ICMC-USP

Lecture Notes
Curso de Difussão

Automatic solution of PDEs with the Finite Element platform `FEniCSx`

A hands-on course

© Roberto F. Ausas & Igor A. Baratta

January 29, 2023

Instituto de Ciências Matemáticas e de Computação
University of São Paulo

Contents

Preface

The finite element method (FEM) is by now one of the most popular methods for numerically solving Partial Differential Equations (PDEs) in science, engineering and applied mathematics. There are a number of good reasons for this:

- ☺ For **elliptic** and **parabolic** problems[1], the FEM provides very accurate solutions;
- ☺ It is **general**, not restricted to linear problems, or to isotropic problems, or to any subclass of mathematical problems;
- ☺ It is **geometrically flexible**, complex domains are quite easily treated, not requiring adaptations of the method itself;
- ☺ It is **easy to code**, and the coding is quite problem-independent. Boundary conditions are much easier to deal with than in other methods;
- ☺ It is **robust**, because in most cases the mathematical problem has an underlying variational structure (energy minimization, for example).

1: **Elliptic problems**: Stationary diffusion, heat conduction, fully developed laminar flows in ducts, linear elasticity. **Parabolic problems**: Transient diffusion or heat conduction, chemical kinetics, fluid dynamics.

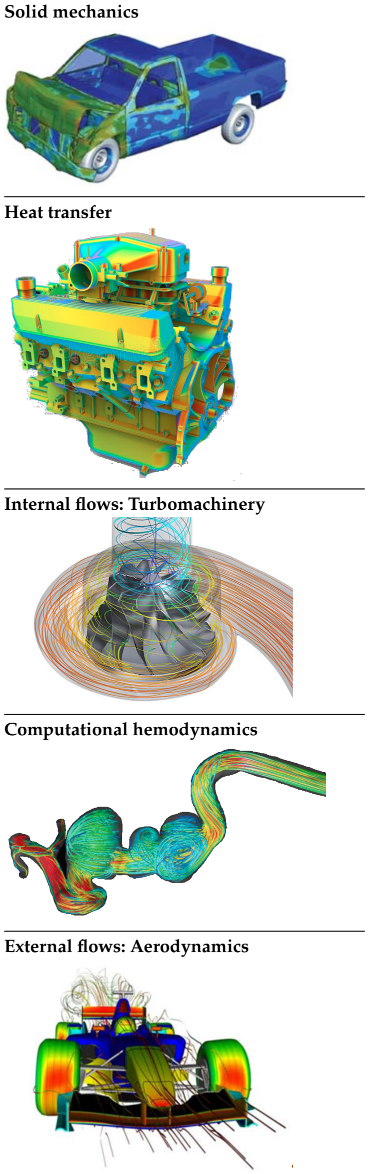


Figure 0.1: Examples solved by FEM.

0.1 What This Course Covers

The main objective of this course is to introduce the student to the `FEniCSx` open source platform for solving PDEs using the Finite Element method, through practical examples that arise in Solid and Fluid Mechanics. The `FEniCS` project is a high-end open source platform that allows you to efficiently automate the resolution of partial differential equations (PDEs) by the finite element method (FEM). By using a domain-specific language called UFL (Uniform Form Language) it is possible to write the variational formulation of complex problems governed by PDEs and their discretization by the FEM. Despite using a high-level language, the library allows solving problems efficiently with parallel computing, since it generates code in C that is compiled at runtime to perform the assembly of matrices and vectors that emerges when the FEM is applied.

This course consists of a series of practical lectures to demonstrate the use of the software and its potential to solve some relevant 2nd order PDEs of interest in fluid and solid mechanics. Although the main focus of the course is practical (it will be a hands-on course), some mathematical and theoretical aspects will be recalled so as to provide the necessary context to understand the physical/mathematical problems considered and their numerical resolution by the finite element method. The target audience for the course is primarily graduate students, however, advanced undergrads in applied mathematics, physics or engineering courses are welcome to take the course. Students are expected to have minimal knowledge about PDEs and the physical models behind the problems to be solved. Some previous background about discretization methods for PDEs is also desirable.

The course will be divided into 5 lectures with the following topics to be covered:

Lecture |01| Introduction:

- Examples of PDEs in fluid and solid mechanics
- Preliminary notions on the Finite Element Method
- The `FEniCSx` platform
- A first example: The heat conduction equation

Lecture |02| Classical problems:

- Transient heat conduction equation
- The elasticity problem

Lecture |03| Mixed problems:

- Coupled thermo-elasticity problem
- The Navier-Stokes equations

Lecture |04| Advanced topics:

- Darcy's flow: $H(\text{div})$ formulations
- Acoustics: The Helmholtz equation

Lecture |05| High Performance Computing with `FEniCSx`:

- Notions on parallel computing
- Scalability: Weak and Strong scaling

Each lecture will be accompanied by a `Colab Notebook` available at the public repository `USP-FEniCSx-Course`, in which some of the relevant mathematical concepts that are part of this document will be recalled. At the end of each lecture, there is a **Homework assignment** that must be presented the day after to the corresponding lecture.

Disclaimer

The following notes are not intended to be a comprehensive exposition of the theory and practice of the finite element method, but only a summary of some relevant concepts of interest for this course. For a complete description, the student is referred to [1–4].

RFA & IAB

INTRODUCTION 1

1.1 Motivating examples in solid and fluid mechanics

We begin by recalling some prototypical examples of PDEs we aim to solve by the finite element method. We introduce things with a relatively informal language and provide some physical interpretation of the different quantities and processes involved.

1.1.1 Poisson's equation

The simplest second order PDE we will consider in this course is Poisson's equation that models several physical phenomena, such as, heat conduction, mass transport by diffusion or even fully developed flows in ducts as we will see later on in this chapter. The problem reads: Given a region $\Omega \subset \mathbb{R}^d$, $d = 1, 2$ or 3 with boundary $\partial\Omega$, find u such as[1]

$$\left\{ \begin{array}{rcll} -\nabla \cdot (\mu(\mathbf{x})\nabla u(\mathbf{x})) & = & f(\mathbf{x}) & \mathbf{x} \in \Omega \\ \\ u(\mathbf{x}) & = & g(\mathbf{x}) & \mathbf{x} \in \partial\Omega \end{array} \right. \tag{1.1}$$

where the source term $f : \Omega \longrightarrow \mathbb{R}$ is a given function and the boundary data $g : \partial\Omega \longrightarrow \mathbb{R}$ is also a given function. This problem falls into the cathegory of **elliptic** problems. The scalar field u represents different physical quantities depending on the problem. Notice that if μ is a constant, the left hand side becomes the Laplace equation, i.e.,

$$\nabla \cdot (\mu(\mathbf{x})\nabla u(\mathbf{x})) = \mu\, \nabla^2 u(\mathbf{x}) \tag{1.2}$$

Of importance to us is a variant of this problem which includes flux boundary conditions over all or in some part of $\partial\Omega$ (ommiting the $\mathbf{x}$ dependence for simplificity of notation)

$$\left\{ \begin{array}{rcll} -\nabla \cdot (\mu\nabla u) & = & f & \text{in } \Omega \\ \\ u & = & g & \text{on } \Gamma_D \\ \\ -\mu\, \nabla u \cdot \check{\mathbf{n}} & = & h & \text{on } \Gamma_N \end{array} \right. \tag{1.3}$$

where $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$ (see Figure 1.2). The so called mixed formulation introduces an additional field to represent the flux of

1: For $d = 2$, recall from Calculus, the *nabla* operator. In cartesian coordinates:

- The gradient of a scalar-valued function
$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right)$$
- The divergence of a vector-valued function
$$\nabla \cdot \mathbf{F} = \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2}$$

Figure 1.1: Siméon Denis Poisson (France, 1781–1840).

the quantity of interest

$$
\begin{cases}
\nabla \cdot \boldsymbol{\mathcal{J}} = f & \text{in } \Omega \\
\boldsymbol{\mathcal{J}} = -\mu \nabla u & \text{in } \Omega \\
u = g & \text{on } \Gamma_D \\
\boldsymbol{\mathcal{J}} \cdot \check{\mathbf{n}} = h & \text{on } \Gamma_N
\end{cases}
\tag{1.4}
$$

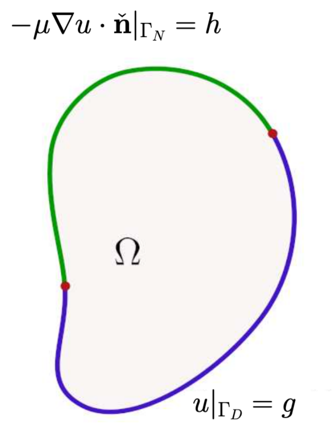


Figure 1.2: Domain Ω whose boundary is partitioned into a Dirichlet and a Neumann part.

Although both problems are equivalent in the exact setting we are focused now, in the discrete setting things may differ a lot, very different approaches being needed to deal with each formulation. The equation defining the relation between $\boldsymbol{\mathcal{J}}$ and u is what we call **constitutive law**. Finally, it is instructive to interpret the different quantities according to the physical problem being solved, as summarized in table Table 1.1.

Table 1.1: Physical interpretation of quantities in Poisson's problem. Units in SI.

Physical problem	u	f	$\boldsymbol{\mathcal{J}}$	μ
Heat conduction	Temperature [$^\circ K$]	Heat source [$\frac{\mathrm{W}}{\mathrm{m}^3}$]	Heat flux [$\frac{\mathrm{W}}{\mathrm{m}^2}$]	Conductivity [$\frac{\mathrm{W}}{\mathrm{m}^\circ\mathrm{K}}$]
Mass diffusion	Mass [$\frac{\mathrm{mol}}{\mathrm{m}^3}$]	Mass source [$\frac{\mathrm{mol}}{\mathrm{m}^3\mathrm{s}}$]	Mass flux [$\frac{\mathrm{mol}}{\mathrm{m}^2\mathrm{s}}$]	Diffusivity [$\frac{\mathrm{m}^2}{\mathrm{s}}$]
Flow in ducts	Velocity [$\frac{\mathrm{m}}{\mathrm{s}}$]	Pressure gradient [$\frac{\mathrm{N}}{\mathrm{m}^3}$]	Shear stress [Pa]	Viscosity [Pa · s]

1.1.2 Transient heat conduction

The transient or unsteady version of the previous problems includes an additional term involving the rate of change of the physical quantity of interest. For the heat conduction problem the equation reads

$$
\begin{cases}
a(\mathbf{x},t)\,\dfrac{\partial u(\mathbf{x},t)}{\partial t} - \nabla \cdot (\mu(\mathbf{x},t)\nabla u(\mathbf{x},t)) = f(\mathbf{x},t) & \mathbf{x} \in \Omega,\ \ t \in [0,T] \\
u(\mathbf{x},t) = g(\mathbf{x},t) & \mathbf{x} \in \partial\Omega,\ \ t \in [0,T] \\
u(\mathbf{x},0) = u_0(\mathbf{x}) & \mathbf{x} \in \partial\Omega
\end{cases}
\tag{1.5}
$$

where now an initial condition for the scalar unknown (e.g. the temperature) has been provided. This problem falls into the cathegory of **parabolic** problems. The parameter a in front of the time derivative has different meanings depending on the problem at hand (see Table 1.2)

The different forms of this problem, such as the one with a flux boundary condition or the mixed formulation can be formulated in a similar way as in the stationary case.

Table 1.2: Physical interpretation of the a factor in front of $\partial_t u$ in the transient Poisson's problem. Units in SI.

Problem	Factor a	Description
Heat conduction	$\rho\, c$	[$\frac{\mathrm{Kg}}{\mathrm{m}^3}\ \frac{\mathrm{J}}{\mathrm{Kg}\,^\circ\mathrm{K}}$]
Mass diffusion	1	[-]
Flow in ducts	ρ	[$\frac{\mathrm{Kg}}{\mathrm{m}^3}$]

1.1.3 The equations of linear elasticity

This is the prototypical example in solid mechanics in which we describe the deformation of a solid domain. If we consider a domain $\Omega \subset \mathbb{R}^d$ its shape is defined by a map $\chi : \Omega \longrightarrow \mathbb{R}^d$, such that for any $\mathbf{x} \in \Omega$ we write

$$\chi(\mathbf{x}, t) = \mathbf{x} + \mathbf{u}(\mathbf{x}, t) \tag{1.6}$$

where the vector field $\mathbf{u} : \Omega \longrightarrow \mathbb{R}^d$ represents the displacement at the fixed location in space $\mathbf{x}$. The PDE governing the problem follows from Newton's dynamical equilibrium equations:

$$\underbrace{\rho(\mathbf{x})\frac{\partial^2 \mathbf{u}(\mathbf{x}, t)}{\partial t^2}}_{\text{mass} \times \text{acceleration}} - \underbrace{\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t)}_{\text{Forces}} \tag{1.7}$$

where the second order time derivative in the left hand side corresponds to the acceleration and $\mathbf{f}$ in the right hand side to the body forces (e.g., gravity), $\boldsymbol{\sigma}(\mathbf{x}, t)$ defines the stresses in the body at location $\mathbf{x}$, which under the small deformation assumption is given by the **constitutive law**

$$\boldsymbol{\sigma} = \underbrace{\mu\left(\nabla \mathbf{u} + \nabla^{\intercal} \mathbf{u}\right)}_{2\mu\, \varepsilon(\mathbf{u})} + \lambda(\nabla \cdot \mathbf{u})\, \mathbf{I} \tag{1.8}$$

where λ and μ are known material parameters (the Lamé parameters) and $\mathbf{I}$ is the identity matrix of $d \times d^2$. In the stationary case this is an elliptic problem in which the unknown field is a vector-valued function. In order to have a well-posed problem the displacement must be restricted in some part of the boundary (say, $\Gamma_{\mathbf{u}}$) so as to eliminate rigid body motions (i.e., translations and rotations). Also, a surface force distribution can be applied on the rest of the boundary $\Gamma_{\mathcal{F}} = \partial\Omega \setminus \Gamma_{\mathbf{u}}$.

2: For $d = 2$, in cartesian coordinates

- The stress tensor is the matrix
$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$
- The divergence of $\boldsymbol{\sigma}$ is the vector
$$\nabla \cdot \boldsymbol{\sigma} = \begin{bmatrix} \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} \\ \frac{\partial \sigma_{21}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} \end{bmatrix}$$
- The gradient of the vector field $\mathbf{u}$ is the matrix
$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} \end{bmatrix}$$

The extension for $d = 3$ is immediate.

$$\begin{cases} -\nabla \cdot \left(2\mu\varepsilon(\mathbf{u}) + \lambda\left(\nabla \cdot \mathbf{u}\right)\mathbf{I}\right) & = & \mathbf{f} & \text{in } \Omega \\ \mathbf{u} & = & \mathbf{u}_D & \text{on } \Gamma_{\mathbf{u}} \\ \left(2\mu\varepsilon(\mathbf{u}) + \lambda\left(\nabla \cdot \mathbf{u}\right)\mathbf{I}\right) \cdot \check{\mathbf{n}} & = & \mathcal{F} & \text{on } \Gamma_{\mathcal{F}} \end{cases} \tag{1.9}$$

where $\mathbf{u}_D$ and $\mathcal{F}$ are given functions.

1.1.4 The incompressible Navier-Stokes problem

A natural extension of the previous problem are the Navier-Stokes equations. Now, we have two primary variables, namely, the velocity field $\mathbf{u}(\mathbf{x}, t)$ and the pressure field $p(\mathbf{x}, t)$. We consider an Eulerian formulation, so, $\mathbf{u}$ is the velocity at fixed position $\mathbf{x}$ in space. Also, we restrict ourselves to the particular case of incompressible fields, i.e.,

$$\nabla \cdot \mathbf{u} = \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3} = 0 \tag{1.10}$$

The stresses in the fluid are described by the Cauchy stress tensor

$$\boldsymbol{\sigma}(\mathbf{x}, t) = -p(\mathbf{x}, t)\mathbf{I} + \boldsymbol{\sigma}^*(\mathbf{x}, t) \tag{1.11}$$

which is the sum of a volumetric part

$$-p(\mathbf{x}, t)\mathbf{I} = \begin{bmatrix} -p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & -p \end{bmatrix} \tag{1.12}$$

and a deviatoric part

$$\boldsymbol{\sigma}^*(\mathbf{x}, t) = 2\,\mu\varepsilon(\mathbf{u}) = \mu\,(\nabla\mathbf{u} + \nabla^\intercal\mathbf{u}) \tag{1.13}$$

where μ is the viscosity of the fluid. Now, we write the momentum equation in the so called convective form

$$\rho\frac{D\mathbf{u}}{Dt} - \nabla\cdot\boldsymbol{\sigma} = \rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \nabla\cdot(2\,\mu\varepsilon(\mathbf{u})) + \nabla p = \mathbf{f} \tag{1.14}$$

where the first (non-linear) term introduces the convective acceleration, the second term the viscous effects and the third term the forces due to pressure gradients. As in the previous case, in the left hand side we have the body forces (e.g., $\mathbf{f} = -\rho\, g\check{\mathbf{e}}_3$). Given an initial velocity field $\mathbf{u}_0$ that satisfies the incompressibility constraint, the Navier-Stokes problem reads

$$\begin{cases} \rho\left(\dfrac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \nabla\cdot(2\,\mu\varepsilon(\mathbf{u})) + \nabla p = \mathbf{f} & \text{in } \Omega \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega \\ \mathbf{u} = \mathbf{u}_D & \text{on } \Gamma_D \\ [-p\mathbf{I} + 2\,\mu\,\varepsilon(\mathbf{u})]\cdot\check{\mathbf{n}} = \boldsymbol{\mathscr{F}} & \text{on } \Gamma_N \end{cases} \tag{1.15}$$

where $\mathbf{u}_D$ is a given function on Γ_D and $\boldsymbol{\mathscr{F}}$ is a given function on Γ_N.

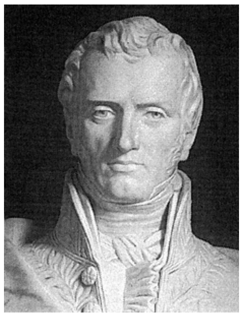


Figure 1.3: Claude Louis Marie Henri Navier (France, 1785–1836).

Figure 1.4: George Stokes (Ireland(1819)–England(1903)).

1.1.5 The Helmholtz's equation

The scalar Helmholtz equation models time-harmonic propagation of linear acoustic waves, however it can be a good approximation for time-harmonic electromagnetic waves in some settings.

It can be derived from the linear wave equation

$$\nabla^2 p = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} \tag{1.16}$$

assuming that the pressure field is a steady state harmonic field:

$$p = \Re\,(\tilde{p}\exp\,(j\omega t)) \tag{1.17}$$

where $\tilde{p}$ is a complex valued field, ω, is the angular frequency of the field in radians per second. By substituting Eq. (1.17) into the wave equation we obtain the Helmholtz equation:

$$\nabla^2\tilde{p} + k^2\tilde{p} = 0 \tag{1.18}$$

where the constant $k = \omega/c$ is called the wavenumber. The Helmholtz equation stands therefore for monochromatic waves, or waves of a given frequency ω. Because the Helmholtz equation is time-independent, it could be solved more efficiently compared to the time-dependent wave equations used for modeling acoustics in the time domain. However finding a good solver and preconditioner for this type of equation is still an open problem.

The Helmholtz equation is deceivingly similar to the reaction-diffusion (i.e., $-(\nabla^2 u - \eta^2 u) = 0$), for sufficiently large k the underlying operator is not positive definite.

1.2 A variational formulation

The simplest example we can consider is the Poisson's problem

$$\begin{cases} -\nabla \cdot (\mu \nabla u) & = & f & \text{in } \Omega \\ u & = & g_D & \text{on } \partial\Omega_D \\ -\mu \nabla u \cdot \check{\mathbf{n}} & = & g_N & \text{on } \partial\Omega_N \end{cases} \tag{1.19}$$

We will proceed quite informally letting some details for more advanced courses. The idea first is to cast this problem into variational form, which amounts to multiply by a sufficiently regular **test** function v that satifies the boundary condition, i.e.,

$$v(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial\Omega_D \tag{1.20}$$

and integrate over Ω, yielding

$$-\int_\Omega \nabla \cdot (\mu(\mathbf{x}) \nabla u(\mathbf{x}))\ v(\mathbf{x})\, dx = \int_\Omega f(\mathbf{x})\, v(\mathbf{x})\, dx \tag{1.21}$$

From our Calculus course we know that (ommiting the $\mathbf{x}$ dependence to simplify notation)

$$\nabla \cdot (\mu \nabla u\, v) = \nabla \cdot (\mu \nabla u)\ v + \mu \nabla u \cdot \nabla u$$

We thus have

$$\int_\Omega \mu \nabla u \cdot \nabla v\, dx - \int_\Omega \nabla \cdot ((\mu \nabla u)\, v)\, dx = \int_\Omega f\, v\, dx \tag{1.22}$$

For the second term in the left hand side we apply Gauss theorem

$$\int_\Omega \nabla \cdot ((\mu \nabla u)\, v)\, dx = \int_{\partial\Omega} v\, (\mu \nabla u) \cdot \check{\mathbf{n}}\, ds \tag{1.23}$$

which is identically zero on $\partial\Omega_D$ by Equation 1.20, yielding[3]

$$\int_\Omega \mu \nabla u \cdot \nabla v\, dx = \int_\Omega f\, v\, dx - \int_{\partial\Omega_N} g_N\, v\, ds \tag{1.24}$$

3: What we have done is also called integration by parts. Also notice that the boundary of Ω must have some regularity for the Gauss theorem to be applicable.

Let us agree that this should be valid for any v belonging to some space of functions $v : \Omega \longrightarrow \mathbb{R}$ (say $V(\Omega)$) that satisfy the homogeneous boundary conditions at $\partial\Omega_D$, for which the integrals in the left and right hand sides at least don't blow up. We may formally write what we call from now on the weak[4] form of the problem

4: The problem is said to be in **weak** form because in this formulation we require less differentiability from u in contrast to the PDEs in **strong** form for which we require u to have at least continuous second order derivatives if we pretend the equation to be satisfied pointwise.

Weak form of Poisson's problem

Find $u \in V(\Omega)$ such that

$$\begin{cases} \underbrace{\int_\Omega \mu(\mathbf{x})\, \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x})\, dx}_{a(u,v)} = \underbrace{\int_\Omega f(\mathbf{x})\, v(\mathbf{x})\, dx - \int_{\partial\Omega_N} g_N(\mathbf{x})\, v(\mathbf{x})\, ds}_{\ell(v)} \\ \forall v \in V(\Omega). \end{cases} \tag{1.25}$$

What appears in the lhs is a bilinear form $a(\cdot, \cdot)$, whereas in the rhs we have a linear form $\ell(\cdot)$, two objects we will encounter frequently in the course.

1.3 Some important function spaces

The main focus of this course is on the practical aspects of the FEM, however, it is useful to recall a few definitions of function spaces that are very often found when solving PDEs by the finite element method.

The first example is the space $C^k(\Omega)$ defined as the set of all real-valued functions that are continuous and have its partial derivatives up to order k also continuous on Ω, particular cases being

$$C^0(\Omega) = \{f : \Omega \longrightarrow \mathbb{R} : f \text{ is continuous } \forall x \in \Omega\} \doteq C(\Omega) \tag{1.26}$$

and

$$C^\infty(\Omega) = \{f : \Omega \longrightarrow \mathbb{R} : f \text{ has continuous derivatives of all orders } \forall x \in \Omega\} \tag{1.27}$$

Another example that we will encounter frequently in this course is the space of square integrable functions

$$L^2(\Omega) = \{f : \Omega \longrightarrow \mathbb{R} : \int_\Omega f^2\, dx < +\infty\} \tag{1.28}$$

and the space of square integrable functions whose derivatives are also square integrable

$$H^1(\Omega) = \{f : \Omega \longrightarrow \mathbb{R} : f \in L^2(\Omega) \text{ and } \int_\Omega (\nabla f \cdot \nabla f)\, dx < +\infty\} \tag{1.29}$$

which is quite relevant to us [5].

5: In Equation 1.29, since $\nabla f : \Omega \rightarrow \mathbb{R}^d$ (is a vector–valued function), the symbol "$\cdot$" stands for the inner (or usual "dot") product of vectors, i.e., $\nabla f(x) \cdot \nabla f(x) = \sum_{i=1}^{d} (\nabla f)_i(x)\, (\nabla f)_i(x) = \sum_{i=1}^{d} \frac{\partial f}{\partial x_i}(x)\, \frac{\partial f}{\partial x_i}(x)$.

Both are Hilbert spaces, i.e., is a vector space equipped with an inner product, that is also a complete metric space with respect to the norm induced by the inner product.

The inner products are the $L^2(\Omega)$-inner product

$$(v,w)_{L^2(\Omega)} = \int_\Omega v\,w\,dx \tag{1.30}$$

and the $H^1(\Omega)$-inner product

$$(v,w)_{H^1(\Omega)} = \int_\Omega (v\,w + \nabla v \cdot \nabla w)\,dx \tag{1.31}$$

which induce the norms

$$\|v\|_{L^2(\Omega)} = \left(\int_\Omega v^2\,dx\right)^{\frac{1}{2}}$$

and the $H^1(\Omega)$-inner product

$$\|v\|_{H^1(\Omega)} = \left(\int_\Omega \left(v^2 + \nabla v \cdot \nabla v\right)dx\right)^{\frac{1}{2}}$$

As a final example consider a subspace of the previous, made up of functions that are equal to zero on the boundary $\Gamma = \partial\Omega$

$$H_0^1(\Omega) = \{f \in H^1(\Omega) : f|_\Gamma = 0\},$$

This one is also of particular interest to us, since it incorporates a **boundary condition**[6]. We will encounter such spaces frequently throughout. Moreover, we will need to construct finite dimensional subspaces of these spaces for the implementation of the FEM.

6: We use the notation $f|_\Gamma$ to denote the restriction of a function f to the boundary Γ, i.e., the values of $f(x),\ x \in \Gamma$. To do this properly, we must introduce the **trace** operator $\gamma : C(\bar{\Omega}) \to C(\Gamma),\ \gamma(f) = f|_\Gamma$.

1.4 The Galerkin method

A key ingredient of the finite element method, is the Galerkin method (also referred to as Ritz-Galerkin) , which takes as starting point the variational formulation. Let recall the abstract variational problem: Find $u \in V$ such that

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

of which we will see several examples in following lectures. The idea behind the Galerkin method is simply to replace the space V by a finite dimensional space $V_h \subset V$. The space V_h can be spanned by a set of linearly independent functions in V, $\{\phi_1, \phi_2, \ldots, \phi_n\}$[7], so, we write a discrete variational formulation:

7: Recall, the subindex h is used to denote finite dimensional or discrete spaces. The h in the finite element method is related to the refinement of a partition of Ω (the mesh), such that $h \to 0$ as $n \to \infty$.

VF$_h$: Find $u_h \in V_h$ such that

$$a(u_h, v_h) \quad = \quad \ell(v_h) \quad \forall\, v_h \in V_h \tag{1.32}$$

The interesting thing about the Galerkin method is that it provides the **methodology** to compute u_h. If we choose a basis $\{\psi_1, \psi_2, \ldots, \psi_N\}$ of

V_h and define $u_h \in V_h$ by

$$u_h(\mathbf{x}) = \sum_{j=1}^{n} \mathsf{U}_j\, \psi_j(\mathbf{x}) \tag{1.33}$$

replace Equation 1.33 and take $w_h = \psi_i$ in Equation 1.32, we obtain

$$a\left(\sum_{j=1}^{N} \mathsf{U}_j \psi_j, \psi_i\right) = \sum_{j=1}^{N} \underbrace{a(\psi_j, \psi_i)}_{A_{ij}} \mathsf{U}_j = \underbrace{\ell(\psi_i)}_{b_i}, \quad i = 1, \dots, N \tag{1.34}$$

Figure 1.5: Boris Galerkin (Russia, 1871-1945).

This clearly is a linear system of equations we can solve on a computer:

$$\mathbb{A}\, \mathbf{U} = \mathbf{b} \tag{1.35}$$

where

- $\mathbf{U} = [\mathsf{U}_1, \mathsf{U}_2, \dots, \mathsf{U}_N]^\intercal$ is the vector of unknown degrees of freedom (dofs);
- $\mathbb{A} \in \mathbb{R}^{N\times N}$ is the system matrix;
- $\mathbf{b} \in \mathbb{R}^N$ is the rhs vector,

which for our prototypical example of the heat diffusion problem, are defined by

$$A_{ij} = a(\psi_j, \psi_i) = \int_\Omega \left(\mu\, \nabla\psi_j\right) \cdot \nabla\psi_i \, dv, \tag{1.36}$$

and

$$b_i = \ell(\psi_i) = \int_\Omega f\, \psi_i \, dv. \tag{1.37}$$

The size of the **Algebraic** problem to be solve clearly depends on how big the dimension of the subspace V_h is, and for solving such system, *Direct* or *Iterative* methods may be used.

About matrix $\mathbb{A}$, in this particular case, we can say:

- Since $a(v, w) = a(w, v)\ \forall v, w \in V$ (i.e., $a(\cdot, \cdot)$ is symmetric), clearly $\mathbb{A}$ is also symmetric;
- if $a(w, w) > 0\ \forall w \in V$ (then, $(\cdot, \cdot)$ is said to be strongly coercive[8]), taking any arbitrary nonzero $w_h = \sum_{j=1}^{N} W_j\, \psi_j$ and defining $\mathbf{W} = [\mathsf{W}_1, \mathsf{W}_2, \dots, \mathsf{W}_N]^\intercal$, we get

$$a\left(\sum_{j=1}^{N} \mathsf{W}_j \psi_j, \sum_{i=1}^{N} \mathsf{W}_i \psi_i\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathsf{W}_i\, A_{ij}\, \mathsf{W}_j = \mathbf{W}^\intercal \mathbb{A} \mathbf{W} > 0, \quad \mathbf{W} \neq \mathbf{0} \tag{1.38}$$

  thus, $\mathbb{A}$ is positive definite and therefore **invertible**, so, we can safely solve the system and find the unique **U** that defines u_h.

8: Strong coercivity is also referred to as V-ellipticity. For the bilinear form at hand, it can be shown this property holds.

1.5 Some classical Finite Element spaces

The question that arises is how to construct a discrete space. The typical choice is to construct piecewise polynomial spaces associated to a **conforming partition** (a finite element mesh) of the computational domain, i.e.,

$$V_h = \{w \in C^0(\bar{\Omega}),\ w|_K \in P_k(K)\ \forall K \in \mathcal{T}_h\}$$

where $P_k(K)$ is the space of polynomials of degree k on K. For instance, for $k = 1$ in 2D we have the classical *hat* nodal functions on 3-node triangular elements illustrated in Figure 1.6.

$$P_1(K) = \{p : K \longrightarrow \mathbb{R},\ p = \alpha + \beta x + \gamma\, y\}$$

With these local functions, a global nodal base $\{\psi_1, \ldots, \psi_n\}$, can be constructed, such that there is one function for each vertex (node) of the mesh. Any piecewise linear function can be written as a linear combination of such functions. They are constructed such that the delta Kronecker property holds, i.e.,

$$\psi_i(\mathbf{x}_j) = \delta_{ij}$$

These are called *Lagrangian* elements. By looking at the basis functions illustrated in Figure 1.6, we notice that they have **compact support**, i.e, they are different than zero only in the cluster of elements shared by a given node, thus leading to a sparse matrix $\mathbb{A}$. Polynomials of higher order (quadratic, cubic, etc.) can be define similarly, just by introducing addition **degrees of freedom** (dofs).

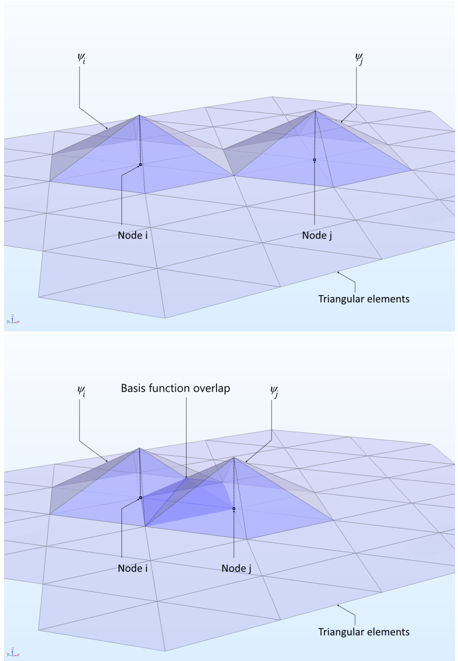


Figure 1.6: Examples of P_1 basis functions associated to the nodes of a 2D triangular mesh. Source: https://https://www.comsol.com/.

An important concept for the construction of finite element spaces as the ones just mentioned, is that of a finite element conforming mesh is, which we define here for completeness and ilustrate in Figure 1.7. These are the type of meshes we will consider in this course.

Definition 1.5.1 *A partition $\mathcal{T}_h$ of a domain Ω is* ***conforming*** *if $\bar{K}_i \cap \bar{K}_j$ is either*

- *empty, or,*
- *a vertex, or*
- *a complete edge.*
- *a complete face (in 3D)*

otherwise the partition is said to be ***nonconforming***.

An interesting site to take a look at the different finite elements that have been invented so far (or, possibly, most of them) is the library `DefElement`, a project related to the `FEniCS` project.

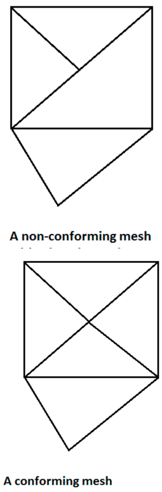
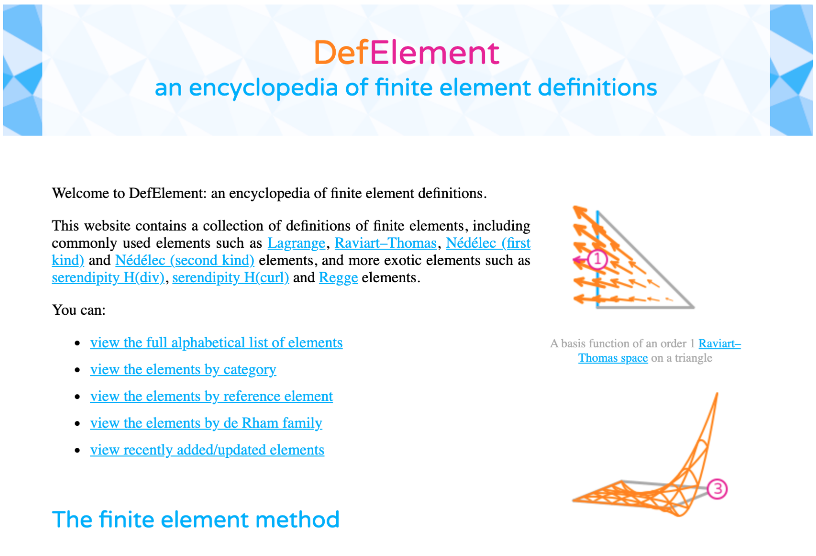


Figure 1.7: Conforming vs. nonconforming partitions of Ω.

We finally state the following theorem which is extremely important to the FEM and justify the choice of piecewise polynomial spaces to approximate the weak solution of PDEs:

Theorem 1.5.1 *Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain which can be partitioned into N_e Lipschitz subdomains K_j (i.e., $\overline{\Omega} = \bigcup_{j=1}^{N_e} \bar{K}_j$, $K_i \cap K_j =$*


DefElement
an encyclopedia of finite element definitions

Welcome to DefElement: an encyclopedia of finite element definitions.

This website contains a collection of definitions of finite elements, including commonly used elements such as Lagrange, Raviart–Thomas, Nédélec (first kind) and Nédélec (second kind) elements, and more exotic elements such as serendipity H(div), serendipity H(curl) and Regge elements.

You can:

- view the full alphabetical list of elements
- view the elements by category
- view the elements by reference element
- view the elements by de Rham family
- view recently added/updated elements

A basis function of an order 1 Raviart–Thomas space on a triangle

The finite element method

$\emptyset$ $\forall i \neq j$. *Let* v *be a* ***piecewise-polynomial*** *function on such a partition, then*

$$v \in H^1(\Omega) \iff v \in C^0(\overline{\Omega})$$

also

$$v \in H^2(\Omega) \iff v \in C^1(\overline{\Omega})$$

1.6 The `FEniCSx` platform

From the lectures, it will become quite clear that many of the computations that are executed when using `FEniCSx` library, such as, **construction of finite element spaces**, **assembly of finite element matrizes**, **numerical integration**, **imposition of boundary conditions**, **matrix solving**, and so on, remains transparent to the user. All these calculations being possible, thanks to several components that form the `FEniCSx` platform. These are[9]

9: The descriptions of the different components are just a brief summary taken from their corresponding repository at `https://github.com/FEniCS`.

1. `UFL`: `Python` library for writing problems in variational form. It provides the syntaxis to define linear and bilinear forms, finite elements spaces, such that the PDEs can be written in weak form in a language that is close to the mathematical one;
2. `DOLFINx`: Dynamic Object Oriented Library for Fini te Element Computation. It provides the computational environment of FEniCSx in C++ and Python and serves, among other things, to interface to parallel linear algebra routines, such as `Petsc`;
3. `FFCx`: FEniCS form compiler. From a high-level descr iption of the form in the Unified Form Language (UFL), it generates efficient low-level C code that can be used to assemble the corresponding discrete operators. Also, the tutorial about *Just-in-time-compilation* reveals interesting information (see `JIT`);
4. `Basix`: Is a finite element definition and tabulatio n runtime library. Basix allows users e.g. to evaluate finite element basis functions,

access geometric and topological information about reference cells, interpolate into a finite element space, among other things;

1.7 1st Tutorial: `FEniCSx` implementation of Poisson's problem

As already commented, we will adopt the `Colab notebook` environment to present the several tutorials in which the course has been divided. For the first tutorial, we will present the `FEniCSx` implementation of the prototypical elliptic second order problem, namely, the Poisson's equation, so, please, click in the link below in order to proceed

`|> Poisson's tutorial`

TWO CLASSICAL 2nd ORDER PROBLEMS 2

2.1 Transient heat conduction

|> Transient Poisson's problem

2.2 The elastostatic problem

|> Elastostatic problem

MIXED PROBLEMS 3

3.1 Thermoelastic problem

|> Thermoelasticity

3.2 The incompressible Navier-Stokes equations

|> Navier-Stokes problem

4 ADVANCED TOPICS

4.1 Darcy's flow in porous media

|> Darcy's problem

4.2 Acoustics: The Helmholtz's equation

|> Helmholtz's problem

HIGH PERFORMANCE COMPUTING WITH FEniCSx

5

5.1 Parallel computing in FEniCSx

|> HPC practice

Bibliography

Here are the references in citation order.

[1] A. Ern and J.L. Guermond. *Theory and Practice of Finite Elements*. Ed. by Springer. Vol. 159. 2004 (cited on page 2).

[2] P. Ciarlet. *Finite Element Methods (Part 1)*. Ed. by Elsevier by P. Ciarlet J.L. Lions. Vol. II. 1991 (cited on page 2).

[3] B.D. Reddy. *Introductory Functional Analysis*. Ed. by Springer. Vol. 27. 1998 (cited on page 2).

[4] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods, 3rt ed.* Ed. by Springer. Vol. 15. 2008 (cited on page 2).

Alphabetical Index