is the case for the classical Sobolev norms.
Then, one would like that the local estimator satisfies the following double inequality:

$$\forall h,\ \forall K \in \mathcal{T}_h,\ c_1 e_K(u_h, f) \leq \|u - u_h\|_{V,K} \leq c_2 e(u_h, f) \tag{3.5.6}$$

where $c_\sqcap$ are constants independent of the mesh.
If this inequality is satisfied then the local estimator $e_K(u_h, f)$ is equivalent to the local error $\|u - u_h\|_{V,K}$.
Unfortunately, such inequalities are generally not satisfied or not established.

Instead, one generally manages to obtain the following two types of inequalities only:

- A global upper bound:

$$\forall h,\quad \sum_{K\in\mathcal{T}_h} \|u - u_h\|_{V,K}^2 \leq c_2 \sum_{K\in\mathcal{T}_h} \left(e_K(u_h, f)\right)^2 \tag{3.5.7}$$

- A perturbed lower bound:

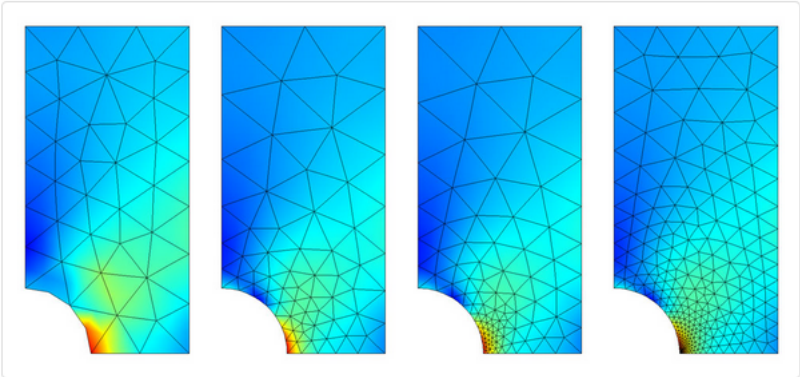
$$\forall h,\ \forall K \in \mathcal{T}_h,\ c_1 e_K(u_h, f) \leq \|u - u_h\|_{V,\Delta_K} + \Pi(h_K, \Delta_K, f) \tag{3.5.8}$$

where $\Delta_K$is a patch of elements around K.
The term $\Pi(h_K, \Delta_K, f)$ is either negligible or of the same order as $\|u - u_h\|_{V,\Delta_K}$.

3.5.2.3 The mesh adaptation algorithm

Given a local error indicator $e_K(u_h, f)$, the mesh refinement strategy is simply as follows (Algorithm 2.1).

This process can be repeated, then this provides a mesh adaptation strategy: the "automatic mesh refinement" procedure, see Algo. 3.2.



From Left to Right: The FE solution on a regular mesh, next (from (L) to (R)), on more and more refined meshes.
Image source: Comsol multi-physics webpage.
Also, note that a mesh refinement strategy enables to approximate the solution more accurately than on a regular mesh with the similar computational resources.

3.5.2.4 The few types of estimators

Few types of a-posteriori error estimator exist. They may be classified as follows:

- Residual-based error estimators,
- Goal-oriented error estimators based on a dual formulation,
- Hierarchical techniques.

We briefly present in next sections these three first types of a-posteriori error estimators.

The presentation proposed in this chapter follows in good part the excellent book [**?**]. The reader may consult e.g. the synthetic review [4] too.

Before addressing these three types of a-posteriori error estimators, we first present a pseudo-empirical method based on the estimation of the Hessian of the FE solution u_h.

3.5.3 A first method based on the interpolation error & anisitropic mesh adaptativity

We derive here a pseudo-empirical error indicator based on an estimation of the Hessian. This method is not based on an actual a-posteriori error estimator. It is based on an *error indicator* inspired by the following *interpolation error estimations.*

3.5.3.1 Interpolation errors in the case of linear elements

Let us consider the case of P_1-Lagrange FE (linear elements). In this case, the a-priori FE error estimation (3.5.2), deriving from Cea's lemma (26) and the interpolation error estimation, reads:

$$\|u - u_h\|_{H^1} \leq c_0 \, \|u - \pi_h(u)\|_V \leq c \; h \; \|u\|_{H^2} \tag{3.5.9}$$

with u the exact solution (u necessarily regular in the sense $u \in H^2(\Omega)$), and $\pi_h(v)$ the Lagrange interpolation operator.

It turns out that in 2D triangular meshes (actually in 3D tetrahedra meshes too), estimations of the interpolation error $\|u(x,y) - \pi_h(u)(x,y)\|$ can be detailed.

[4] Grätsch, Thomas, and Klaus-Jürgen Bathe. "A posteriori error estimation techniques in practical finite element analysis." Computers & structures 83.4-5 (2005): 235-265.

Proposition 53. *In 2D triangular meshes, we have the following interpolation error estimations:*

$$\begin{cases} |u(x,y) - \pi_h(u)(x,y)| & \leq \frac{1}{2} h_{max}^2 \sup_{(x,y)\in\Omega} \|D^2 u(x,y)\| \\ |\nabla u(x,y) - \nabla \pi_h(u)(x,y)| & \leq 3 \frac{h_{max}}{\sin(\theta_{min})} \sup_{(x,y)\in\Omega} \|D^2 u(x,y)\| \end{cases} \tag{3.5.10}$$

where $D^2(\cdot)$ denotes the Hessian, $\|D^2 u\|$ its spectral norm, h_{max} the greatest triangle edge and θ_{min} the lowest triangle angle.

Moreover, we have the following local interpolation error estimation too:

$$\forall K \in \mathcal{T}_h, \quad |u(x,y) - \pi_h(u)(x,y)|_K \leq c \sup_{E\in\partial K} \sup_{(x,y)\in\Omega} < e, D^2 u(x,y) e > \tag{3.5.11}$$

with ∂K the set of edges of K, e an edge vector linking the two vertices, and the constant c depending on the element geometry type (triangle, tetrahedra).

The proof of (3.5.10) can be found of e.g. in **[?]**. Those of (3.5.11) can be found e.g. in [Ainsworth, Oden, book, 2011]

Let us mention a few consequences of these estimations.

On the triangle-tetrahedra quality

- The dominating error is related to the gradient quantity, see (3.5.10)(b).
  Recall that in physics-mechanics (in the large sense), the gradients represent the fluxes, the strains. It is therefore important to accurately compute them.
- The constant value c in the general a-priori estimation, see (3.5.2), is unknown. This is not the case for linear elements.
  And this shows that in the present case the gradient estimation bound is minimal where $\frac{h_{max}}{\sin(\theta_{min})}$ is minimal, see (3.5.10)(b).

  This implis that in an uniform mesh, the optimal triangles shape is the equilateral one.
  This is not true anymore for non-isotropic equations where the solution varies greatly in a given direction, e.g. in a boundary layer in an advection-diffusion problem. In this case, the optimal triangle shapes will be the ones which maximizes its surface $|K|$ in an ellipse defined by the eigenvalues of the Hessian matrix.

On the triangle - tetrahedra anisotropy The a-priori error estimation relying on (3.5.11) can be summarized as follows:

$$\text{FE error } \|u - u_h\|_V \leq cst \text{ Interpolation error } \|u - \pi_h(u)\| \leq cst \text{ Square of edge lengths in metric } D^2(u) \tag{3.5.12}$$

The local interpolation errors are proportional to the square of the longest edge of K, *provided that the edge is measured in the metric determined by the Hessian $D^2(u)$.*

This result seems natural: where the second derivatives is large one wants to decrease the edge lengths; conversely, where the second derivatives are small, longer edge lengths may be employed.

Based on the estimation (3.5.11), the mesh adaptativity strategy will consist *to build triangles-tetrahedra equilateral for the $D^2(u_h)$-based metric.* In practice, this requires to compute the eigenvalues of the Hessian.

3.5.3.2 Refinement based on the interpolation error and the Hessian eigenvalues

Basic principle For linear triangular elements, the interpolation error estimation (3.5.10) suggests to locally refine the mesh where the Hessian norm $\|D^2(u)\|$ is large. However the Hessian of the exact solution is not known. Then, the basic principle here consists to approximate $\|D^2(u)\|$ by the Hessian of the FE solution $\|D^2(u_h)\|$.

Estimation of the Hessian $\|D^2(u_h)\|$ for P_k-Lagrange elements First, let us observe that for P_k-Lagrange FE, the gradient are piecewise $P_{(k-1)}$, but they are non continuous between two elements. As a consequence, a direct computation of the Hessian $D^2(u_h)$ may be an issue. Then, a way to estimate $D^2(u_h)$ is to do it in the weak sense as follows.

Let us denote by $H_{ij}(u_h) = \partial^2_{ij} u_h$, $1 \leq i, j \leq n$.

To compute $H_{ij}(u_h)$, we solve the equation:

$$\int_\Omega H_{ij}(u_h) v_h(x)\ dx = -\int_\Omega \partial_i u_h \partial_j v_h(x)\ dx + \int_{\partial\Omega} \partial_i u_h v_h n_j(x)\ ds \qquad \forall v_h \in V_h \tag{3.5.13}$$

Therefore:

$$\sum_{K\in\mathcal{T}_h} \int_K H^l_{ij}(u_h)\varphi_l(x)\ dx = -\int_\Omega \partial_i u_h \partial_j \varphi_l(x)\ dx + \int_{\partial\Omega} \partial_i u_h \varphi_l n_j(x)\ ds \qquad \forall l,\ 1 \leq l \leq NN \tag{3.5.14}$$

By applying the mass lumping technique, we obtain:

$$H^l_{ij}(u_h) \approx \frac{1}{\left(\sum_K \int_K \varphi_l(x)\ dx\right)} \left(-\int_\Omega \partial_i u_h \partial_j \varphi_l(x)\ dx + \int_{\partial\Omega} \partial_i u_h \varphi_l n_j(x)\ ds\right) \qquad \forall l,\ 1 \leq l \leq NN \tag{3.5.15}$$

The term $H^l_{ij}(u_h)$ represents an approximation of the Hessian of u_h at node l.

Local metric and anistropic adaptativity Inspired by the estimation (3.5.10), we define the following *empirical error indicator* (abusely denoted by $e_K(u_h, f)$):

$$e_K(u_h, f) = max_{E\in\partial K} max_{x_{node}\in K} (< e, H(u_h)(x_{node})e >) \tag{3.5.16}$$

Observe that the equation $< e, D^2u(x_{node})e >= 1$ represents an ellipsoid in nD (an ellipse in 2D), whose the n axes are the eigenvectors of $H(u_h)$ and the semi-axes lengths equal $\frac{1}{\sqrt{\lambda_i}}$, λ_i the eigenvalues, $i = 1, \cdots, n$.
Make a draw.

In practice, the eigenvalues of $H(u_h)$ are obtained by computing the zeros of an order 2 (quadratic) or order 3 polynomial.

Given the empirical error indicator $e_K(u_h, f)$ above, one next applies the mesh adaptivity process as previously described, see Algo. 3.2.

Recall that the present procedure does not rely on an actual a-posteriori error estimator.

Indeed, $e_K(u_h, f)$ is here an "uncontrolled" approximation of the Hessian, and it does not satisfy the fundamental estimation (3.5.3).

Moreover this pseudo-empirical error indicator is inspired from the interpolation estimation (3.5.10) valid for linear elements. For higher-order elements, the estimations are more difficult to establish.
However, this method provides an easy and relatively efficient tool for many real-life problems.

Finally, in practice it is convenient to define the even more simpler error indicator defined by:

$$e_K(u_h, f) = e_{K,max}\ \lambda_{max}(H(u_h)) \tag{3.5.17}$$

3.5.4 Residual-based error estimator

In this section, we present an actual a-posterior error estimator. It is based on the equation residual.

3.5.4.1 The toy BVP

Let us consider a linear BVP whose the weak formulation reads as usual:

$$a(u, v) = b(v) \qquad \forall v \in V = H^1_0(\Omega) \tag{3.5.18}$$

where the Lax-Milgram theorem conditions are satisfied.

The FE approximation is P_k-Lagrange, $k \geq 1$. The discrete weak formulation reads as usual.

V_h denotes the corresponding internal approximation of V. $V_h \subset V$.

For the sake of simplicity, we detail below the residual-based error estimator for the following linear BVP:

$$\begin{cases} -\Delta u(x) + u(x) &= f(x) \text{ in } \Omega \\ u(x) &= 0 \text{ on } \partial\Omega \end{cases} \tag{3.5.19}$$

3.5.4.2 The residual

Let us define the *residual* as follows. $\forall v \in V$,

$$r(u_h, f; v) = (f, v) - a(u_h, v) \tag{3.5.20}$$
$$= a(u - u_h, v) \tag{3.5.21}$$

For the present BVP, we have: $r(u_h, f; v) = (\nabla(u - u_h), \nabla v) + ((u - u_h), v)$.

Let us recall the Galerkin orthogonality relation (37). Then, we have:

$$r(u_h, f; v_h) = a(u - u_h, v_h) = 0 \qquad \forall v_h \in V_h \tag{3.5.22}$$

3.5.4.3 The fundamental estimation

Theorem 54. *Let us consider the linear problem (3.5.18) solved by the P_k-Lagrange FEM, $k \geq 1$.*
Then, it exist a constant c such that:

$$\forall h, \quad \|u - u_h\|_{1,\Omega} \leq c \left(\sum_{K \in \mathcal{T}_h} \left(e_K(u_h, f)\right)^2 \right)^{1/2} \tag{3.5.23}$$

In the case of the particular BVP (3.5.19), the local error indicator reads:

$$e_K(u_h, f) = h_K \|f + \Delta u_h + u_h\|_{0,K} + \frac{1}{2} \sum_{F \in \mathcal{F}_K} h_F^{1/2} \|[\partial_{n_K} u_h]\|_{0,F} \tag{3.5.24}$$

F denotes the interior faces (edges in 2d) of the element K and $\mathcal{F}_K$ denotes the set of faces of K which are not on $\partial\Omega$.

Proof. Let π_h be the P_k-Lagrange interpolation operator, see (2.4.1). We have: $\pi_h : v \in V \to \pi_h(v) \in V_h$.

Let us develop the expression of $a(u - u_h, v - v_h)$. Using (3.5.22), we obtain:

$$a(u - u_h, v - \pi_h v) = r(u_h, f; v - \pi_h v) = r(u_h, f; v) \tag{3.5.25}$$

due to the orthogonality relation above.
Therefore:

$$r(u_h, f; v) = (f, v - \pi_h v) - (\nabla u_h, \nabla(v - \pi_h v)) - (u_h, v - \pi_h v) \tag{3.5.26}$$

By decomposing the integrals on each cell K and by applying the Green formula, we obtain:

$$r(u_h, f; v) = \sum_{K \in \mathcal{T}_h} (f + \Delta u_h - u_h, v - \pi_h v)_K - \sum_{F \in \mathcal{F}_K} (\partial_n u_h, v - \pi_h v)_F \tag{3.5.27}$$

In vertu of the Cauchy-Schwarz inequality, it follows:

$$|r(u_h, f; v)| \leq \sum_{K \in \mathcal{T}_h} \|f + \Delta u_h - u_h\|_{0,K} \|v - \pi_h v\|_{0,K} + \sum_{F \in \mathcal{F}_K} \frac{1}{2} \|[\![\partial_n u_h]\!]\|_{0,F} \|v - \pi_h v\|_{0,F} \tag{3.5.28}$$

where $\frac{1}{2}[\![\partial_n u_h]\!]$ denotes the mean value of the jump of $\partial_n u_h$ through the interior face F.
Therefore:

$$|r(u_h, f; v)| \leq \sum_{K \in \mathcal{T}_h} \left(\eta_K(v - \pi_h v) \left(h_K \|f + \Delta u_h - u_h\|_{0,K} + \sum_{F \in \mathcal{F}_K} \frac{1}{2} h_F^{1/2} \|[\![\partial_n u_h]\!]\|_{0,F} \right) \right) \tag{3.5.29}$$

with:

$$\eta_K(v - \pi_h v) = \max \left(h_K^{-1} \|v - \pi_h v\|_{0,K},\ h_F^{-1/2} \max_{F \in \mathcal{F}_K} \|v - \pi_h v\|_{0,F} \right) \tag{3.5.30}$$

h_K (resp. h_F) denotes the measure of K(resp. F).
That is:

$$|r(u_h, f; v)| \leq \sum_{K \in \mathcal{T}_h} \eta_K(v - \pi_h v) e_K(u_h, f) \tag{3.5.31}$$

with the local error indicator $e_K(u_h, f)$ defined by (3.5.24).

In other respect, estimations of the interpolation errors in the expression of η_K enable to show that, see e.g. **[?]** Chapter 10 for details:

$$\sum_{K \in \mathcal{T}_h} \left(\eta_K(v - \pi_h v) \right)^2 \leq c \, \|v\|_{1,K}^2 \tag{3.5.32}$$

We obtain from (3.5.31):

$$|r(u_h, f; v)| = a(u - u_h, v - \pi_h v) \leq c \, \|v\|_{1,K} \left(\sum_{K \in \mathcal{T}_h} \left(e_K(u_h, f) \right)^2 \right)^{1/2} \tag{3.5.33}$$

In other respect, the inf-sup condition () provides the following *stability inequality:*

$$\|u - u_h\|_1 \leq \sup_{v \in V} \frac{a(u - u_h, v)}{\|v\|_1} \leq \sup_{v \in V} \frac{|r(u_h, f; v)|}{\|v\|_1} \tag{3.5.34}$$

Hence the a-posteriori estimation (3.5.23).

3.5.5 Goal-oriented error estimator

3.5.5.1 Problem setup

The usual BVP context Let us consider a linear BVP whose the weak formulation reads as usual:

$$a(u, v) = b(v) \qquad \forall v \in V = H_0^1(\Omega) \tag{3.5.35}$$

where the Lax-Milgram theorem conditions are satisfied.
The FE approximation is P_k-Lagrange, $k \geq 1$. The discrete weak formulation reads as usual.
As previously, see (3.5.20), we define the residual:

$$r(u_h, f; v) = (f, v) - a(u_h, v) = a(u - u_h, v) = a(e_h, v) \quad \forall v \in V \tag{3.5.36}$$

with $e_h = (u - u_h)$ the error.

The model output In practice, the analyst is often interested by a particular model output / quantity of interest. For example,

$$J(u) = \frac{1}{|\omega|}\int_\omega |\nabla u|^2 dx \text{ or } J(u) = \frac{1}{|\omega|}\int_\omega \partial_x u \, dx \tag{3.5.37}$$

with ω a subset of Ω.
The output functional J, $J : V \to \mathbf{R}$ can be linear or not. If non-linear, it is supposed to be differentiable.

The goal is here to compute this model output J at a given accuracy.

To do so, one focuses on the following *error measure* $\mathcal{E}(u)$ defined by: $\mathcal{E}_h(u) = J(u) - J(u_h)$.
If J is linear then:

$$\mathcal{E}_h(u) = J(u - u_h) = J(e_h) \tag{3.5.38}$$

For a sake of simplicity, we assume in the sequel that J is linear, therefore (3.5.38) holds.

3.5.5.2 Duality-based estimation

The presentation below follows those proposed in [?]Chapter 10. For details and demonstrations, the reader may consult this reference and references therein.

We have the following result.

Proposition 55. *The error $\mathcal{E}_h(u)$ on the model output $J(u)$ satisfies:*

$$\mathcal{E}_h(u) = r(u_h, f; z - v_h) \quad \forall v_h \in V_h \tag{3.5.39}$$

where $r(u_h, f; v)$ is the residual defined by (3.5.36)
and z is the unique solution of the *dual problem:*

$$\begin{cases} \text{Given the solutions } u \text{ and } u_h, \text{find } z \in V \text{ such that:} \\ a^*(z, v) = J(u - u_h) \qquad \forall v \in V \end{cases} \tag{3.5.40}$$

where $a^*(\cdot,\cdot)$ is the adjoint form of $a(\cdot,\cdot)$: $a^*(\cdot, v) = a(v, \cdot)\ \forall v \in V$.

Remark 56. In the case $J(u)$ non-linear, the estimation (3.5.39) still holds. However, in this case the dual problem reads:

$$a^*(z, v) = \int_0^1 J'(u_h + s\ (u - u_h)) \cdot v \ ds \qquad \forall v \in V \tag{3.5.41}$$

3.5.5.3 Local error computation

We decompose the residual as sum of its values on each element:

$$r(u_h, f; v) = \sum_{K \in \mathcal{T}_h} r_K(u_h, f; v) \tag{3.5.42}$$

Next, we write the local residual as follows:

$$r_K(u_h, f; v) = (f - Au_h, v)_{0,K} + (\partial_{n_K} u_h, v)_{0,\partial K} \tag{3.5.43}$$

with A the differential operator, e.g. $A(u) = -div(\lambda \nabla u) + w \cdot u + c\ u$.

We denote by F the interior faces (edges in 2d) of the element K, $F \in \mathcal{F}_K$, such that $F = K_1 \cap K_2$.
We define the following jumps and means values:

$$[\![\partial_{n_K} u_h]\!] = \nabla u_h^{(1)} \cdot n_1 + \nabla u_h^{(2)} \cdot n_2 \tag{3.5.44}$$

$$\{v\} = \frac{1}{2}\left(v^{(1)} + v^{(2)}\right) \tag{3.5.45}$$

Then we have the following result.

Proposition 57. *The error $\mathcal{E}_h(u)$ on the model output $J(u)$ can be computed as follows:*
$\forall v_h \in V_h$,

$$\mathcal{E}_h(u) = \sum_{K\in\mathcal{T}_h} (f - Au_h, z - v_h)_{0,K} + \sum_{F\in\mathcal{F}_K} (\partial_{n_K} u_h, z - v_h)_{0,\partial K} \tag{3.5.46}$$

$$+ \sum_{F\in\mathcal{F}_K} ([\![\partial_{n_K} u_h]\!], \{z - v_h\})_{0,\partial K} + \sum_{F\in\mathcal{F}_K} (\{\nabla u_h\}, [\![z - v_h]\!])_{0,\partial K} \tag{3.5.47}$$

where A denotes the differential operator, e.g. $A(u) = -div(\lambda\nabla u) + w \cdot u + c\ u$.
z is the solution of the dual problem (3.5.40).

Chapter 4

Weak Constraints: Mixed Formulations

It is frequent in mathematical - numerical modeling to have a constraint on the field to be respected, in addition to the PDE model(s). And it may be interesting or necessary to consider this constraint in the weak sense and in the regular-"strong" sense (i.e. point-wise).

In the present chapter, the considered PDE systems contain a constraint to be imposed in the weak sense.

The first illustrative example is the incompressibility condition in the (Navier)-Stokes equations which is in a mathematical point of view nothing else than a constraint on the velocity field u.

An other important example are the continuity conditions at interfaces for unmatching meshes (in a model coupling context or domain decomposition context).

This chapter aims at studying the derivation of the resulting equations: it is mixed formulations; net the numerical resolution of such systems is addressed.

4.1 The (Navier-)Stokes fluid flow model

4.1.1 The flow model(s)

Let us consider the Navier-Stokes equations for an incompressible flow in a bounded domain $\Omega \subset R^n$.

We denote by $\mathbf{u}$ the velocity field and by p the pressure.

Assuming that the fluid is incompressible, the mass conservation equation reads: $div(\mathbf{u}) = 0$.

Given external forces $f(x)$, the momentum equations reads:

\rho

$$\rho d_t \mathbf{u} - div(\sigma(\mathbf{u})) = f \text{ in } \Omega \times (0,T) \tag{4.1.1}$$

with ρ the fluid viscosity, $\sigma(\mathbf{u})$ the constraint tensor: $\sigma(\mathbf{u}) = 2\mu D(\mathbf{u}) - pI_d$; μ the fluid viscosity (Newtonian fluid).

$D(\mathbf{u})$ is the strain rate (deformation) tensor: $D(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + {}^T\nabla \mathbf{u}\right)$.

If using the incompressibility condition $div(\mathbf{u}) = 0$, we obtain: $div(\sigma(\mathbf{u})) = \mu\, \Delta \mathbf{u} - \nabla p$.

In dimensionless form, the Navier-Stokes system may read as:

$$\begin{cases} Re\left(\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right)) - \Delta \mathbf{u} + \nabla p = f & \text{in } \Omega \times (0,T) \\ div(\mathbf{u}) = 0 & \text{in } \Omega \times (0,T) \end{cases} \tag{4.1.2}$$

with $Re = \rho \frac{L^* U^*}{\mu}$ the Reynolds number.

This PDE system is parabolic, non-linear (due to the inertial term $(\mathbf{u}\cdot\nabla)\mathbf{u}$). It has to be closed with adequate boundary conditions on $\partial\Omega \times (0,T)$.

Remark 58. If Neumann type boundary conditions are considered (e.g. normal constraints are imposed as $\sigma_n = -p_{ext} n$), then the momentum equation has to be kept in the divergence form as: $Re\left(\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right)) - div(\sigma(\mathbf{u})) = f$.

For low Reynolds number (typically $Re \approx 1$ and lower values), the Navier-Stokes system reduces to the *linear Stokes system*:

$$\begin{cases} -\Delta \mathbf{u} + \nabla p = f & \text{in } \Omega \\ div(\mathbf{u}) = 0 & \text{in } \Omega \end{cases} \tag{4.1.3}$$

The Stokes model can be employed to model creeping flows e.g. in biology, micro-fluidics, geophysics.

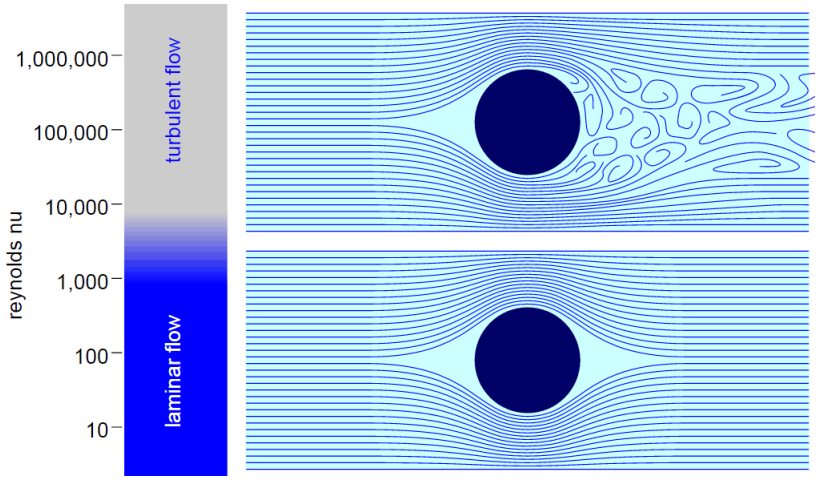


Figure 4.1.1: Viscous flows around a cylinder. Stokes model is valid for low Reynolds numbers $R_e \approx 1$ and lower (e.g. creeping geophysical or biological flows).

For a sake of simplicity, from now homogeneous Dirichlet conditions are imposed on the boundary : $\mathbf{u} = 0$ on $\partial\Omega$. This boundary condition is natural since representing the adherence of the viscous fluid on solid walls.

4.1.2 Formulation in the divergence free space V_{div}

A natural weak form of the Stokes system reads as follows. Find $\mathbf{u} \in V_{div}$ such that:

$$\int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\ dx = \int_\Omega f\mathbf{v}\ dx \qquad \forall \mathbf{v} \in V_{div} \tag{4.1.4}$$

with

$$V_{div} = \{\mathbf{v} \in \left(H_0^1(\Omega)^n\right) \text{ such that } div(\mathbf{v}) = 0\} \tag{4.1.5}$$

and the product: $A : B = (a_{ij}b_{ij})_{1\le i,j\le d}$.

The divergence free space V_{div} is a Hilbert space. In vertu of Lax-Milgram theorem, (4.1.4) is well-posed in V_{div}.

Since imposing the incompressibility condition in the functional space V_{div}, the pressure p has disappeared from the equation (4.1.4). Recovering the pressure field p from the unique solution $u \in V_{div}$, is not trivial; it is based on the de Rham theorem, see e.g. [**?**] Chapter 5 for a proof.

Let us point out that the Stokes system respects the same regularity results as the Laplacian or the system of linear elasticity. On the contrary since it is a system, no maximum principle holds (like in the elasticity system case).

The bilinear form $\int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\ dx$ in (4.1.4) is symmetrical. Then the energy corresponding to the Stokes model (4.1.3) reads:

$$J(v) = \frac{1}{2}\int_\Omega |\nabla u|^2\ dx - \int_\Omega f\ v\ dx \tag{4.1.6}$$

Then the unique (weak) solution $\mathbf{u} \in V_{div}$ of (4.1.13) minimizes the energy above in V_{div}:

$$J(u) = \min_{v\in V_{div}} J(v) \tag{4.1.7}$$

4.1.3 Formulation in variables $(\mathbf{u}, p)$: a mixed formulation

To approximate the Stokes solution by using a conforming FE method (i.e. an internal approximation), the solution space V_{div} defined by (4.1.5) is not adequate. Indeed it is difficult (possible but difficult) to built up basis functions respecting the divergence free condition... Then an other approach is adopted.

The approach consists to write the weak forms of both equations: the momentum equation (4.1.3)(a) and the mass equation (4.1.3)(b).

To do so, we multiply (4.1.3)(a) by a velocity test function $\mathbf{v} \in \left(H_0^1(\Omega)\right)^n$ and (4.1.3)(b) by a pressure test function $q \in L^2(\Omega)$.

After the usual integration by part it follows the weak form of the Stokes model:

$$\begin{cases} \text{Find } (\mathbf{u}, p) \in \left(H_0^1(\Omega)\right)^n \times L^2(\Omega)/\mathbf{R} & \text{such that:} \\ \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\ dx - \int_\Omega p\ div(\mathbf{v})\ dx & = \int_\Omega f\mathbf{v}\ dx \qquad \forall \mathbf{v} \in \left(H_0^1(\Omega)\right)^n \\ \int_\Omega div(\mathbf{u})\ q\ dx & = 0 \qquad \forall q \in L^2(\Omega)/\mathbf{R} \end{cases} \tag{4.1.8}$$

The weak formulation (a system) (4.1.8) is well-posed and its (unique) solution $(\mathbf{u}, p)$ is the solution of (4.1.3) (in the weak sense). The reader may refer to [**?**] for more details.

We set:

$$V = \left(H_0^1(\Omega)\right)^n \text{ and } M = L^2(\Omega)/\mathbf{R} \tag{4.1.9}$$

We define the following bilinear and linear forms.

$$\begin{array}{rl} a : V \times V \to & \mathbf{R} \\ (\mathbf{u}, \mathbf{v}) \mapsto & \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\ dx \end{array} \tag{4.1.10}$$

$$\begin{array}{rl} b : V \times M \to & \mathbf{R} \\ (\mathbf{v}, q) \mapsto & -\int_\Omega q\ div(\mathbf{v})\ dx \end{array} \tag{4.1.11}$$

$$\begin{array}{rl} l : V \to & \mathbf{R} \\ \mathbf{v} \mapsto & \int_\Omega f\ \mathbf{v}\ dx \end{array} \tag{4.1.12}$$

The bilinear form $a(.,.)$ is V-elliptic; moreover it is symmetric.

The weak formulation (4.1.8) re-reads as:

$$\begin{cases} \text{Find } (\mathbf{u}, p) \in V \times M & \text{such that:} \\ a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) & = l(\mathbf{v})\ \forall \mathbf{v} \in V \\ b(\mathbf{u}, q) & = 0\ \forall q \in M \end{cases} \tag{4.1.13}$$

The weak formulation system (4.1.13) is somehow an extension of the previous standard weak formulation $a(u, v) = l(v)$; it is called a *mixed formulation*. Other mixed formulations will be derived in next section.

4.1.4 The incompressibility constraint: p is the Lagrangian multiplier

In the Stokes model (4.1.3), the pressure can be interpreted as the Lagrangian multiplier of the constraint $div(\mathbf{u}) = 0$. Indeed let us define the Lagrangian $\mathcal{L}$ by:

$$\mathcal{L}(\mathbf{v}, q) = \frac{1}{2} a(\mathbf{v}, \mathbf{v}) + b(\mathbf{v}, q) - l(\mathbf{v})\ ,\ \forall \mathbf{v} \in V\ \forall q \in M \tag{4.1.14}$$

Observe that the Lagrangian $\mathcal{L}$ is defined as: *Lagrangian=Energy + weak constraint.*

The stationary point(s) of $\mathcal{L}(.,.)$ are denoted (u, p). they satisfy:

$$\nabla\mathcal{L}(u, p) = 0 \tag{4.1.15}$$

This 1st order necessary optimality condition is equivalent to the mixed formulation (4.1.13) !
Indeed we have:

$$\begin{cases} \partial_v \mathcal{L}(\mathbf{u}, p).v & = a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) - l(\mathbf{v}) = 0 \qquad \forall \mathbf{v} \in V \\ \partial_q \mathcal{L}(\mathbf{u}, p).q & = b(\mathbf{u}, q) = 0 \quad \forall q \in M \end{cases} \tag{4.1.16}$$

Therefore the fluid pressure p is nothing else than the Lagrangian multiplier of the incompressibility constraint $div(\mathbf{u}) = 0$.

Moreover the unique solution $(\mathbf{u}, p) \in V \times M$ of (4.1.13) satisfies:

$$\mathcal{L}(\mathbf{u}, p) = \min_{\mathbf{v}\in V} \max_{q\in M} \mathcal{L}(\mathbf{v}, q) \tag{4.1.17}$$

The Stokes solution $(\mathbf{u}, p) \in V \times M$ of (4.1.13) is the unique saddle-point of the Lagrangian $\mathcal{L}$ defined by (4.1.14).

The min-max formulation (4.1.17) suggests to compute the Stokes solution by using the Uzawa algorithm. (Please refer to your previous optimization course). This approach is an excellent one to solve the Stokes model (4.1.3); other approaches are possible, they are mentioned later.

4.1.5 Discrete form & linear system

Considering the mixed formulation (4.1.13) of the Stokes system, it is now much easier to build up internal approximations than if considering the divergence free form (4.1.4). To do so we consider the standard FE spaces:

$$V_h = (V_{h,d})^n \ ; \quad V_{h,d} = \{v_h, v_h \in C^0(\Omega), \quad v_h|_{K_i} \in \mathbf{P}_k \quad \forall K_i \in \mathcal{T}_h\} \tag{4.1.18}$$

$$M_h = \{q_h, q_h \in \left(C^0(\Omega)\backslash\mathbf{R}\right), \quad q_h|_{K_i} \in \mathbf{P}_{k'} \quad \forall K_i \in \mathcal{T}_h\} \tag{4.1.19}$$

Next we set as usual: $V_{0h} = \{\mathbf{v}_h, \mathbf{v}_h \in V_h, \quad \mathbf{v}_h = 0 \text{ on } \partial\Omega\}$.
Then the discrete weak formulation is straightforwardly obtained as:

$$\begin{cases} \text{Find } (\mathbf{u}_h, p_h) \in V_{0h} \times M_h & \text{such that:} \\ a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) & = l(\mathbf{v}_h) \ \forall \mathbf{v}_h \in V_h \\ b(\mathbf{u}_h, q_h) & = 0 \ \forall q_h \in M_h \end{cases} \tag{4.1.20}$$

Let us write the linear system equivalent to (4.1.20).
We denote by $\{\varphi_i(x)\}_{i=1..NNV}$ and by $\{\psi_k(x)\}_{k=1..NNM}$ the function basis of $V_{h,d}$ and M_h respectively.
Then the discrete variables are decomposed as:

$$v_{h,d}(x) = \sum_{i=1}^{NNV} v_{i,d} \ \varphi_i(x) \quad d = 1 \cdots n \text{ and } q_h(x) = \sum_{l=1}^{NNM} q_l \ \psi_l(x) \tag{4.1.21}$$

We arrange the unknown variables as follows:

$$U_{h,d} = (u_{1,d} \cdots u_{NNV,d}) \in \mathbf{R}^{NNV} \text{ for } d = 1 \cdots n \text{ and } P_h = (p_1 \cdots p_{NNM}) \in \mathbf{R}^{NNM} \tag{4.1.22}$$

Then (4.1.20) is equivalent to the following linear system (here $n = 3$):

$$\begin{bmatrix} A_{11} & 0 & 0 & B_1^T \\ 0 & A_{22} & 0 & B_2^T \\ 0 & 0 & A_{33} & B_3^T \\ B_1 & B_2 & B_3 & 0 \end{bmatrix} \begin{bmatrix} U_{h,1} \\ U_{h,2} \\ U_{h,3} \\ P_h \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix} \tag{4.1.23}$$

with:

$$A_{11} = A_{22} = A_{33} \equiv A = \left(\int_\Omega \nabla\varphi_j \ \nabla\varphi_i \ dx \right)_{1\le i,j\le NNV}, \ d = 1, .., n \tag{4.1.24}$$

and $B_d = \left(-\int_\Omega \psi_l \ \partial_d\varphi_i \ dx \right)_{1\le l\le NNM; \ 1\le j\le NNV}$.
The matrix A is of dimension NNV^2; it is symmetric, positive definite.
Each matrix B_d is rectangular of dimension $(NNM \times NNV)$.
The global matrix of (4.1.23) is of dimension $(n \times NNV + NNM)^2$; it is symmetric but *it is a-priori not positive definite...*
As a consequence the global matrix may be not invertible !...

In fact one has to set *compatible FE spaces* V_h, M_h such that the solution $(U_{h,1}, .., U_{h,n}, P_h)$ is unique.

Remark 59. If the viscous term is written as $div(\sigma(\mathbf{u}))$ and not as $(\Delta\mathbf{u} - \nabla p)$, see Remark 58, then the "extra diagonal blocks $A_{\square *}$" in (4.1.23) do not equal 0.

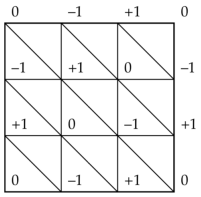


Figure 4.1.2: $(\mathbf{P}_1 - \mathbf{P}_1)$-Lagrange FE: a bad FE for the Stokes model. Unstable mode of the pressure p appear (here on a uniform triangular mesh). Image source: [**?**].

Well-posedness depending on the FE spaces pair (V_h, M_h)

We follow the presentation and proofs proposed in [**?**] Section 6.3.4.

As a first existence - uniqueness result, we have

Proposition 60. *Let us consider the linear Stokes system (4.1.23).*
We denote by: $U_h = (U_{h,1}, \cdots, U_{h,n}) \in \mathbf{R}^{n \times NNV}$ *and* $B = (B_1, \cdots, B_n) \in \mathcal{M}(\mathbf{R}^{NNM} \times \mathbf{R}^{n \times NNV})$.
i) Existence. The discrete Stokes system (4.1.23) admits a solution $(U_h, P_h) \in \mathbf{R}^{n \times NNV} \times \mathbf{R}^{NNM}$.
ii) Uniqueness. The velocity vector U_h *is unique;* P_h *is unique up to the addition of an element of* $Ker(B^T)$ *only.*
iii) $Ker(B^T)$ *contains at least the vector* $\mathbf{1}_{NNM}$ *,* $\mathbf{1}_{NNM} = (1, \cdots, 1) \in \mathbf{R}^{NNM}$.
As a consequence, the discrete pressure p_h *is at best defined up to a constant.*

Proof. i) (4.1.23) reads as: $AU_h + B^T P_h = b$ with $P_h \in Ker(B)$.

We have: $(Ker(B))^{\perp} = Im(B^T)$. Therefore (4.1.23) is equivalent to:

$$\text{Find } U_h \in Ker(B) \text{ such that: } (AU_h, W_h) = (b, W_h) \text{ for all } W_h \in Ker(B) \tag{4.1.25}$$

Next in vertu of the Lax–Milgram theorem (applied here in finite dimension), the existence and uniqueness of $U_h \in Ker(B)$ follows.

Therefore (4.1.23) has at least a solution $(U_h, P_h) \in R^{n \times NNV} \times R^{NNM}$.

ii) As U_h must belong to $Ker(B)$, it is unique in $R^{n \times NNV}$.

iii) Next it can be verified that P is unique up to the addition of an element of $Ker(B^T)$.

Let us consider $(\mathbf{w}_h, r_h) \in V_{0h} \times M_h$. By definition we have:

$$(W_h, B^T R_h) = (BW_h, R_h) = \int_\Omega r_h \; div(\mathbf{w}_h) \; dx \tag{4.1.26}$$

Let us set: $r_h = 1$; $r_h \in M_h$; that is: $R_h = \mathbf{1}_{NNM} = (1, \cdots, 1) \in \mathbf{R}^{NNM}$.

Observe that we have: $\int_\Omega div(\mathbf{w}_h) \; dx = \int_{\partial\Omega} \mathbf{w}_h \cdot n \; ds = 0$ for all $\mathbf{w}_h \in V_{0h}$.

Therefore: $(W_h, B^T \mathbf{1}_{NNM}) = 0$ for all W_h.

Therefore $\mathbf{1}_{NNM}$ belongs to $Ker(B^T)$, hence the result.

Let us clarify the uniqueness of the discrete solution $(U_h, P_h) \in R^{NNV} \times R^{NNM}$ for particular FE spaces.

Proposition 61. $(\mathbf{P}_2 - \mathbf{P}_1)$***−Lagrange FE, the Hood-Taylor element.***

Let us consider $\mathbf{P}_2$*-Lagrange FE for the velocity and* $\mathbf{P}_1$*-Lagrange FE for the pressure i.e.* $(k, k') = (2, 1)$ *in (4.1.18)(4.1.19).*

Then $dim(Ker(B^T)) = 1$*:* $Ker(B^T)$ *is generated by the vector* $\mathbf{1}_{NNM} = (1, \cdots, 1) \in \mathbf{R}^{NNM}$ *only.*

As a consequence, the $(\mathbf{P}_2 - \mathbf{P}_1)$-Lagrange FE Stokes solution is unique, with p_h defined up to a constant.

The Hood-Taylor element is the most classical element if considering continuous pressure p_h. It is an order 2 element.

In practice the constant may be set by imposing a value of pressure at a particular node, or by imposing its average on Ω (equal to zero for example).

Now let us show that the following a-priori simple and good element is actually unstable, therefore unusable.

Proposition 62. $(\mathbf{P}_1 - \mathbf{P}_1)$*-Lagrange FE: a bad element.*

Let us consider $\mathbf{P}_1$*-Lagrange FE both for the velocity and the pressure i.e.* $(k, k') = (1, 1)$ *in (4.1.18)(4.1.19).*

Then generally: $dim(Ker(B^T)) > 1$.

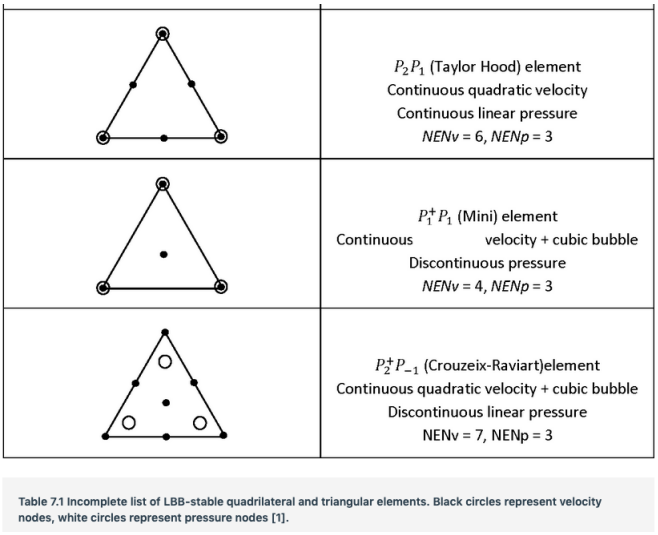


Figure 4.1.3: A few elements for the Stokes model respecting the LBB inf-sup condition. (Up) Hood-Taylor element ($\mathbf{P}_2$-Lagrange, $\mathbf{P}_1$-Lagrange): the classical one. Order 2, respect the mass conservation globally only. (Middle) The mini-element ($\mathbf{P}_1$-bubble,$\mathbf{P}_1$). order 1: ok for u_h but bad accuracy of p_h. (Down) Crouzeix-Raviart element ($\mathbf{P}_2$-bubble, $\mathbf{P}_1$-disc). Order 2; respect locally the mass conservation. Image source: semanticscholar.org

As a consequence, if using $(\mathbf{P}_1 - \mathbf{P}_1)$-Lagrange FE to solve the Stokes system then p_h is not unique even up to a constant....

In practice spurious pressure modes appear in the numerical solution, see e.g. Fig. 4.1.2. This FE scheme is *unstable* therefore useless to solve the Stokes model.

FE pairs for the Stokes model and corresponding order. First let us point out that FE schemes applied to fluid mechanics are studied in detail e.g. in [1].

In Fig. 4.1.3 are indicated a few standard FE pairs to solve the Stokes model (4.1.3) in variables $(\mathbf{u}, p)$.

We point out that if considering discontinuous pressure field, like e.g. the $(\mathbf{P}_2 - b, \mathbf{P}_1 - disc)$ element, the mass is preserved on each element K, see the dedicated exercise for the proof.

On the contrary if considering continuous pressure field e.g. the Hood-Taylor element, then the mass is globally preserved only.

4.1.6 On the Ladyzhenskaya–Babuška–Brezzi (LBB) inf-sup condition

To go further...

We introduce below the condition such that the solution of (4.1.20) is unique. Equivalently a condition such that it exists an unique solution (U_h, P_h) to (4.1.23).

To show the uniqueness, we show that $b = 0$ implies that the unique solution of (4.1.20) is $(\mathbf{u}_h, p_h) = (0, 0)$.

In (4.1.20) we set the test functions as: $(\mathbf{v}_h, q_h) = (\mathbf{u}_h, p_h)$. It follows:

$$\begin{cases} a(\mathbf{u}_h, \mathbf{u}_h) + b(\mathbf{u}_h, p_h) & = 0 \\ b(\mathbf{u}_h, p_h) & = 0 \end{cases} \tag{4.1.27}$$

Therefore: $a(\mathbf{u}_h, \mathbf{u}_h) = 0$. Since $a(\mathbf{u}_h, \mathbf{u}_h)$ is elliptic in V_{0h}, it follows that $\mathbf{u}_h = 0$ in V_{0h}.

Next the momentum equation of (4.1.20) reads:

$$b(\mathbf{v}_h, p_h) = 0 \qquad \forall \mathbf{v}_h \in V_{0h} \tag{4.1.28}$$

Let us assume that the following condition holds:

[1] Pironneau O., "Finite element methods for fluids"; John Wiley & Sons; Masson (1989).

$$\inf_{q_h \in M_h} \sup_{\mathbf{v}_h \in V_{0h}} \left(\frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_V} \right) \geq \beta \|q_h\|_M \tag{4.1.29}$$

That is $\exists \beta > 0$ such that:

$$\sup_{\mathbf{v}_h \in V_{0h}} \left(\frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_V} \right) \geq \beta \|q_h\|_M \text{ for all } q_h \in M_h \tag{4.1.30}$$

This is the so-called inf-sup condition, also called the Ladyzhenskaya–Babuška–Brezzi (LBB) condition.

By combining (4.1.28) and (4.1.30), it follows that:

$$0 = b(\mathbf{v}_h, p_h) \geq \beta \|p_h\|_M \|\mathbf{v}_h\|_V \qquad \forall \mathbf{v}_h \in V_{0h} \tag{4.1.31}$$

Therefore $p_h = 0$, hence the uniqueness.

The LBB condition (4.1.30) is a sufficient condition for a saddle point problem to have a unique solution.
In the Stokes model case, the LBB inf-sup condition ensures that $Ker(B^T)$ is reduced to the constant vectors.

To be admissible, any considered FE space pairs to solve the mixed formulation (4.1.20) should satisfy this LBB inf-sup condition.

For the Stokes model, this is the case of all the FE spaces pairs $(V_h \times M_h)$ indicated in Fig. 4.1.3.

4.2 Mixed formulations: other examples

4.2.1 General form & origins

Let us consider a general elliptic BVP and its variational form: find $u \in X$ such that

$$a(u, v) = b(v) \qquad \forall v \in X \tag{4.2.1}$$

Where $a(.,.)$ satisfies the Lax-Milgram theory assumptions. The problem (4.2.1) is assumed to be well-posed in X, X an adequate Hilbert space.

Note that the mixed formulation formalism may be extended to forms $a(.,.)$ which are non-linear with respect to u e.g. the Navier-Stokes equation.

There is many potential reasons to finally solve the corresponding mixed formulation:

$$\begin{cases} \text{Find } (u, \lambda) \in V \times M & \text{such that:} \\ a(u, v) + b(v, \lambda) & = l(v) \ \forall v \in V \\ b(u, \mu) & = 0 \ \forall \mu \in M \end{cases} \tag{4.2.2}$$

Let us cite the few following examples.

Ex 1. The usual Dirichlet boundary condition $u = 0$ on $\Gamma \subset \partial\Omega$ may be weakly enforced i.e. as a weak constraint while the original PDE is closed with the "natural"-"do-nothing" homogeneous Neumann b.c. on Γ.
For fluid flows (more precisely for the incompressible Navier-Stokes equations and the advection-diffusion equation), strongly imposed no-slip conditions at a wall may lead to inaccurate mean flow quantities for coarse boundary-layer meshes. Weakly imposed Dirichlet boundary conditions improve the scheme accuracy, see e.g. [2].

Ex 2. The non-penetration condition $u \cdot n = 0$ on $\Gamma \subset \partial\Omega$. This boundary condition is common in micro-fluidics e.g. for biological flows, also for geophysical flows (with friction conditions at bottom). This condition has to be imposed in the weak sense; indeed if strongly enforced, the numerical solution may be locally inaccurate.

Ex 3. A non-linear rheology flow law may be relaxed from the primitive equation (4.2.1) and taken into account as an additional (weak) constraint: $b(u, \mu) = 0 \quad \forall \mu$.

Ex 4. To couple two different models with non necessarily consistent grids, the coupling conditions at interface have to be weakly enforced.

[2] Bazilevs, Yuri, and Thomas JR Hughes. "Weak imposition of Dirichlet boundary conditions in fluid mechanics." Computers & Fluids 36.1 (2007): 12-26.

(Weak model coupling is addressed in the structural mechanics course, INSA Toulouse Applied mathematics department).

Below we briefly detail the principle for the cases Ex 1. and Ex 2.

4.2.2 Dirichlet boundary condition

Let us consider the general (and widely present in applications) linear elliptic BVP:

$$\begin{cases} -div(A\nabla u) = f & \text{in } \Omega \\ u = g & \text{on } \partial\Omega \end{cases} \tag{4.2.3}$$

Instead of imposing the boundary Dirichlet condition in the solution space V (e.g. $V = H_0^1(\Omega)$ for $g = 0$) next in the stiffness matrix A as described in Algorithm 2.2, one consider the same BVP ($-div(A\nabla u) = f$in Ω) with the "natural - do nothing" boundary condition plus the additional *constraint*: $u = g$ on $\partial\Omega$.

Then the problem reads as the following (weak) mixed formulation.

Find $(u, \lambda) \in H^1(\Omega) \times L^2(\partial\Omega)$ satisfying:

$$\begin{cases} \int_\Omega A\nabla u \ \nabla v \ dx + \int_{\partial\Omega} v \ \lambda \ dx = \int_\Omega f \ v \ dx & \forall v \in H^1(\Omega) \\ \int_{\partial\Omega} (u - g) \ \mu \ dx = 0 & \forall \mu \in L^2(\partial\Omega) \end{cases} \tag{4.2.4}$$

The constraint "$u = g$ on $\partial\Omega$" is imposed in the weak sense.

The system (4.2.4) can be interpreted as follows.

To obtain $u = g$ on $\partial\Omega$, one has to find the corresponding normal incoming flux $\lambda = (-\nabla u.n)$ such that $(u - g) = 0$ on $\partial\Omega$, see (4.2.4)(a).

For fluid flows, weakly enforced Dirichlet conditions may provide better results than strongly enforced conditions. For details on the formulation considered in fluid flow contexts (where adherence at a wall reads $g = 0$), the reader may refer to [Bazilevs-Hughes, Computers & Fluids] aforementioned.

4.2.3 Non-penetration boundary condition

Let us consider the Stokes system in variables $(\mathbf{u}, p)$:

$$\begin{cases} -(div(\sigma(\mathbf{u})))_d = f_d & \text{in } \Omega \ , \quad d = 1, \cdots, n \\ div(\mathbf{u}) = 0 & \text{in } \Omega \end{cases} \tag{4.2.5}$$

with $\sigma(\mathbf{u})$ the constraint tensor: $\sigma(\mathbf{u}) = 2\eta D(\mathbf{u}) - pId$, η the fluid viscosity and $D(\mathbf{u})$ the deformation tensor.
We have: $\sigma_{ij}(\mathbf{u}) = -p\delta_{ij} + \eta(\partial_i u_j + \partial_j u_i)$ $1 \leq i, j \leq n$ and $(div(\sigma(\mathbf{u})))_i = \sum_{j=1}^n \partial_j \sigma_{ij}(\mathbf{u})$.

On the boundary $\partial\Omega$, we set:

$$\sigma_n \equiv \sigma(\mathbf{u}) \cdot \mathbf{n} = \sigma_{nn}\mathbf{n} + \sigma_{n\tau}\tau \tag{4.2.6}$$

with $(\tau, \mathbf{n})$ direct.

Here we set: $\partial\Omega = \Gamma_d \cup \Gamma_f$. On Γ_d , we consider homogeneous Dirichlet conditions:

$$\mathbf{u} \cdot \mathbf{n} = 0 \tag{4.2.7}$$

On Γ_f we consider a *friction condition* defined as:

$$\begin{cases} \sigma_{n\tau} & = -\beta^2 \ \mathbf{u} \cdot \tau \\ \mathbf{u} \cdot \mathbf{n} & = 0 \end{cases} \tag{4.2.8}$$

where β^2 is the friction parameter, a given positive function.
We denote by: $u_n = \mathbf{u}.\mathbf{n}$ and $u_\tau = \mathbf{u}.\tau$.

If the equation $\mathbf{u}.\mathbf{n} = 0$ is strongly enforced (i.e. point-wise imposed in the FE space V_h) then the accuracy of the pressure p_h is bad in the vicinity of the wall Γ_f ... Then the good option is to consider the equation $\mathbf{u}.\mathbf{n} = 0$ as a *weak constraint* of the system.

Then we seek the (weak) velocity $\mathbf{u}$ solution of (4.2.5)-(4.2.8) in the functional space V_0 defined as:
$V_0 = \{\mathbf{v},\ \mathbf{v} \in (H^1(\Omega)^n,\ \mathbf{v} = 0 \text{ on } \Gamma_d\}$. The condition $\mathbf{u} \cdot \mathbf{n} = 0$ on Γ_f is not strongly enforced in the function space but weakly as a constraint. The pressure p is sough as usual in $M = L^2(\Omega)/\mathbf{R}$.

Then we solve the following mixed formulation.
Find $(\mathbf{u}, p, \lambda) \in V_0 \times M \times \Lambda = L^2(\Gamma_f)$ such that:

$$\begin{cases} a(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) + \int_{\Gamma_f} v_n\ \lambda\ ds & = l(\mathbf{v}) \qquad \forall \mathbf{v} \in V \\ b(\mathbf{u},q) & = 0 \qquad \forall q \in M \end{cases} \tag{4.2.9}$$

with the additional equation:

$$\int_{\Gamma_f} u_n\ \mu\ ds = 0 \qquad \forall \mu \in \Lambda \tag{4.2.10}$$

The forms $a(\mathbf{u},\mathbf{v})$, $b(\mathbf{v},q)$ and $l(\mathbf{v})$ are defined as previously, see (4.1.10)-(4.1.12).

The resulting system (4.2.9)(4.2.10) reads as a mixed formulation with two Lagrangian multipliers: the pressure p and the normal component of the normal constraint $\lambda = \sigma_{nn}$.
These two Lagrange multipliers are associated to the constraints $div(\mathbf{u}) = 0$ in Ω and $\mathbf{u} \cdot \mathbf{n} = 0$ on Γ_f respectively
The present example is studied more in detail as an exercise; see the accompanying exercises documents.

4.2.4 On the numerical resolution

Assuming that the PDE of the BVP is linear then the form $a(.,.)$ is bilinear and the discrete mixed formulation (4.2.2) have the following structure:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} U_h \\ \Lambda_h \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{4.2.11}$$

As already mentioned the well-posedness of the problem (4.2.2), equivalently the invertibility of the global matrix of (4.2.11), depends on the compatibility between the FE spaces V_h and M_h. It is assumed here that V_h and M_h are such that (4.2.11) is invertible.

We set: $dim(V_h) = NNV$ and $dim(M_h) = NNM$. Therefore $\dim(A) = NNV^2$ and $\dim(B) = NNM \times NNV$.

As already mentioned in the Stokes model section, to solve the particular linear system (4.2.11) it is natural to employ the Uzawa algorithm based on the augmented Lagrangian. A Schur complement algorithm is well-suited too (see details in Appendix). Oher approaches may be considered see e.g. the method(s) suggested in the FEniCS demo codes.

4.2.4.1 Augmented Lagrangian & Uzawa's algorithm

Below we recall the augmented Lagrangian method to solve (4.2.11). For more details the reader may refer to his previous optimization course.

We set the Lagrangian $\mathcal{L} : V \times M \to \mathbf{R}$ defined by:

$$\mathcal{L}(v,\mu) = \frac{1}{2}a(v,v) + b(v,\mu) - l(v) \tag{4.2.12}$$

Recall that the forms $a(.,.)$ and $b(.,.)$ are assumed to be bilinear V-elliptic and bilinear respectively. We have

Proposition 63. The pair (u,λ) is solution of the mixed formulation (4.2.2) if and only if it is saddle-point of the Lagrangian $\mathcal{L}(v,\mu)$ defined by (4.2.12). That is:

$$\mathcal{L}(u,\lambda) = \min_{v \in V} \max_{\mu \in M} \mathcal{L}(v,\mu) \tag{4.2.13}$$

Next for numerical efficiency purpose we define the *augmented Lagrangian*:

$$\mathcal{L}_r(v,\mu) = \mathcal{L}(v,\mu) + \frac{r}{2}\langle Bv, Bv\rangle \tag{4.2.14}$$

with $r > 0$, r large.

It is easy to show that any saddle-point of $\mathcal{L}_r(.,.)$ is saddle-point of $\mathcal{L}(.,.)$ too.

Let us set: $\lambda_r = r\,Bu_r,\ \lambda_r \in M_h$. Then computing the saddle-point of the augmented Lagrangian $\mathcal{L}_r(.,.)$ is equivalent to solve the following penalized system, [3].

Find $(u_r, \lambda_r) \in V_h \times M_h$ such that:

$$\begin{cases} Au_r + {}^T B_r\lambda_r & = b \\ Bu_r - \frac{1}{r}\lambda_r & = 0 \end{cases} \tag{4.2.15}$$

Resolution by the Uzawa algorithm.

The augmented - penalized system (4.2.15) is solved by the Uzawa algorithm.

After discretization in the FE spaces V_h and M_h, the algorithm reads as follows.

Let Λ_h^0 be given, for all $k \geq 0$, solve:

$$\begin{cases} (A + rB^TB)U_{h,r}^{k+1} + B^T\Lambda_{h,r}^k & = b \\ M(\Lambda_{h,r}^{k+1} - \Lambda_{h,r}^k) & = \rho\ BU_{h,r}^{k+1} \end{cases} \tag{4.2.16}$$

where M is the pressure mass-matrix.

Note that M may be interpreted as a preconditioning matrix, hence potentially defined such that it improves the iterative algorithm convergence.

Recall that we have

Proposition 64. For $0 < \rho < 2r$ and for all Λ^0, the Uzawa algorithm converges to the FE solution U_h of the mixed formulation (4.2.11).

In practice, r has to be set very large; typically one (empirically) set: $\rho = r \approx 10^7$.

Larger r is, more efficient the minimization is, but worse the conditioning number of the momentum equation is too...

Remark 65. - Recall that the Uzawa algorithm may be viewed as a gradient algorithm applied to the dual function $J_r^*(\mu) = -\mathcal{L}(v,\mu)$. (Observe that J^* is linear in μ).

- It follows from (4.2.16):

$$\Lambda_{h,r}^{k+1} = \rho\ M^{-1}BU_{h,r}^{k+1} + \Lambda_{h,r}^k \tag{4.2.17}$$

Next the momentum equation reads:

$$CU_{h,r}^{k+1} = b - B^T\Lambda_{h,r}^k \text{ with } C = (A + \rho\ B^TM^{-1}B) \tag{4.2.18}$$

The matrix C represents the "stiffness Stokes matrix" in the penalized system.

4.2.4.2 The Schur complement method

The Schur complement method consists to decompose and reduce the linear system (4.2.11); the reduced system may be solved by the preconditioned conjugate gradient algorithm.

This approach leads to an efficient algorithm since it is parallelizable. It constitutes the earliest version of non-overlapping Domain Decomposition Method (DDM).

For details the reader may consult the hand written notes available on the INSA Moodle platform and more importantly the complimentary DDM part of the present course.

[3] see e.g. Fortin, M., & Brezzi, F. (1991). Mixed and hybrid finite element methods (Vol. 734). Springer-Verlag, pp. 80-82

4.2.5 On the mixed FE method

To go further...

Let us finish by point out that the concept of mixed formulation may be considered for the Laplace operator or even any elliptic operator.

For example let us consider the elliptic BVP:

$$-div(A\nabla u) = f \tag{4.2.19}$$

with adequate boundary conditions e.g. $u = 0$ on $\partial\Omega$.

Instead of solving these equations, one solve an extended problem taken into account the flux as a variable. To do so, one set the new additional variable:

$$\phi = A\nabla u \tag{4.2.20}$$

and one consider the following mixed formulation. Find (ϕ, u) such that:

$$\begin{cases} -\int_\Omega div(\phi)\ v\ dx = \int_\Omega f\ v\ dx & \forall v \\ -\int_\Omega A^{-1}\phi\cdot\tau\ dx + \int_\Omega u\ div(\tau)\ dx & \forall \tau \end{cases} \tag{4.2.21}$$

for all $(v, \tau) \in L^2(\Omega) \times H(div)$.

Then the corresponding FE method differs from the one presented in the present course: it is the so-called *mixed Finite Element method.*

Part III

PDE Models Reductions

Introduction

The CPU-time of a FE code can be high, especially for 3D time-dependent non-linear models. High CPU-time for computing a numerical solution can be critical in many modeling scenari. *Extremly fast computations* (achieving "real-time" computations) may be necessary, for instance, when employing iterative optimization methods (requiring $O(10^p)$, $p = 2 - 4$, model solvings) or where real-time model outputs are necessary, such as for on-board predictions.

Another situation where fast computations are required arises when computing solutions for various values input parameter values μ, for instance if considering the μ-*parametrized* diffusive *model* $-div(\mu \nabla u) = f$.

The computation of the model outputs u corresponding to different values input parameter μ is necessary in the following cases:

a) *optimal control problems* where the objective is to control the model by adjusting the parameter μ;

b) *calibration problems* where the aim is to determine the value of the uncertain parameter μ;

c) *optimization problems* and design explorations.

In all of the aforementioned scenari, a large number of FE model resolutions are required. Therefore, in such contexts, a *drastical reduction in the CPU time needed for solving the model is necessary, while maintaining an acceptable level of numerical solution accuracy....*

Remark 66. Reduced modeling may be done by reducing the complexity of the mathematical model e.g. by reducing a spatial dimension by integrating 3D equations along one direction.

This is the reduction technique classicaly employed e.g. in:

- (fluid mechanics) the depth-integrated shallow-water flows models;
- (structural mechanics) shell models where the solid thickness is small compared to the two other directions.

Even for such mathematically reduced models, an additional *numerical model dimension* may be required.

By a reduction of degrees of freedoms (dof) of the numerical model, an approximation of the *High Resolution (*also called high-fidelity*) FE solution* is computed: this is a *Reduced Basis (RB) method.*

The latter are *projection-based techniques* for reducing the computational complexity of *parametrized PDEs.*

In order to be applicable to real-world problems, the *requirements of a reduced order model* are :

- A low computational cost (nearly real-time) while conserving fundamentals properties e.g. consistancy and stability,
- A reasonable approximation error compared to the HR FE solution.

The methods presented in this part are:

a) the Proper Orthogonal Decomposition (POD) method (equivalently PCA) which is optimal in some sense in the case *of linear PDEs*, moreover solved by FEM,

b) a hybrid POD-Neural Network method valid for *non-linear PDEs* (or for linear PDEs solved by the Finite Volume method for example).

Moreover, the greedy algorithm is briefly presented.

Finally, the reduction property of Auto-Encoders is briefy presented too.

For linear PDE cases, the presentation done in this part has been greatly inspired by [**?**, **?**, **?**].

For non-linear PDE cases, the presentation is inspired from recent research articles (including ours) which are cited along the text.

All the numerical results illustrating this chapter have been performed by Mustapha Allabou, PhD INSA-IMT, 2021-24.

Chapter 5

POD-based reduction

Contents

5.1 Reduced-Basis models in a FE context: basic principles

5.1.1 The original High-Resolution (HR) FE model

Let us consider the following steady-state *parametrized* bilinear form.

Given $\mu \in \mathbf{P}$, find $u(\mu) \in V$ satisfying:

$$a(\mu; u, v) = l(\mu; v) \qquad \forall v \in V \tag{5.1.1}$$

In this section and the POD method section, it is assumed that the PDE (5.1.1) is *linear* that is the mapping $u \mapsto a(\mu; u, \cdot)$ is linear.

Example. The parameter μ denotes the diffusivity in the diffusion-reaction operator $A(\mu; u) = -div(\mu \nabla u) + cu$ with $\mu \in \mathbf{P} \subset L^\infty(\Omega)$.

In this example the map $\mu \mapsto A(\mu; u)$ is affine.

In the general problem (5.1.1), the map $\mu \mapsto A(\mu; u)$ is a-priori non-affine: it is a non-affinely parametrized PDE.

As a consequence, the form $a(\mu; \cdot, \cdot)$ is bilinear. Moreover it is assumed to be V-coercitive. Therefore, Problem (5.1.1) fits into the Lax-Milgram theory.

The corresponding FE *parametrized model* reads as follows.

Given $\mu \in \mathbf{P}$, find $u_h(\mu) \in V_h$ satisfying:

$$a(\mu; u_h, v_h) = l(\mu; v_h) \qquad \forall v_h \in V_h \tag{5.1.2}$$

with: $u_h(x) = \sum_{i=1}^{NN} u_i \varphi_i(x)$; u_i the i-th dof, $\varphi_i(x)$ the i-th FE function basis of V_h.

We denote by $\Phi(x)$ the FE basis, $\Phi(x) = \{\varphi_i(x)\}_{1 \le i \le NN}$. Therefore: $V_h = span\ \Phi(x)$ and $dim(V_h) = NN$.

The vector of dof $U_h = (u_1, \cdots, u_{NN}) \in \mathbf{R}^{NN}$ satisfies the following linear system:

$$A^\mu U_h = F^\mu \tag{5.1.3}$$

with $A^\mu = (a^\mu_{ij})_{i,j=1..NN}$ *the stiffness matrix;* $a^\mu_{ij} = a(\mu; \varphi_j, \varphi_i)$.

Since it is assumed that the problem fits into the Lax-Milgram theory, the a-priori error estimation (2.4.5) holds. For regular exact solutions $u(\mu)$ (given μ), the estimation reads:

$$\|u(\mu) - u_h(\mu)\|_V \le C(u(\mu), \Omega)\ h^k \tag{5.1.4}$$

for a given order interpolation k. (Classically $k = 2$ or 1).

In all the sequel, the FE basis Φ(x) is called the High-Resolution (HR) basis.

5.1.2 The Reduced Basis FE model

Let us define the *Reduced Basis* (RB) V_{rb} as an internal approximation of V_h:

$$V_{rb} = span\ \Xi(x),\ \Xi(x) = (\xi_1(x), \dots, \xi_{N_{rb}}(x)), \tag{5.1.5}$$
$$V_{rb} \subset V_h \quad (V_h = span\ \Phi(x)) \tag{5.1.6}$$
$$\xi_n(\mathrm{x}) \in V_h \text{ for all } n,\ 1 \leq n \leq N_{rb} \tag{5.1.7}$$
$$dim(V_{rb}) = N_{rb} << NN = dim(V_h) \tag{5.1.8}$$

At this stage, the RB V_{rb} is not specified. It will done in next section.

We define the RB problem as the same as the original one but in the RB V_{rb} :

Given $\mu \in \mathbf{P}$, find $u_{rb}(\mu) \in V_{rb}$ satisfying:

$$a(\mu; u_{rb}, v_{rb}) = l(\mu; v_{rb}) \qquad \forall v_{rb} \in V_{rb}, V_{rb} \subset V_h \tag{5.1.9}$$

$u_{rb}(\mu)$ is by definition the RB solution.

Let us set B_{rb} the change of variable matrix between V_h and V_{rb}:

$$B_{rb} = [\xi_1 | \dots | \xi_{N_{rb}}],\ B_{rb} \in \mathcal{M}_{NN \times N_{rb}} \tag{5.1.10}$$

The vector ξ_n denotes here the coordinate vector of the function $\xi_n(x)$ in the FE basis $\Phi(x)$.
It follows:

$$\Xi(x) = B_{rb}^T\ \Phi(x) \tag{5.1.11}$$

The matrix B_{rb}^T encodes the change of variable from the HR FE basis $\Phi(x)$ to the Reduced Basis $\Xi(x)$.

A classical algebra result states that the matrix $\left(B_{rb}^T A^\mu B_{rb}\right)$ represents the bilinear form $a(\cdot,\cdot)$ in the basis V_{rb}.

Given the HR FE system ($A^\mu U_h = F^\mu$), see (5.1.3), the RB solution $U_{rb} \in \mathbf{R}^{N_{rb}}$ is defined as the solution of the following reduced dimension linear system:

$$\left(B_{rb}^T A^\mu B_{rb}\right) U_{rb} = B_{rb}^T F^\mu \tag{5.1.12}$$
$$\Leftrightarrow R^\mu U_{rb} = f^\mu \text{ with } R^\mu = B_{rb}^T A^\mu B_{rb} \in R^{N_{rb} \times N_{rb}},\ f^\mu = B_{rb}^T F^\mu \in R^{N_{rb}}. \tag{5.1.13}$$

Note that $R^\mu = B_{rb}^T A^\mu B_{rb}$ is a dense matrix, however of small dimension $N_{rb} \times N_{rb}$.
Note that $B_{rb} U_{rb} \in \mathbf{R}^{NN}$.
Then, one *expects* to have:

$$B_{rb} U_{rb} \approx U_h \tag{5.1.14}$$

Of course, the key point of a Reduced Basis method is the definition of V_{rb}, $V_{rb} = span\{\xi_{1,\cdots,}\xi_{N_{rb}}\}$, equivalently the definition of the matrix B_{rb}.

Remark 67. Note that if writing the basic FE decompositions, $u_{rb}(x) = \Xi^T(x) U_{rb}$ and $u_h(x) = \Phi^T(x) U_h$, with U_{rb} and U_h the respective dof vectors, it follows that:

$$u_{rb}(x) = \Phi^T(x) B_{rb}\ U_{rb} \tag{5.1.15}$$

Remark 68. The difference between the HR solution and the RB solution satisfies the following estimation:

$$\|u(\mu) - u_{rb}(\mu)\|_V \leq \|u(\mu) - u_h(\mu)\|_V + \|u_h(\mu) - u_{rb}(\mu)\|_V \tag{5.1.16}$$

with (5.1.4) which holds.
The definition of V_{rb} has to provide a satisfying estimation of the error $\|u_h(\mu) - u_{rb}(\mu)\|_V$.

5.2 Linear PDEs case: the POD reduction method

The *Proper Orthogonal Decomposition (POD)* method is far to be recent. It has been applied e.g. in turbulent fluid mechanics by J.L. Lumley in 1967. Its fundamental feature remains very interesting: under some assumptions, the POD-based RB is optimal *in the energy norm* $\|\cdot\|_V$.

Let us point out that given a matrix, its POD denotes nothing else than its *Principal Component Analysis (PCA)* in statistics.

Indeed, the PCA performs an orthogonal transformation to convert a set of measured variables (potentially correlated) into a set of linearly uncorrelated variables known as principal components.

5.2.1 The POD reduction method at a glance

The POD reduction method relies on a *offline-online strategy*.

- *Offline phase.*
  - A set of HR solutions corresponding to a whole set of parameters values μ, $\mu \in \{\mu_1, \ldots, \mu_M\}$, are computed. These HR solutions are called *snapshots*. They are stored in the snapshot matrix $\mathbf{S}$, $\mathbf{S} \in \mathcal{M}_{NN\times M}$, $\mathbf{S} = (\mathbf{u}_{\mu,1}|\cdots|\mathbf{u}_{\mu,M})$.
  - From this collection of snapshots, a Reduced Basis representing "at best" $\mathbf{S}$ is extracted by using a POD (equivalently a PCA).
    The POD method consists to retain the most influent modes of $\mathbf{S}$ by computing its Singular Value Decomposition (SVD).
    The retained singular vectors, typically $\mathcal{O}(10)$, constitute the so-called POD modes.
- *Online phase.*
  Given a new value of parameter μ^{new},
  - The reduced model matrix $\left(B_{rb}^T A^{\mu^{new}} B_{rb}\right)$ corresponding to (5.1.9) is (re-)built. it is a matrix of dimension N_{rb}, see (5.1.12).
  - This small dimension linear system is solved in "real-time" (in ms CPU-time). [1]

It is worth to notice that the *offline phase* is not restricted to FEM!

Indeed, the snapshots matrix may be obtained by employing any numerical scheme type such as Finite Volumes (FV) for example.

On the contrary, the online phase *as presented here* relies on the weak form of the model equation therefore suitable to FEM only.

For other type schemes, we will derive in next section a NN-based online phase which can applied to any type of numerical schemes therefore to FV schemes.

5.2.2 Solution manifolds

A manifold is a collection of points forming a certain kind of set such as a closed surface (or an analogue of this) in three or more dimensions. We present below a few concept related to the parametrized model solutions manifold.

5.2.2.1 Solution manifolds definition

Let us denote by $\mathcal{M}$ the set of all solutions u when the parameter μ describes $\mathbf{P}$:

$\mathcal{M} = \{u(\mu),\ u(\mu)$ solution of 5.1.9 with $\mu \in \mathbf{P}\}$. The corresponding set for discrete solutions reads:

$$\mathcal{M}_h = \{u_h(\mu),\ u_h(\mu) \text{ solution of 5.1.9 with } \mu \in \mathbf{P}\} \tag{5.2.1}$$

$\mathcal{M}_h$ (resp. $\mathcal{M}$) is a subset of V_h (resp. V) .

These sets are the *solution manifolds*, see Fig. 5.2.1.

[1] For steady-state models, real-time means here the shortest possible wall-clock time e.g. computations in ms.
For time-dependent computations, real-time typically denotes computations in 1 unit wall-clock time for each each second dynamic model time.

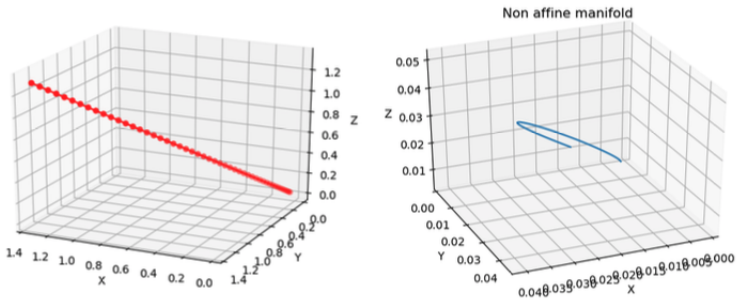


Figure 5.2.1: A solution manifold in: (Left) an affinely parametrized case (R) a non-affinely parametrized case.
The 3 axis represent here the 3 first components of the solutions in V_h.
Here $M = 50$ snapshots are represented (therefore 50 points).

5.2.2.2 Snapshots set

Let us consider the M-dimensional parameter space $\mathbf{P}_\delta$, $\mathbf{P}_\delta \subset \mathbf{P}$. $dim(\mathbf{P}_\delta) = M$.

Given $\mu_m \in \mathbf{P}_\delta$, $1 \leq m \leq M$, $u_{\mu,m} \equiv u_h(\mu_m)$ denotes the HR FE solution, unique solutions of (5.1.2).
The set

$$\mathcal{M}_{h,\delta} = \{u_h(\mu_1), \cdots, u_h(\mu_M)\} \tag{5.2.2}$$

is represented by the $(NN \times M)$-snapshot matrix S.

5.2.2.3 The Kolmogorov-width*

This is a "to go further section". [2]

Let us now introduce the "distance" between the functions $u_h(\mu)$ and a subspace X, $X \subset V$, as follows:

$$E(\mathcal{M}_h, X) = \sup_{u_h(\mu) \in \mathcal{M}_h} \inf_{v \in X} \|u_h(\mu) - v\|_X \tag{5.2.3}$$

The Kolmogorov N_{rb}-width of $\mathcal{M}_h$ in reduced spaces of dimension N_{rb} is then defined as:

$$d(\mathcal{M}_h, N_{rb}) = \inf_{X_{rb}, dim(X_{rb}) = N_{rb}} E(\mathcal{M}_h, X_{rb}) \tag{5.2.4}$$

In other words, the Kolmogorov N-width measures how $\mathcal{M}_h$ (the set of all HR FE solutions when μ describes $\mathbf{P}$) can be approximated by reduced solutions u_{rb} of dimension N_{rb}.

5.2.3 Recalls on the Singular Value Decomposition (SVD) and pseudo-inverse

SVD denotes a *diagonalization process* that can be applied to *rectangular matrices*. It involves left and right multiplications by *orthogonal* matrices.

The POD method and PCA method in data analysis are the same mathematical tool. Both rely on a SVD and an analysis of the most influencial modes.

5.2.3.1 SVD, singular vectors

The SVD of a rectangular (real) matrix A is as follows.

For $A \in \mathcal{M}_{N \times M}$ a (real) matrix, it exists two orthogonal matrices $L = (l_1 | \dots | l_N) \in \mathcal{M}_{N \times N}$, $R = (r_1 | \dots | r_M) \in \mathcal{M}_{M \times M}$ such that:

$$A = L \Sigma R^T \text{ with } \Sigma = diag(\sigma_1, \dots, \sigma_D) \in \mathcal{M}_{N \times M} \tag{5.2.5}$$

with $\sigma_1 \geq \cdots \geq \sigma_D \geq 0$, $D = \min(N, M)$.

[2] The Kolmogorov n-width si a tool to assess the effectiveness of approximating functions. It provides a characterization of optimal n-dimensional spaces for approximating functions and their associated errors.

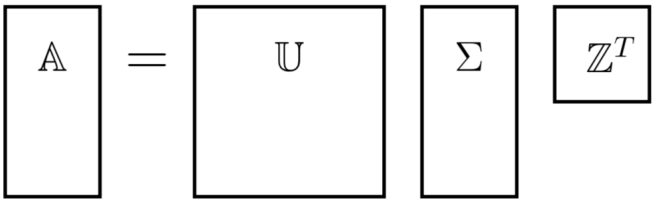


Figure 5.2.2: SVD decomposition of a matrix $A \in \mathcal{M}_{N\times M}$ with $N > M$. (Here: $U \equiv L$ and $Z \equiv R$).

The numbers σ_i are the singular values of A.
The vectors $\{l_i\}_{1\leq i\leq N}$ are the *left singular vectors* of A, $\{r_i\}_{1\leq i\leq M}$ are the *right singular vectors*. They satisfy:

$$\text{For } i = 1, \ldots, D, \qquad Ar_i = \sigma_i l_i \text{ and } A^T l_i = \sigma_i r_i \tag{5.2.6}$$

Moreover the SVD (5.2.5) implies the following spectral decompositions:

$$AA^T = L\Sigma\Sigma^T L^T \text{ and } A^T A = R\Sigma^T \Sigma R^T \tag{5.2.7}$$

with

$$\Sigma\Sigma^T = diag(\sigma_1^2, \ldots, \sigma_D^2, 0, \ldots, 0) \text{ and } \Sigma^T\Sigma = diag(\sigma_1^2, \ldots, \sigma_D^2, 0, \ldots, 0) \tag{5.2.8}$$

We have the relation: $\sigma_i(A) = \sqrt{\lambda_i(A^T A)}$, $i = 1, \ldots, D$.

Since AA^T (resp. $A^T A$) is symmetric, the left (resp. right) singular vectors of A are the *eigenvectors* of AA^T (resp. $A^T A$).

Recall that Rank(A)=Rank(Σ).

5.2.3.2 Pseudo-inverse

Let us denote by r the rank of $A \in \mathcal{M}_{N\times M}$. Then the matrix $A^\dagger = R\Sigma^\dagger L^T$ with $\Sigma^\dagger = diag(\sigma_1^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0)$, is the pseudo-inverse of A.

It satisfies:

$$A^\dagger = \left(A^T A\right)^{-1} A^T \tag{5.2.9}$$

for rank(A) = $M < N$.
Note that if rank(A) = $M = N$, we have: $A^\dagger = A^{-1}$.

5.2.3.3 The Schmidt-Eckart-Young theorem

Let us mention the *Schmidt-Eckart-Young theorem* (beginning of 20th century) which provides the best approximation (in some sense) of an arbitrary matrix by a matrix of lower rank n, n given.

We refer e.g. to [**?**] Chapter 6.

Theorem 69. (Schmidt-Eckart-Young theorem). Let $A \in \mathcal{M}_{N\times M}$ *be a real matrix of rank* r.
The matrix

$$A_k = \sum_{i=1}^{k} \sigma_i l_i r_i^T \text{ with } k \leq r \tag{5.2.10}$$

is optimal in the sense

$$\|A - A_k\|_F = \min_{B_k\in\mathcal{M}_{N\times M},\ rk(B)\leq k} \|A - B_k\|_F = \left(\sum_{m=(k+1)}^{r} \sigma_m^2\right)^{1/2} \tag{5.2.11}$$

The Schmidt-Eckart-Young approximation theorem will enable in next paragraph to establish an error estimation of the POD method.

5.2.4 The POD reduction method

The POD method enables to reduce the dimension of a given snapshot set by representing it onto an orthonormal basis which is optimal in the least-squares sense (following the Schmidt-Eckart-Young theorem).

The primitive variables are transformed into uncorrelated variables (the POD modes).
The first modes are the most influencial in the sense they contain most of the "energy" of the snapshot set.
The presentation below follows those proposed in [**?**] Chapter 6.

5.2.4.1 Preliminaries: L^2-norm vs energy V-norm

Let us first recall a basic result on scalar products and norms.

In the sequel we denote by $(\cdot,\cdot)_\square$ the scalar product either in L^2 ($\square = L^2$ or by simply omiting the symbol $\square$) or in V: $\square = V$.

Let us consider here the classical case $V = H^1(\Omega)$.
Let us assume that the PDE model is well-posed in V. In this case, V is called the energy space: V is the largest space such that the energy expression is well-defined in V.

With $V = H^1(\Omega)$, $(\cdot,\cdot)_V = (\cdot,\cdot)_{L^2} + (\nabla\cdot,\nabla\cdot)_{L^2}$.

In the discrete FE space V_h, considering internal approximation ($V_h \subset V$), we have:

$$\forall v \in V_h, \quad (v,v)_V = (N_h v, v) = v^T N_h v \equiv (v,v)_{N_h} \tag{5.2.12}$$

where N_h is a symmetric positive definite linear operator, a spd matrix of $\mathcal{M}_{NN\times NN}$, .

In this case, the matrix N_h is computed as the rigidity matrix of the bilinear form corresponding to the operator $(-\Delta v + v)$.

5.2.4.2 Definition of V_{POD}: SVD, eigenvectors

The snapshot matrix S and its SVD Let $\mathbf{u}_{\mu,m} = ((u_{\mu,M})_1, \ldots, (u_{\mu,M})_{NN}) \in \mathbf{R}^{NN}$ be the dof vector of the m-th snapshot. Each snapshot $\mathbf{u}_{\mu,m}$ belongs to V_h.

The snapshot matrix $\mathbf{S}, \mathbf{S} \in \mathcal{M}_{NN\times M}$, is built as:

$$\mathbf{S} = (\mathbf{u}_{\mu,1}|\cdots|\mathbf{u}_{\mu,M}) \tag{5.2.13}$$

In all the sequel it is assumed that $M < NN$.

Moreover the snapshots are supposed to be linearly independent therefore rank($\mathbf{S}$)= M.

Let us write the SVD of $\mathbf{S}$:

$$\mathbf{S} = L\Sigma R^T \text{ with } \Sigma = diag(\sigma_1, \ldots, \sigma_M) \in \mathcal{M}_{NN\times M} \tag{5.2.14}$$

with $\sigma_1 \geq \cdots \geq \sigma_M \geq 0$.

Then:

$$\mathbf{S}r_m = \sigma_m l_m \text{ and } \mathbf{S}^T l_m = \sigma_m r_m \quad m = 1, \ldots, M \tag{5.2.15}$$

with $\{l_m\}_{1\leq m\leq M}$ the *left singular vectors* and $\{r_m\}_{1\leq m\leq M}$ the *right singular vectors.*

Equivalently, we have:

$$\mathbf{S}^T\mathbf{S}r_m = \sigma_m^2 r_m \text{ and } \mathbf{S}\mathbf{S}^T l_m = \sigma_m^2 l_m \quad m = 1, \ldots, M \tag{5.2.16}$$

The correlation matrix C Recall the m-th snapshot decompositions: $u_{\mu,m}(x) = \sum_{i=1}^{NN}(u_{\mu,m})_i\varphi_i(x)$ and $\mathbf{u}_{\mu,m} = ((u_{\mu,m})_1, \ldots, (u_{\mu,m})_{NN}) \in \mathbf{R}^{NN}$.

We define the $M \times M$ *correlation matrix* $\mathbf{C}$ by $\mathbf{C} = (c_{mn})_{1\leq m,n\leq M}$,

$$c_{mn} = (u_{\mu,m}(x), u_{\mu,n}(x))_\square \qquad 1 \leq m,n \leq M \tag{5.2.17}$$

In the case of the basic **L^2-scalar product**, $(u_{\mu,m}, u_{\mu,n})_\square = (u_{\mu,m}, u_{\mu,n})_{L^2}$, we obtain:

$$\mathbf{C} = \mathbf{S}^T\mathbf{S} \tag{5.2.18}$$

In the case of the V**-scalar product**, we obtain:

$$\mathbf{C} = \mathbf{S}^T N_h \mathbf{S} \tag{5.2.19}$$

with N_h the symmetric positive definite matrix as previously defined.

Spectrum of C In both cases, $\mathbf{C}$ is symmetric positive definite then its spectrum is real strictly positive.

The eigenelements of $\mathbf{C}$ satisfies the relation, see (5.2.15):

$$\mathbf{C}\, r_m = \lambda_m r_m \qquad 1 \leq m \leq M \tag{5.2.20}$$

where the eigenvalues $\lambda_m(\mathbf{C}) = \sigma_m^2(\mathbf{S}),\ m = 1, \ldots, M$.

The eigenvectors of $\mathbf{C}$ *defined as above are the right singular vectors of* $\mathbf{S}$.

Remark 70. One could consider the "other correlation matrix" i.e. the large dimension one defined by $\mathbf{S}\mathbf{S}^T$ (L^2-scalar product case). In this case, the eigenelements problem to solve would be of large dimension therefore CPU-time consuming.

Definition of V_{POD}: the RB $\Xi(x)$ and the matrix B_{rb}

For a reduced basis dimension $N_{rb} \leq M$ (see later for the choice of the rank N_{rb}), the POD space V_{POD} is defined as:

$$V_{POD} \equiv V_{rb} = span\ \{l_1(x), \ldots, l_{N_{rb}}(x)\} \tag{5.2.21}$$

with $l_n(x) \in V_h$ defined as the N_{rb} *first left singular vectors* $\{l_m\}_{1\leq m\leq N_{rb}}$ *of* $\mathbf{S}$.
(The singular vector l_m is the coordinate vector of the function $l_m(x)$ in the FE basis $\Phi(x)$).

In the case of the basic L^2 **scalar product**, these singular vectors $\{l_m\}_{1\leq m\leq N_{rb}}$ are defined by (5.2.15).

In the case of the V **scalar product**, these singular vectors $\{l_m\}_{1\leq m\leq N_{rb}}$ are defined as in (5.2.15) **but for the matrix $\tilde{\mathbf{S}}$, $\tilde{\mathbf{S}} = N_h^{1/2}\mathbf{S}$.**

The left singular vectors $\{l_m\}_{1\leq m\leq M}$ can be efficiently deduced from the eigenvectors $\{r_m\}_{1\leq m\leq M}$ of $\mathbf{C}$ (equivalently the right singular vectors of $\mathbf{S}$) as:

$$l_m = \frac{1}{\sigma_m}\mathbf{S} r_m \qquad 1 \leq m \leq M \tag{5.2.22}$$

Following the basic principle (5.1.5)(5.1.12), we set:

$$B_{rb} = [\xi_1 | \ldots | \xi_{N_{rb}}] \text{ with } \xi_m = l_m.\ B_{rb} \in \mathcal{M}_{NN\times N_{rb}}. \tag{5.2.23}$$

The m-th column of B_{rb} contains the coefficients of the m-th RB vector l_m in the FE basis $\Phi(x) = \{\varphi_i(x)\}_{1\leq i\leq NN}$, see (5.2.23).

B_{rb}: a semi-orthonormal matrix. By construction, B_{rb} is an (semi-)orthonormal basis since we have (L^2-scalar product case) $B_{rb}^T B_{rb} = I_{N_{rb}}$; however $B_{rb}B_{rb}^T \neq I_{NN}$.

5.2.4.3 The orthogonal projector and error estimation

The orthogonal projector $\mathbf{P}_{rb}$ Let us detail the expression of the orthogonal projection from V_h on $V_{rb} = span\ \Xi(x)$:

$$\forall v \in V_h, \qquad \mathbf{P}_{rb}(v(x)) = \sum_{m=1}^{N_{rb}} (v(x), \xi_m(x))_\square\ \xi_m(x) \tag{5.2.24}$$

Let us write the projector expression in matrix form.

For the L^2-scalar product $(\cdot,\cdot)$, the projector expression reads:

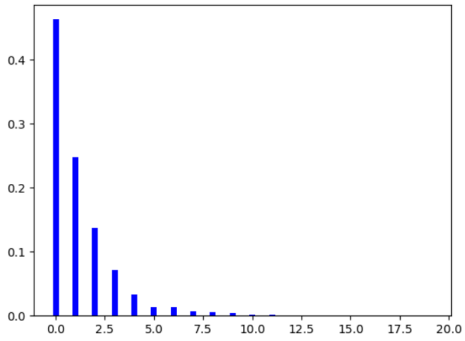


Figure 5.2.3: Normalized eigenvalues of the correlation matrix C for $M = 20$ snapshots of a non-affinely parametrized linear BVP. *In a dominant diffusive phenomena,* the very first modes only are important.

$$\mathbf{P}_{rb}\mathbf{v} = B_{rb}B_{rb}^T\mathbf{v} \text{ for } \mathbf{v} \in \mathbf{R}^{NN} \tag{5.2.25}$$

For the V-scalar product $(\cdot,\cdot)_V$ (energy norm case), the projector expression reads:

$$\mathbf{P}_{rb}\mathbf{v} = B_{rb}B_{rb}^T N_h\mathbf{v} \text{ for } \mathbf{v} \in \mathbf{R}^{NN} \tag{5.2.26}$$

Observe that $B_{rb}^T\mathbf{v} \in R^{N_{rb}}$; also recall that $B_{rb}B_{rb}^T \neq I_{NN}$.

Error estimation Before stating the error estimation, let us define the set of semi-orthonormal[3] bases of dimension N_{rb}:

$$\mathcal{B}_{N_{rb}}^{\perp} = \{B \in \mathcal{M}_{NN\times N_{rb}}, B^T N_h B = I_{N_{rb}}\} \tag{5.2.27}$$

Recall that $\mathbf{N}_h = Id_{NN}$ in the case of the basic L^2-norm.

Proposition 71. Among the semi-orthonormal basis of dimension N_{rb}, the POD space V_{POD} represented here by B_{rb} is optimal in the least square sense. Indeed we have:

$$\sum_{m=1}^{M} \|\mathbf{u}_{\mu,m} - B_{rb}B_{rb}^T N_h\mathbf{u}_{\mu,m}\|_2^2 = \min_{B\in\mathcal{B}_{N_{rb}}^{\perp}} \sum_{m=1}^{M} \|\mathbf{u}_{\mu,m} - BB^T N_h\mathbf{u}_{\mu,m}\|_2^2 = \sum_{m=(N_{rb}+1)}^{M} \lambda_m \tag{5.2.28}$$

where $\mathbf{u}_{\mu,m}$ denotes the m-th snapshot and λ_m denotes the m-th eigenvalue of the correlation matrix C.

In the present formalism, we necessarily have $N_{rb} \leq M$ (M the total number of snapshots and N_{rb} the RB dimension).

The proof of Proposition 71 derives from the Schmidt-Eckart-Young theorem showing that the best approximation of a given matrix by a lower rank matrix is the one obtained by SVD.

The complete proof of the proposition can be found in [**?**] Chapter 6.

Remark 72. Proposition 71 states that given N_{rb}, the POD basis B_{rb} is the optimal basis (optimal in the least square sense) to represent the M snapshots.
However it is *a-priori* not optimal for solutions corresponding to other values of parameter μ !

In practice, it is frequent to observe an exponential decay of the eigenvalues λ_m, see Fig. 5.2.3.

Reduced basis dimension: the maximal conserved rank N_{rb} In practice, the dimension of the RB is defined from the error estimation (5.2.28).

From the "energy-based ratio" $R(n) = \left(\sum_{i=1}^{M}\lambda_i / \sum_{i=1}^{n}\lambda_i\right)$, $0 < R(n) \leq 1$, the modeler chooses the energy ratio to be preserved.

[3]Semi-orthonormal property denotes here the orthonormal property for rectangular matrices

The later is defined as defined as $R(n) = (1-\varepsilon_{lost})$ with ε_{lost} to be set. Typical values of energy loss will be $\varepsilon_{lost} \sim 10^{-5}$. Then, N_{rb} is defined as the minimal value such that:

$$R(N_{rb}) \geq (1 - \varepsilon_{lost}) \tag{5.2.29}$$

N_{rb} is the maximal conserved rank.

5.2.4.4 What relation(s) between U_{rb} and U_h ?

Recall that:

- $U_h \in R^{NN}$ is defined as the unique solution of $AU_h = F$, see (5.1.3).
- $U_{rb} \in R^{N_{rb}}$ is defined as the unique solution of $\left(B_{rb}^T A B_{rb}\right) U_{rb} = B_{rb}^T F$, see (5.1.12).
- Given the HR FE solution U_h, if we define

$$W_{rb} = B_{rb}^T U_h \tag{5.2.30}$$

then W_{rb} represents U_h in the RB V_{rb}.

- Proposition 71 states that the snapshots approximations $W_{\mu,m} = B_{rb} B_{rb}^T \mathbf{u}_{\mu,m}$ ($W_{\mu,m} \in \mathbf{R}^{NN}$, $B_{rb}^T \mathbf{u}_{\mu,m} \in \mathbf{R}^{N_{rb}}$) are optimal in the least square sense (considering the L^2-scalar product).
- The pseudo-inverse $B_{rb}^\ddagger$ satisfies: $B_{rb}^\ddagger = \left(B_{rb}^T B_{rb}\right)^{-1} B_{rb}^T = B_{rb}^T$, see (5.2.9). Therefore: $B_{rb}^\ddagger = B_{rb}^T$.
  Indeed (L^2-scalar product case), $B_{rb}^T B_{rb} = I_{N_{rb}}$ since B_{rb} is semi-orthonormal. However $B_{rb} B_{rb}^T \neq I_{NN}$.
- As a consequence: $B_{rb} U_{rb} \in R^{NN}$ however $U_h \neq B_{rb} U_{rb}$.
  Also, $U_{rb} \neq B_{rb}^T U_h \equiv W_{rb}$.

Indeed, let us calculate the residual of the reduced model applied to W_{rb}:

$$\begin{aligned} \left(B_{rb}^T A B_{rb}\right) W_{rb} - B_{rb}^T F &= B_{rb}^T A \left(B_{rb} B_{rb}^T U_h - A^{-1} F\right) \\ &= B_{rb}^T A \left(B_{rb} B_{rb}^T - I_{NN}\right) U_h \\ &\neq 0_{N_{rb}} \end{aligned} \tag{5.2.31}$$

Hence the statements.
However, for U_h not being a snapshot we *expect to* have:

$$B_{rb} U_{rb} \approx U_h \text{ equivalently } U_{rb} \approx B_{rb}^T U_h \tag{5.2.32}$$

5.2.4.5 A few other remarks

- The POD solution accuracy (the accuracy of u_{rb}) depends on the discretization of the parameter space P i.e. both on the number M and on the choice of these M snapshots.
  The sampling of P, equivalently the choice of the snapshots, is a key point of the method.
  Unfortunately, the strategy to define P remains an open question.
- On the orthonormalization of the snapshots
  If the M snapshots $\{u_{\mu,1}, \cdots, u_{\mu,M}\}$ are nearly linearly dependent then the snapshot matrix $\mathbf{S}$ presents a large condition number. In this case, it is a good idea to orthonormalize the snapshots by using the *Gram-Schmidt algorithm* before computing the eigenvectors.

5.2.5 The algorithm

The POD-based reduction method provides the algorithm below, Algo. 5.1.

Algorithm 5.1 The POD-based Reduced Basis algorithm

Offline phase

- Define the M values of parameters μ you want the solution.
  For $\mu \in \mathbf{P}_\delta$ with $\mathbf{P}_\delta$ of dimension d, the parameter space may be regularly digitalized as a hypercube of dimension d, with $M = M_0^d$.

- Compute the M corresponding snapshots (HR FE solutions) $u_{\mu,m} \equiv u_h(\mu_m)$, $1 \leq m \leq M$ with μ_m previously chosen. The dof of the HR solution $u_{\mu,m}(x)$ are denoted by $\mathbf{u}_{\mu,m}$.

  Store these HR solutions in the $(NN \times M)$-snapshot matrix: $\mathbf{S} = (\mathbf{u}_{\mu,1}|\cdots|\mathbf{u}_{\mu,M})$.

- Build up the correlation $(M \times M)$-matrix $\mathbf{C}$,
$$\mathbf{C} = \mathbf{S}^T \mathbf{N}_h \mathbf{S} \tag{5.2.33}$$
  $\mathbf{C}$ is symmetric positive definite. It is a dense matrix.

- Compute its eigenelements: $(\lambda_n, r_n) \in \mathbf{R} \times \mathbf{R}^M$ with $\|r_n\|_V = 1$,
$$\mathbf{C}\, r_n = \lambda_n r_n \qquad 1 \leq n \leq M \tag{5.2.34}$$
  Define the target conserved ratio energy $(1 - \varepsilon_{lost})$ providing the maximal conserved rank N_{rb} from the "conserved energy ratio" $R(n) = \left(\sum_{i=1}^{M} \lambda_i / \sum_{i=1}^{n} \lambda_i\right)$.
  That is N_{rb} is defined the minimal rank such as $R(N_{rb}) \geq (1 - \varepsilon_{lost})$.

- Deduce the N_{rb} largest left singular vectors of $\mathbf{S}$ from the eigenvectors r_n:
$$l_n = \frac{1}{\sqrt{\lambda_n}} \mathbf{S} r_n \qquad 1 \leq n \leq N_{rb} \tag{5.2.35}$$

- Build up the $(NN \times N_{rb})$-Reduced Basis matrix B_{rb}:
$$B_{_{rb}} = \left(l_1|\cdots|l_{_{N_{rb}}}\right) \tag{5.2.36}$$

End: the reduced system has been built from the M (arbitraly?) chosen snapshots.

Online phase (real-time computations)
Given a new parameter value μ,

- Re-assembly the $(NN \times NN)$-rigidity matrix A^μ.
  For *non-affinely parametrized* PDEs, this step necesitates $\mathcal{O}(kNN)$ operations: it is CPU-time consuming !... To avoid this bottlneck, Empirical Interpolation Method (EIM) or a Hyper Reduction Method (HRM) can be considered, see e.g. [**?**] for details.
  For *affinely parametrized* PDEs, the stiffness matrix can be decomposed as $A^\mu = A_0 + \mu A_1$. As a consequence, the computation of A^μ can be done in real-time.

- Build up the RB stiffness matrix R^μ, $R^\mu = B_{rb}^T A^\mu B_{rb}$, and the RHS f^μ, see (5.1.12).

- Solve the low dimension $(N_{rb} \times N_{rb})$-linear system (5.1.12): the Reduced Basis (POD) solution U_{rb} is obtained in real-time, $U_{rb} \in R^{N_{rb}}$.

  The RB solution $u_{rb}(x)$ can be written in the FE basis $\{\varphi_i(x)\}_{1 \leq i \leq NN}$, specifically to visualize it on the FE mesh, as: $U_{rb}^{NN} = B_{rb} U_{rb}$; $U_{rb}^{NN} \in R^{NN}$.

  Given N_{rb} and following Proposition 71, the vector U_{rb}^{NN} belongs to the optimal RB.

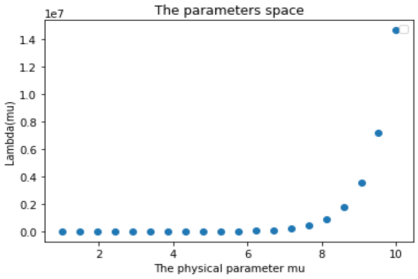


Figure 5.2.4: Sampling of the parameter space P: a key point of snapshots-based reduction methods.
Here a simple 1D example, $P = [\mu_{min}, \mu_{max}]$, sampled by using a regular grid (equidistant points). Is it the best strategy ? Likely not...

5.2.6 Discussions

5.2.6.1 Summary of the method

Let us summarize the POD method (also called POD-Galerkin method).

An offline-online strategy is adopted.

At offline phase, a sample in the parameter space P is chosen.

The HR solver is performed to computed the M corresponding HR solutions. This constitutes the reference snapshots set **S**.

The (spd dense) correlation matrix **C** is built and its eigenelements are computed (equivalently a SVD of the snapshot matrix **S**).

The reduced dimension basis $V_{rb} = V_{POD}$ of dimension N_{rb} given, is the optimal basis to represent the snapshots, see Proposition 71.

The N_{rb} first most influencial modes of **C** are retained. The resulting $(NN \times N_{rb})$-Reduced Basis matrix B_{rb} follows.

As already mentioned, the offline phase can rely on any numerical method type: FE but also FV or FD etc.

At on-line phase, given a new parameter value μ^{new}, the corresponding solution $U_{rb}^{\mu^{new}}$ is computed as the solution of the reduced dimension (5.1.9).

Since the RB stiffness matrix $R^{\mu^{new}} = B_{rb}^T A^{\mu^{new}} B_{rb}$ is small dimension, the linear system (5.1.12) is solved in extremely short time ("real-time").

In the end, we expect to have:

$$B_{rb}U_{rb} \approx U_h \text{ equivalently } U_{rb} \approx B_{rb}^T U_h \tag{5.2.37}$$

Remark 73. For non-affinely parametrized PDEs, an EIM method or a "Hyper Reduction Method" has to be adopted to assembly the reduced matrix R^μ in real-time.

- The on-line phase above relies on the weak form of the model equation (5.1.9): the computation of the RB solution U_{rb} relies on the expression of the bilinear form $a(\cdot,\cdot)$ in the RB V_{rb}, see (5.1.9).
  Consequently, the present online phase is specific to a FEM context.
- In a FV context, the on-line phase has to be differently built up e.g. using the NN-based approximation described in the next section.

5.2.6.2 Advantages & disadvantages of the POD method

The advantages of the POD method are clear.

- Given the snapshots set $\{u_{\mu,1}, \cdots, u_{\mu,M}\}$, the corresponding RB solutions $u_{rb}(\mu)$ are optimal, see Proposition 71.
- At the offline phase, the POD-based method is non-intrusive: the original HR code providing the snapshots set can be employed as a black box.

The snapshots can be generated using indifferently a Finite Element (FE) code or a Finite Volume (FV) code for instance.

The disavantages of the POD method are clear too.

- The dimension of P must be very small. For instance a subset of $\mathbf{R}^3$ is already large if finely digitalized e.g. with $M = (10^2)^3$.
- The choice of the snapshots and the choice of their number M is a critical point and no one really known how to choose them.
- *Offline phase.* The computation of the eigenvalues of the dense matrix $\mathbf{C}$ by a Lanczos method requires $\mathcal{O}(N_{rb} \cdot NN^2)$ operations. Therefore for a large number M of snapshots and for NN large, the offline phase can be very CPU-time consuming.
- *Online phase.* For *non-affinely parametrized* linear model, one has $A(\mu) \neq A_0 + \mu A_1$.
  As a consequence, the assembly of the HR stiffness matrix A^μ depends on NN.
  To be actually real-time, an Emprical Interpolation Method (EIM) or the Hyper Reduction Method as in e.g. [**?**], has to be adopted to built up A^μ in real-time.

Finally, *the POD method constitutes the reference method for linear PDEs, affinely parametrized or not, presenting a few number of parameter μ only ($\mu \in P \subset \mathbf{R}^d$ with $d \approx 3$).*

5.2.6.3 How about in the context of Finite Volumes or if dealing with a non-linear PDE?

In the case of a *non-linear* PDEs solved by a FEM, one naturally guess to reduce the linearized PDE solved e.g. in the Newton-Raphson algorithm.

However this natural simple approach does not offer any guarantee of convergence.
Worse, the RB linearized may be ill-posed. In short, this approach may at best roughly work, or more likely not at all...

For non -linear PDEs, a few approaches are possible but none is universal for all types of non-linearity.
For a more detailed discussions, the reader may consult [**?**] Chapter 11.

However either for non-linear PDEs or in a FV context (whatever of the PDE is linear or not), the POD basis V_{POD} can be employed if completed by a Machine Learning process.
Such a hybrid POD-DNN approach is presented in the next section.

5.2.7 Greedy algorithm*

Greedy algorithm denotes an iterative procedure where one (1) basis function is added at each iteration. Therefore the computation of 1 HR FE solution is required at each iteration.

For a *linear coercive equation*, the fundamental feature of the greedy algorithm is the availability of an *error estimation predicting the error made* at each iteration.

For complex non-linear problems, the greedy method can be empirically used only since no criteria is known for the choice of the snapshots.

If interested, the reader may consult [**?**] Chapter 1.

5.3 Numerical results

5.3.1 Advection-diffusion equation

The numerical results below have been extracted from the proposed Programming Practical (PP).

Consult the INSA Moodle page of the course for details and to obtain Python codes. Other examples can be produced by performing your own PP code :)

The domain Ω is a square. The boundary $\partial\Omega$ is decomposed as: $\partial\Omega = \Gamma_{diri} \cup \Gamma_{neumann}$.

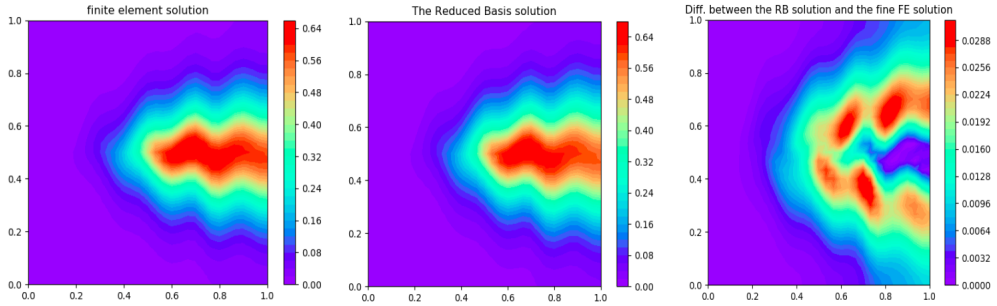


Figure 5.3.1: POD method. Solutions of the μ-parametrized linear advection-diffusion model 5.3.1. Non-affine case with $\lambda(\mu) = exp(\mu_0(1+\mu))$.
Offline phase: $M = 40$ snapshots are considered.
Online phase. Solution for $\mu = 2.68$ with 99% of energy which is conserved (this corresponds to $N_{rb} = 2$ i.e. 2 modes).
(Left) The HR FE solution. (Middle) The RB solution (projected on the mesh by using B_{rb}). (Right) The absolute difference between the two solutions (error).
Computations performed by M. Allabou (INSA-IMT, 2021).

The model is the following linear convection-diffusion one:

$$\begin{cases} -div(\lambda(\mu)\nabla u(x)) + w(x)\nabla u(x) = f(x) & \text{in } \Omega \\ -\lambda\nabla u \cdot n(x) = \varphi & \text{on } \Gamma_{neuman} = \Gamma_{up} \cup \Gamma_{down} \cup \Gamma_{right} \\ u & \text{given on } \Gamma_{diri} = \Gamma_{left} \end{cases} \tag{5.3.1}$$

The model is μ-parameter through the diffusivity coefficient $\lambda(\mu)$, $\lambda(\mu) > 0$ a.e.
Offline phase
The number M of snapshots is empirically chosen following a few trial-error tests.
Each snapshot is a HR FE solution corresponding to a different value of the parameter μ.
Online phase
Given a new parameter value, the RB solution $\mathbf{u}_{rb}$ is computed.

5.3.2 A few Python libraries

Few open-source Python libraries to develop reduced models have been released. Let us mention some of them existing in 2021. (This may have evolved since !).

RBniCS (Reduced Basis in FEniCS) library [4] is "an implementation in FEniCS of several reduced order modelling techniques (and, in particular, certified reduced basis method and Proper Orthogonal Decomposition-Galerkin methods) for parametrized problems".

We may mention a few others:

- The pyMOR library [5][6]which is a "software library for building model order reduction applications with the Python programming language. Implemented algorithms include reduced basis methods for parametric linear and non-linear problems, as well as system-theoretic methods such as balanced truncation or IRKA (Iterative Rational Krylov Algorithm)".
- The EZyRB[7] (Easy Reduced Basis method) Python library based on POD.

Please consult the indicated webpages to obtain detailed information and the accompanying scientific references.

[4]https://gitlab.com/RBniCS/RBniCS
[5]R. Milk, S. Rave, F. Schindler. "pyMOR - Generic Algorithms and Interfaces for Model Order Reduction", SIAM J. Sci. Comput., 38(5), pp. 194--216, 2016.
[6]https://mathlab.github.io/EZyRB/build/html/index.html
[7]https://mathlab.github.io/EZyRB/build/html/index.html

Chapter 6

Hybrid POD - ML method

Given the limitations of the standard POD method (linear PDEs solved by FEM only), we present here a method combining the POD and a Machine Learning technique enabling to reduce non-linear PDEs or linear models solved by FV, on contrary to the standard POD method.

The present hybrid POD - Machine Learning approach consists to:

1. consider the POD reduced basis, represented by B_{rb},
2. identify the coefficients of U_{rb} in B_{rb} by a Machine Learning (ML) process.

Consequently, the offline phase requires here an huge amount of data (ML process). Moreover, it is very CPU-time consuming, without a-posteriori error estimation.

However, it can be applied to non-linear PDEs or to linear models solved by FV, on contrary to the standard POD method.

This hybrid POD-NN method has been first proposed in [S. Hesthaven et al. JCP 2018].

For a *non-linear* PDEs, one has the following discrete *non-linear system* to solve:

$$A^{\mu}(U_h)U_h = F^{\mu} \text{ with } U_h \in R^{NN} \tag{6.0.1}$$

By applying the same change of variables as in the linear case, one obtains the following reduced system:

$$\left(B_{rb}^T A^{\mu}(B_{rb}U_{rb})B_{rb}\right)\ U_{rb} = B_{rb}^T F^{\mu} \tag{6.0.2}$$

with $U_{rb} \in R^{N_{rb}}$, $B_{rb}U_{rb} \in R^{NN}$.

The goal is to obtain: $B_{rb}U_{rb} \approx U_h$.

The issue is here to handle the NN *non-linear* equations represented by the term $A^{\mu}(B_{rb}U_{rb})$.

During the last decades, a large litterature has tackled a variety of non-linear problems. The mathematical analyses are extremely challenging. A few semi-empirical methods have been developed. Recently to handle non-linear model terms, it has been proposed to combine classical RB methods, like POD, with ML techniques. This is what is done here.

Contents

6.1 Construction of the same RB $\Xi(x)$ as in POD

The **offline phase** of the POD method consisting to built up B_{rb} is performed exactly the same as in the linear case, see 5.1:

- The parameter space P is digitalized by choosing M values of parameters $(\mu_1, \ldots, \mu_M)$.
- The corresponding M snapshots (HR FE solutions) $u_{\mu,m}$, $1 \leq m \leq M$, are computed.
  They are stored in the $(NN \times M)$-snapshot matrix $\mathbf{S} = (\mathbf{u}_{\mu,1}| \cdots |\mathbf{u}_{\mu,M})$.
- The $M \times M$-correlation matrix is built up: $\mathbf{C} = \mathbf{S}^T \mathbf{N}_h \mathbf{S}$.
  Its eigenelements are computed: $(\lambda_n, \psi_n) \in \mathbf{R} \times \mathbf{R}^M$, $\|w_n\|_V = 1$, $1 \leq n \leq M$.
- N_{rb} is set from the ratio $R(n)$, see (5.2.29).
- The N_{rb} largest left singular vectors of $\mathbf{S}$ are deduced from the eigenvectors ψ_n:

$$l_n = \frac{1}{\sqrt{\lambda_n}} \mathbf{S} \psi_n \qquad 1 \leq n \leq N_{rb} \tag{6.1.1}$$

- The $(NN \times N_{rb})$-Reduced Basis matrix B_{rb} is built up as:

$$B_{_{rb}} = \left(l_1 | \cdots | l_{_{N_{rb}}} \right) \tag{6.1.2}$$

The matrix B_{rb} encodes the change of variable from the FE basis $\Phi(x)$ to the RB $\Xi(x)$.

The reduced dimension basis $V_{rb} = V_{POD}$ is the optimal one to represent the snapshots (Proposition 71), whatever if the undelying PDE is linear or not.

Indeed, the estimation (5.2.28) is purely algebraic and based on the snapshots set (dataset) $\mathbf{S}$.

6.2 Learning the coefficients of each snapshot in the RB $\Xi(x)$

Next, the idea of the method is as follows.

Given $V_{rb} = span\ \Xi(x)$, equivalently the matrix B_{rb}, see (5.2.23), *a Deep Neural Network (DNN) is trained to learn the snapshot coefficients in $\Xi(x)$.*

More precisely, given the set of parameters $(\mu_1, \ldots, \mu_M)$, given the corresponding snapshots set $\mathbf{S} = (\mathbf{u}_{\mu,1}| \cdots |\mathbf{u}_{\mu,M})$, given the RB matrix B_{rb}, a DNN is trained to learn the following map:

$$\boxed{F : \mu \in P \mapsto \left(B_{rb}^T U_h^\mu \right) \in R^{N_{rb}}} \tag{6.2.1}$$

where U_h^μ denotes the dof vector of the HR FE solution $\mathbf{u}_\mu$, $\mathbf{u}_\mu \in V_h$.

This mapping F represents the HR FE solution $u(\mu)$ of the PDE in the RB V_{rb}.

From a large set of (input(s), output(s)) pairs (called "examples" or "samples" in the ML jargon), a trained deep NN may enable to simulate the non-linear map F , in an "interpolation mode".

Here, the learning samples are the M (parameter values μ_m, snapshots $\mathbf{u}_{\mu_m}$).

Training a DNN necessitates to compute a very large number M of snaphots, typically $M = 10^p$ with $p \approx 3-4$ in $1D$ only (i.e. for $P \subset \mathbf{R}$).

The offline phase is very CPU time consuming too.

After training, the DNN outputs denoted by $\mathcal{N}(\mu)$, are supposed to satisfy:

$$\mathcal{N}(\mu_m) \approx \left(B_{rb}^T U_h^{\mu_m}\right) \text{ for each snapshot parameter value } \mu_m. \tag{6.2.2}$$

Remark 74. This ML-based projection method, Eq. 6.2.1, can be applied to non linear PDEs.

Moreover, this approach is relevant too in a FV context.

Indeed, in a FV context, no natural reduced bilinear form (5.1.9) is available at online phase.

6.3 Online phase: definition of U_{rb}

Given a new value of parameter μ, the reduced basis solution U_{rb}^{μ} is simply obtained by performing the trained Neural Network with μ as a new input value, see (6.2.1).

Indeed, the reduced dimension solution U_{rb}^{μ} , $U_{rb}^{\mu} \in R^{N_{rb}}$, is here defined as:

$$\boxed{U_{rb}^{\mu} = \mathcal{N}(\mu)} \tag{6.3.1}$$

Recall that U_{rb}^{μ} reads in the HR FE basis as:

$$U_{rb,NN}^{\mu} = B_{rb} U_{rb}^{\mu} = B_{rb} B_{rb}^T U_h^{\mu} \ , \ U_{rb,NN}^{\mu} \in R^{NN}. \tag{6.3.2}$$

6.4 Summary of the method & remarks

The **offline phase** is the same as the in the standard POD method (adopted for linear PDEs) plus the training phase of a NN to learn the map:

$$\mu \in P_h \mapsto \left(B_{rb}^T U_h^{\mu}\right) \in R^{N_{rb}} \tag{6.4.1}$$

That is:

A sample in the parameter space P is chosen.

The HR solver is performed to computed the M corresponding HR solutions. This constitutes the reference snapshots set $\mathbf{S}$. M must be large enough to well perform the NN.

The (spd dense) correlation matrix $\mathbf{C}$ is built and its eigenelements are computed (equivalently a SVD of the snapshot matrix $\mathbf{S}$).

The reduced dimension basis $V_{rb} = V_{POD}$ of dimension N_{rb} given, is the optimal basis to represent the snapshots, see Proposition 71.

The N_{rb} first most influencial modes of $\mathbf{C}$ are retained. The RB B_{rb} is stored. The resulting $(NN \times N_{rb})$-Reduced Basis matrix B_{rb} follows.

Like in the standard POD method, this phase may be done by employing any numerical methods: FE, FV, FD etc.

The **on-line phase** simply consists to compute a RB solution U_{rb}^{μ} given a new parameter value μ by performing the trained NN. Therefore, the computations can be done in real-time.

We obtain:

$$U_{rb}^{\mu} = \mathcal{N}(\mu) \tag{6.4.2}$$

We *expect* to have:

$$B_{rb} U_{rb}^{\mu} \approx U_h^{\mu} \text{ equivalently } U_{rb}^{\mu} \approx B_{rb}^T U_h^{\mu} \tag{6.4.3}$$

Finally let us point out that:

- The POD-NN method remains the same for linear or non-linear PDEs, affinely parametrized or not.
- The POD-NN method is relevant for non Galerkin-based solvers like Finite Volume schemes, even if the PDE is linear.
  It can be developed from any computational codes (employed as black box) *relying on any numerical methods (FE, FV, FD etc).*
- Moreover, since the online phase does rely on any matrix-based system, the complete POD-NN method is *non-intrusive.*
- Numerical experiments show that this *empirical method* provides reduced models with acceptable accuracies, at least in a few cases...

6.5 Numerical results

We present below two examples. The first example relies on the classical linear unsteady convection-diffusion equation, non-affinely parametrized. In this case, both the POD method and the POD-NN method can be applied. They are then compared.

The second example relies on a more complex model: the 2D shallow-water flow model which is solved by a Finite Volume (FV) scheme. This model is widely employed for instance to simulate flood plain dynamics (inundations).

6.5.1 Unsteady convection-diffusion equation

We consider a 2D parameter space: $\mathbf{P}_h \in \mathbb{R}^2$, $\mu = (\mu_1, \mu_2)$ with $\mu_1, \mu_2 \in \mathbb{R}$.

The model is the following unsteady advection-diffusion BVP in $\Omega = (-1,1) \times (-1,1)$:

$$\begin{cases} \partial_t u(\mu;t) - \mathrm{div}(\lambda(\mu_1)\nabla u(\mu;t)) + \boldsymbol{w} \cdot \nabla u(\mu;t) = f(\mu_2) & \text{in } \boldsymbol{Q}_T = (0,T) \times \Omega, \\ u(\mu;t) = 0 & \text{in } \Gamma_D, \\ -\lambda(\mu_1)\nabla u \cdot n = 0 & \text{in } \Gamma_N, \\ u_h(\mu;0) = u_0(\mu) & \text{a.e in } \Omega. \end{cases} \tag{6.5.1}$$

The boundary of Ω is split into two parts as:
$\Gamma_N = (-1,1) \times \{-1\} \cup (-1,1) \times \{1\} \cup \{1\} \times \{-1,1\}$ and $\Gamma_D = \{-1\} \times (-1,1)$.

This BVP is here *non-affinely parameterized* since we set:

- $$\lambda(\mu_1) = \exp(\mu_1 - 11), \tag{6.5.2}$$
  where $\mu_1 \in [\mu_{\min}, \mu_{\max}]$ with $\mu_{\min} = 1$, $\mu_{\max} = 10$.
- $$f(\mu_2) = A\cos(\mu_2 \mathrm{L} x), \tag{6.5.3}$$
  with $A = 10$, L a domain boundary lenght Ω ($L = 2$ here). We have: $\mu_2 \in [0, \frac{\pi}{L}]$.

  The parameter space $P = .$ The number of snapshots $M =$. The RB dimension $N_{rb} =$.

POD method results

POD-NN method results

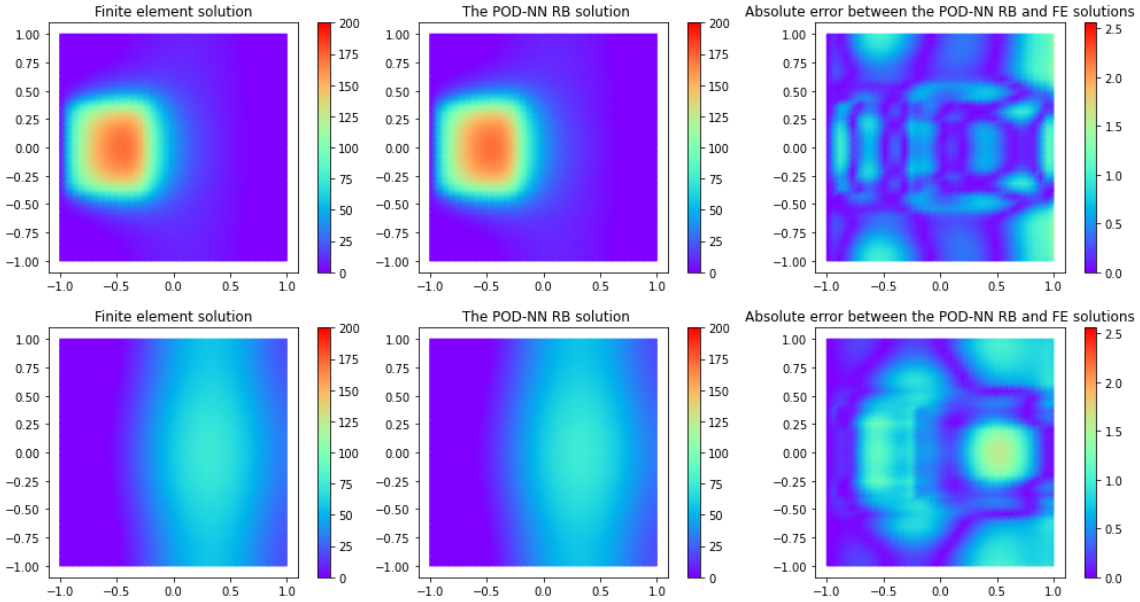


Figure 6.5.1: POD method. Linear PDE case with $\mu = (\mu_1, \mu_2)$, non-affine parameterization: $\lambda(\mu_1) = \exp(\mu - 11))$, $\mu_1 = 9.3$ and $f(\mu_2)$ given by 6.5.3 with $\mu_2 = 1.48$. (Left) The HR solution. (Middle) The RB solution with $N_{rb} =$. (Right) The absolute error between the FE and RB solutions.
(Top) At initial instant $t = 0$s. (Bottom) At time instant $t = 0.87$s.

6.5.2 2D Shallow-Water model

The model is the 2D Shallow Water (SW) system which is implemented in the computational code DassFlow[1].

The 2D SW model is a non-linear hyperbolic system whose the variables are the water depth $h(t, x, y)$ (m) and the discharge components $(q_x, q_y)(t, x, y)$ (m^3/s). Thus, the model output is $(h; q_x, q_y)(t, x, y)$.

The equations are detailed in the DassFlow documentation.

They are solved by a Finite Volume scheme.

The parameter μ is here bi-dimensional, $\mathbf{P} \subset \mathbb{R}^2$, defining the inflow hydrographs i.e. the BC at the incoming boundary, see Fig. 6.5.4.

Each parameter value is defined as $\mu = (t_{montee}, Q_{max})$, with t_{montee} the peak instant (s) and Q_{max} the maximum value of the inflow discharge (m^3/s). We consider: $t_{montee} \in [2343.08, 3843.08]$ and $Q_{max} \in [500., 3000.]$, see Fig. 6.5.3.

Therefore the parameter space $P = [2343.08, 3843.08] \times [500., 3000.]$.

Moreover, the number of snapshots $M =$ and the RB dimension $N_{rb} =$.

On Fig. 6.5.4, the flow model outputs, the POD-NN solution and the relative errors at $t = 5681.182$(s) (= 1h57min) are plotted.

[1] DassFlow computational code: https://www.math.univ-toulouse.fr/DassFlow

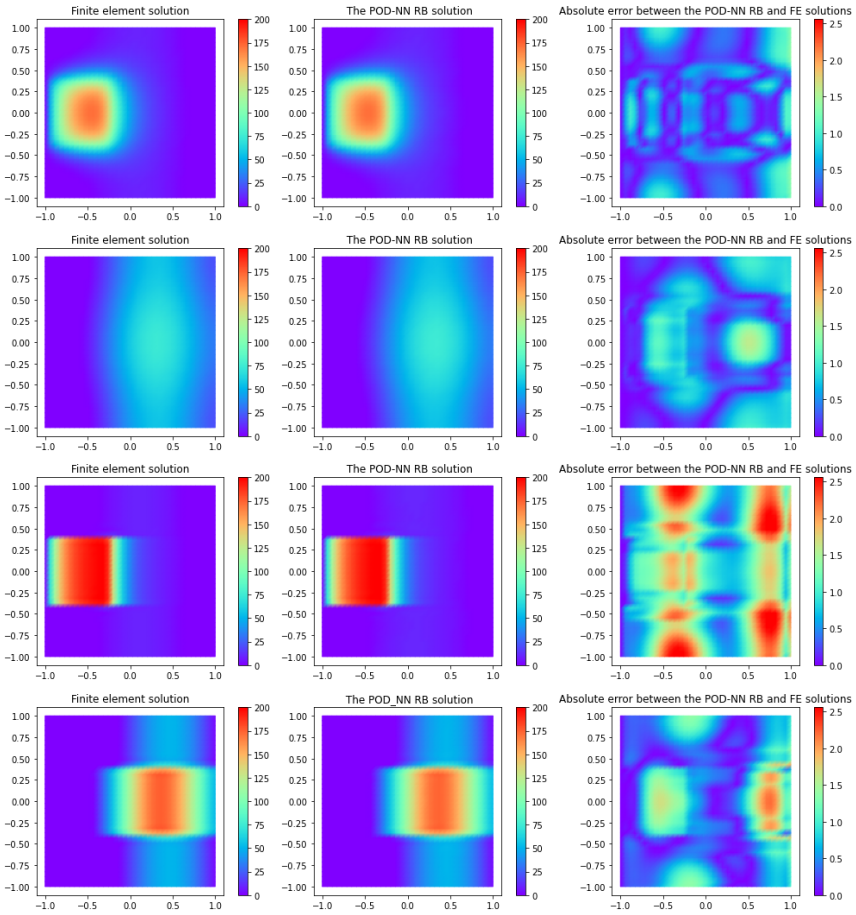


Figure 6.5.2: POD-NN method. Same case as above. (Left) The HR solution. (Middle) The POD-NN RB solution with $N_{rb} =$. (Right) The absolute error between the HR solution and the POD-NN solution.
Lines 1 and 2: $mu_1 = 9.3$ at $t = 0$ and $t = 0.87$. Lines 3 and 4: $mu_2 = 2.68$ at $t = 0$ and $t = 0.87$.

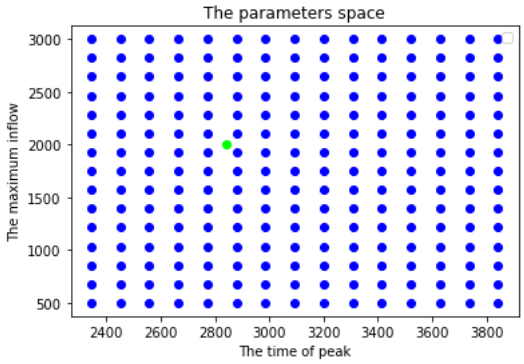


Figure 6.5.3: The 2-D parameter space. In green, the parameter value at online phase ($t_{montee} = 2843.08$s, $Q_{max} = 2000 m^3 s^{-1}$).

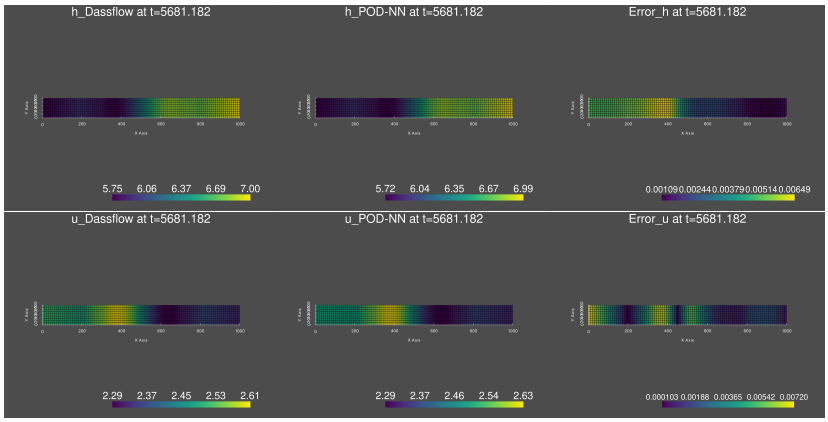


Figure 6.5.4: POD-NN method. 2DSW flow model with μ defined as follows: $t_{montee} = 2843.0$ and $Q_{max} = 2000$ (see Fig. 6.5.3).
(Line 1) The water depth h. (Line 2) The x-direction velocity u.
(Left) The HR model output. (Middle) The POD-NN solution. (Right) The relative error between both.

Chapter 7

Model reductions using Auto-Encoders (AE)*

This is a "to go further section".

This section has been written with the help of M. Allabou, PhD INSA-IMT 2021-24.

AEs have been recently employed to reduce non-linear BVP, see e.g. [Fresca et al. JSC 2021]. This is the idea briefly presented in this section.

Model reductions using AE constitutes a purely-data driven method. As a consequence no conservation law is respected by the reduced model. This lack of physical consistency can be an issue.

Purely-driven approaches should be improved in the next few years by taking into account additionnal physical constraints such as e.g. mass conservation, symmetries of solutions etc. We called them "hybrid AI" or "Physics-Informed Machine Learning".

Contents

7.1 Basic principles of AEs

Let us first recall what is an AE. An AE consists of two parts: the Encoder and the Decoder. Each part is a Feed-Forward Neural Network.

The AE is trained to approximate the function F^{AE} defined by:

$$\begin{array}{rcccc} F^{AE}(W_E, W_D; \cdot) & : & \mathbf{R}^{NN} & \rightarrow & \mathbf{R}^{NN} \\ & & X_h & \mapsto & \hat{X}_h = F^D \circ F^E(W_E, W_D; \cdot) \end{array}$$

with the goal to obtain:

$$\hat{X}_h \approx X_h \tag{7.1.1}$$

W_E (resp. W_D) denotes the parameters of the Encoder (resp. of the Decoder).
The functions F^D and F^E are such that:

$$\begin{array}{rcccc} F^E(W_E; \cdot) & : & \mathbf{R}^{NN} & \rightarrow & \mathbf{R}^{N_{rb}} \\ & & \boldsymbol{U}_h & \mapsto & \boldsymbol{U}_{rb} \\ F^D(W_D; \cdot) & : & \mathbf{R}^{N_{rb}} & \rightarrow & \mathbf{R}^{NN} \\ & & U_{rb} & \mapsto & \hat{X}_h \end{array}$$

The Encoder (resp. the Decoder) aims at approximating the function F^E (resp. F^D).

AEs are trained by solving the following optimization problem:

$$(W_E, W_D) = arg\ min_{(w_E, w_D)} \left(\hat{X}_h - F^D \circ F^E(w_E, w_D; X) \right) \tag{7.1.2}$$

That is $F^{AE} = (F^D \circ F^E)$ aims at approximating the Identity function.

AE are forced to reconstruct the input by preserving the "most relevant aspects" of the data.
AEs are considered as an unsupervised ML method. AEs have appeared in the 80s.

The Encoder is classically chosen as a Convolution Neural Network aiming at down-sampling the input variable U_h. In other words, the Encoder computes a projection of U_h onto a low-order space (the "latent space").

On the contrary, the Decoder aims at up-sampling the Encoder output U_{rb}, see Fig. 7.1.1. It is then defined as a deconvolution NN.

7.2 The fundamental result

It is shown e.g. in (Kunin-Bloom et al., 2019) that a basic two-layers linear AE is equivalent to the probabilistic PCA (that is the POD with Gausiann perturbed noise).

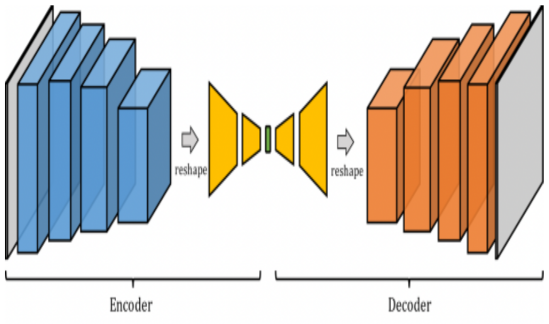


Figure 7.1.1: (Left) An Auto-Encoder (AE) architecture. Image source: Wikipedia (Right) A simple Linear Auto-Encoder (LAE) architecture.

Definitions

Definition 75. Let us consider a small dimension N_{rb}-dimensional vector $\boldsymbol{X}_{rb}$with $\boldsymbol{X}_{rb} \sim \mathcal{N}(0, I)$, a scalar value μ and a NN-dimensional vector $\boldsymbol{\epsilon}$ representing an error term, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.

We set:

$$\boldsymbol{X} = \boldsymbol{W_0 X_{rb}} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{7.2.1}$$

with $\boldsymbol{W_0}$ a $NN \times N_{rb}$-matrix.
X_{rb} is sometimes called a *probabilistic PCA model.*

Note that one has: $p(\boldsymbol{X}/\boldsymbol{X}_{rb}) = \mathcal{N}(\boldsymbol{X}/\ \boldsymbol{W_0 X_{rb}} + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$.

Definition 76. We call a Linear Auto Encoder (LAE) an AE with one Encoder layer only, and one Decoder layer only, with Identity as activation functions.
With the previous notations, we then have: $\boldsymbol{W_1} \in \mathcal{M}_{N_{rb} \times NN}$ and $\boldsymbol{W_2} \in \mathcal{M}_{NN \times N_{rb}}$.

The loss function to be minimized is then defined as:

$$\mathcal{L}(\boldsymbol{W_1}, \boldsymbol{W_2}) = ||\boldsymbol{X} - \boldsymbol{W_2 W_1 X}||_F^2$$

where $\| \cdot \|_F$ denotes the Frobenius norm.

The result of equivalence We have the following result, see [**?**].

Theorem 77. *Let us consider the following regularization of the LAE loss function:*

$$\mathcal{L}_\lambda(\boldsymbol{W_1}, \boldsymbol{W_2}) = ||\boldsymbol{X} - \boldsymbol{W_2 W_1 X}||_F^2 + \lambda(||\boldsymbol{W_1}||_F^2 + ||\boldsymbol{W_2}||_F^2)$$

Then, the critical points of $\mathcal{L}_\lambda$ coincide with the probabilistic PCA model.

In other words, a regularized LAE is nothing else than the POD with a probabilistic input variable.

7.3 Reducing PDE-based models using AEs

Following the equivalency result above (Theorem 77 which is valid for a simple LAE only), one may be inspired to attempt to reduce non-linear models using AEs... This is roughly the idea developed in [Fresca et al. JSC 2021].

The proposed algorithm is as follows.

Offline phase

- The parameter space P is digitalized: M snapshots (HR FE solutions) are computed and stored in the snapshot matrix $\mathbf{S}$. M has to be large enough to train an AE at next step.
  We denote as previously: $U_h^m = u_h(\mu_m)$, $1 \leq m \leq M$.

Figure 7.4.1: AE method. Solutions of the μ-parametrized linear advection-diffusion model **??**. Non-affine case with $\lambda(\mu) = exp(\mu_0(1+\mu))$.
Offline phase: $M = 40$ snapshots are considered.
(Left) The HR FE solution. (Middle) The AE solution (projected on the mesh by using B_{rb}). (Right) The absolute difference between the two solutions (error).
Computations performed by M. Allabou (INSA-IMT, 2021).

- The AE is trained by solving the optimization problem:

$$\min_{(W_D, W_E)} ||U_h^\mu - \boldsymbol{F}^D\left(W_D; \boldsymbol{F}^{\boldsymbol{E}}(W_E; U_h^\mu)\right)||_2^2$$

  The Encoder provides a reduced vector U_{rb}^μ hopefully approximating (in some sense) the HR FE solution U_h^μ.

- Given the numerous examples (μ, U_{rb}^μ) obtained at the previous step, another NN representing the following map is trained:

$$\begin{array}{rccc} G(W_G;\cdot) & : & \mathbf{R}^M & \rightarrow & \mathbf{R}^{N_{rb}} \\ & & \mu & \mapsto & \boldsymbol{U}_{rb}^\mu \end{array} \tag{7.3.1}$$

Online phase

- Given a new parameter value μ, the trained AE is performed to obtain the reduced dimension vector U_{rb}^μ.

Note that here no approximating result insures that the reduced vector U_{rb}^μ is satisfying approximation of the HR FE solution U_h^μ...

7.4 Numerical results

Numerical results are shown for the same BVP as previously (Section X): the steady-state linear convection-diffusion equation.

The model is non-affinely parametrized through the diffusivity coefficient $\lambda(\mu)$, $\lambda(\mu) > 0$ a.e.