People's Democratic Republic of Algeria
Ministry of Higher Education and
Scientific Research
Higher Normal School of Technological
Education-Skikda

Department of Mathematics

Dissertation

Presented in partial fulfillment of the requirements for the degree in Mathematics,
specializing in Secondary Education

Entitled

Aposteriori Error Estimates, Theoretical and Numerical Study

Presented by:
Arif Raouf

Board of Examiners:

President:	Dr. Kidouchi Wahida	MCA	ENSET SKIKDA
Supervisor:	Dr. Gouassmia Mohamed	MCA	ENSET SKIKDA
Examiner:	Dr. Ghomrani Sarra	MCA	ENSET SKIKDA
Examiner:	Dr. Kenef Esma	MCA	ENSET SKIKDA

2024- 2025

Acknowledgement

First and foremost, I would like to express my deepest gratitude to God for giving me the strength, wisdom, and perseverance to complete this academic journey.

I am profoundly grateful to my advisor **Mr . Mohammed Gouasmia** , for his invaluable guidance, patience, and constant encouragement throughout my research. His expertise and constructive feedback were instrumental in shaping this work.

My sincere thanks to the ENSET SKIKDA'S Teachers for their unwavering support, insightful lectures, and the knowledge they imparted, which laid the foundation for my academic growth.

I extend my heartfelt appreciation to my family especially my parents, sisters , for their endless support, sacrifices, and motivation during

challenging times. Their belief in me kept me going.

I also owe a debt of gratitude to my friends and classmates for their companionship, stimulating discussions, and moral support, which made this journey more enjoyable and less daunting.

Thank you all for being part of this significant milestone in my life.

[RAOUF]

Dedication

To My Beloved Parents,

Your unwavering love, endless sacrifices, and steadfast support have been my foundation. You believed in me even when I doubted myself, and your encouragement carried me through every challenge. This achievement is as much yours as it is mine. Thank you for shaping my dreams and standing by me every step of the way.

To My Amazing Friends,

You have been my pillars of strength, my source of laughter, and my refuge in moments of doubt. The memories we’ve shared, the late-night study sessions, and the words of motivation kept me going. I am forever grateful for your friendship and the joy you bring into my life.

To My Respected Colleagues,

Your collaboration, wisdom, and camaraderie made this journey enriching and inspiring. Each discussion, challenge, and shared success contributed to my growth. Thank you for being incredible teammates and mentors.

This milestone is a testament to the love, support, and inspiration I’ve received from all of you. With deepest gratitude, I dedicate this achievement to you.

[Raouf]

Contents

Introduction

Today, a posteriori error estimates are well developed for a large class of simple linear elliptic model problems. The crucial question is whether these procedures are also effective in practical solutions such as in the linear anal- ysis of geometrically complex 2D, 3D and shell problems, and in the analysis of problems including nonlinear ef- fects, time-dependent loads or multi-physics phenomena. From a practical point of view, there is much interest in reliable and efficient methods to estimate the error in complex analyses. In this context we shall consider reli- ability to mean that the error estimates can be expected to be accurate, and efficiency to mean that the computer time to obtain these estimates is small when compared to the total processing time used. In reviewing the state of a posteriori error estimators, we will conclude that efficient error estimates still need to be used with care because they are generally not based on guaranteed error bounds. In fact, nearly-guaranteed error bounds are still quite expensive to evaluate for complex problems and are fre- quently not yet available . this memory is classified into 4 chapters , in the first one we will see some classes of sobolev spaces , then we got in the second one the most importants ideas about the finite elements methode and some importants theorems and definitions, which we used in studying the laplacien model case in the third chapter, thena FreeFem++ Application in the fourth chapter .

Chapter 1

Generalities About Sobolev Spaces

1.1 Important Definitions

Support of a function

Definition 1.1.1. *Let f be a function defined from a domain Ω to either $\mathbb{R}$ or $\mathbb{C}$. The support of f, denoted as* $\operatorname{supp}(f)$*, is the closure of the set of points where f is non-zero. In other words, it's the set of all points where f does not equal zero, along with any limit points of that set.*

$$\operatorname{supp}(f) = \overline{\{x \in \Omega : \quad f(x) \neq 0\}} \tag{1.1}$$

The Space $C^n(\Omega)$

Definition 1.1.2. *Let n be a nonzero natural number, and let f be a function defined on Ω with values in $\mathbb{R}$. We say that f is a function of class C^n on Ω (or f is n-times differentiable) if and only if all partial derivatives up to order n exist and are continuous on Ω. We denote by $C^n(\Omega)$ the vector space of functions of class C^n on Ω.*

The Space $C^\infty(\Omega)$

Definition 1.1.3. *Let f be a function defined on Ω with values in $\mathbb{R}$. We say that f is a function of class C^∞ on Ω if and only if it is of class C^n on Ω for every natural number n (i.e., if f is infinitely differentiable on Ω). We denote by $C^\infty(\Omega)$ the vector space of functions of class C^∞ on Ω.*

The Space $C_0^k(\Omega)$

Definition 1.1.4. *For $k \in \mathbb{N} \cup \{0, \infty\}$, we define the space of k-times continuously differentiable functions with compact support:*

$$C_0^k(\Omega) := \{v \in C^k(\Omega) \mid \operatorname{supp(v)} \textit{ is compact in } \Omega\}$$

When $k = \infty$, this denotes smooth (C^∞) functions with compact support.

The Space $C^k(\overline{\Omega})$

Definition 1.1.5. *Let $\Omega \subset \mathbb{R}^N$ be a bounded domain. For any $k \in \mathbb{N} \cup \{0\}$, we denote by $C^k(\overline{\Omega})$ the linear space of all functions in $C^k(\Omega)$ whose partial derivatives up to order k admit continuous extensions to the closure $\overline{\Omega}$.*

Remark 1.1.1. 1 *When $k = 0$, we write simply $C(\overline{\Omega})$ instead of $C^0(\overline{\Omega})$.*

2 *These spaces are Banach spaces when equipped with the norms:*

$$\|y\|_{C(\overline{\Omega})} := \max_{x \in \overline{\Omega}} |y(x)|$$

and for $k \in \mathbb{N}$:

$$\|y\|_{C^k(\overline{\Omega})} := \sum_{|\alpha| \leq k} \|D^\alpha y\|_{C(\overline{\Omega})}$$

where $\alpha = (\alpha_1, \ldots, \alpha_N)$ is a multi-index with $|\alpha| = \alpha_1 + \cdots + \alpha_N$.

The Space $D(\Omega)$

Definition 1.1.6. *Let f be a function defined on Ω with values in $\mathbb{R}$. We say that f is a test function if and only if it is infinitely differentiable and has compact support. We denote by $D(\Omega)$ the vector space of test functions.*

$$D(\Omega) = \{f \in C^\infty(\Omega); \, Supp(f)\mathrm{Compact}\Omega\}$$

1.2 Sobolev Spaces

We will start our study by introducing a very useful tool in solving Partial Diffirential Equations , we will recall basic notions from the theory of L^p spaces and Sobolev spaces, which are indispensable prerequisites for the next chapters. In the following, $\Omega \in \mathbb{R}^n$ denotes a nonempty, bounded, and Lebesgue measurable set having the N-dimensional Lebesgue measure |E| .

1.3 L^p Spaces

Definition 1.3.1. *We denote by $L^p(\Omega), 1 \leq p < \infty$, the linear space of all (equivalence classes of) Lebesgue measurable functions y that satisfy :*

$$\int_\Omega |y(x)|^p dx < \infty$$

In this connection, functions that differ only on a set of zero measure are identified with each other and considered to belong to the same equivalence class. Endowed with the norm :

$$\|y\|_{L^p(\Omega)} = \left(\int_\Omega |y(x)|^p dx \right)^{\frac{1}{p}} \tag{1.2}$$

$L^p(E)$, with $1 \leq p < \infty$, becomes a Banach space .

1.3.1 The Space $L^\infty(E)$

Let $E \subset \mathbb{R}^n$ be a measurable set. We denote by $L^\infty(E)$ the Banach space of all (equivalence classes of) Lebesgue measurable, essentially bounded functions $y : E \to \mathbb{R}$, equipped with the norm:

$$\|y\|_{L^\infty(E)} := \text{esssup}_{x\in E}|y(x)| = \inf_{\substack{F\subset E\\ |F|=0}} \left(\sup_{x\in E\setminus F} |y(x)| \right) . \tag{1.3}$$

By ess sup we mean the essential maximum or supremum of a function. This excludes any maxima that change upon the removal of single points that are isolated in a certain sens and thus not essential. For instance, the function $y : [0, 1] \to \mathbb{R}$ which attains the values zero on $(0, 1]$ and one at $x = 0$ has maximum 1 but essential supremum 0.

Weak Derivatives

Definition 1.3.2. *Let $y \in L^1_{loc}(\Omega)$, and some multi index α be given , if a function $w \in L^1_{loc}(\Omega)$ satisfies :*

$$\int_\Omega y(x) D^\alpha v(x) dx = (-1)^{|\alpha|} \int_\Omega w(x) y(x) dx$$

, $\forall v \in C^k_0(\omega) Then w is called the weak derivative of y associated with \alpha$.

1.3.2 The Spaces $W^{k,p}$

Definition 1.3.3. *Let* $1 \leq p \leq \infty$ *and* $K \in \mathbb{N}$ *, we denote by* $W^{k,p}(\Omega)$ *the linear space of all functions* $y \in L^p(\Omega)$ *having weak derivatives* $D^\alpha y$ *in* $L^p(\Omega)$ *for all multi indices* α *of length*

$|\alpha| \leq k$ endowed with the norm :

$$||y||_{w^{k,p}}(\Omega) = \left(\sum_{|\alpha| \leq k} \int_\Omega |D^\alpha y(x)|^p dx \right)^{\dfrac{1}{p}}$$

The spaces $W^{k,p}(\Omega)$ are **Banach spaces** , they are reffered to as *Sobolev Spaces* .

Remark 1.3.1. *For the particularly interesting case of* $p = 2$*, we write :*

$$H^k(\Omega) = W^{k,2}(\Omega)$$

1.4 The Space H^1

Definition 1.4.1. *The Space* H^1 *is defined by :*

$$H^1(\Omega) = \{y \in L^2(\Omega) : \tfrac{\partial y}{\partial x_i} \in L^2(\Omega), i = 1......N\}$$

Where $\dfrac{\partial y}{\partial x_i}$ *is the weak derivative of* v*.*

And the norm is given by :

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

where $(|\nabla y|)^2 = (D_1 y)^2 + ... + (D_n y)^2$. with the scalar product :

$$(u, v)_{H^1} = \int_\Omega (u.v)dx + \int_\Omega (\nabla u.\nabla v)dx$$

H^1 becomes a **Hilbert space** .

Weak Gradient

Definition 1.4.2. *Let a function* $v : \Omega \to \mathbb{R}$ *be given. We say that* v *admits a weak gradient if* v *admits the weak i-th partial derivative for all* $1 \leq i \leq d$*. We set*

$$\nabla v := (\partial x_1 v, \ldots ....\partial x_d v)^t$$

Definition 1.4.3. *The space* $H^1(\Omega)$ *is the space of all the functions which admit a weak gradient.*

1.4.1 Closure of A Space

Definition 1.4.4. *Let E be a subset of a normed space $(X, \|\cdot\|)$. The closure of E, denoted by $\overline{E}$ or E^-, is defined as:*
$\overline{E} = \{x \in X :$ *there exists a sequence* $\{x_n\}_{n=1}^{\infty} \subset E$ *such that* $x_n \to x\}$
or equivalently,

$$\overline{E} = \left\{x \in X : \exists \{x_n\}_{n=1}^{\infty} \subset E \text{ with } \lim_{n\to\infty} x_n = x\right\}$$

- We say that a set $E \subset X$ is dense in X if its closure is equal to X, that is, $\overline{E} = X$.

1.5 The Space H_0^1

The closure of $C_0^\infty(\Omega)$ in $W^{k,p}(\Omega)$ is denoted by $W_0^{k,p}$, We put

$$H_0^k(\Omega) = W_0^{k,2}(\Omega) \ .$$

The space $H_0^1(\Omega)$ is the closure of $C_0^\infty(\Omega)$ in $H^1(\Omega)$.

Definition 1.5.1. *Let Ω be a regular bounded open set of class C^1. The space $H_0^1(\Omega)$ coincides with the subspace of $H^1(\Omega)$ consisting of functions that vanish on the boundary $\partial\Omega$.*

The Space H^m

Definition 1.5.2. *For an integer $m \geq 0$, the Sobolev space $H^m(\Omega)$ is defined by*

$$H^m(\Omega) = \left\{v \in L^2(\Omega) \text{ such that, } \forall \alpha \text{ with } |\alpha| \leq m, \partial^\alpha v \in L^2(\Omega)\right\}, \tag{1.4}$$

here the partial derivative $\partial^\alpha v$ is to be taken in the weak sense.

Proposition 1.5.1. *Equipped with the scalar product*

$$\langle u, v\rangle = \int_\Omega \sum_{|\alpha|\leq m} \partial^\alpha u(x)\partial^\alpha v(x)dx \tag{1.5}$$

and the norm $\|u\|_{H^m(\Omega)} = \sqrt{\langle u, u\rangle}$, the Sobolev space $H^m(\Omega)$ is a Hilbert space.

1.6 The Space $H(div, \Omega)$

1.6.1 Weak divergence

Definition 1.6.1. *Let a function $v : \Omega \to \mathbb{R}^d$ be given. We say that v admits a weak divergence if :*

1. $v \in [L^2(\Omega)]^d$
2. *There exists a function $w : \Omega \to \mathbb{R}$ such that :*
   - $w \in L^2(\Omega)$;
   - $(v, \nabla\phi) = (w, \phi), \forall\phi \in D(\Omega)$.

Definition 1.6.2. *The Space H(div) is defined by :*

$$H(div) = \left\{\sigma \in L^2(\Omega)^N, div\sigma \in L^2(\Omega)\right\}$$

Where $div\sigma$ is the weak divergence of σ

1.7 The Dual Space

Definition 1.7.1. *Let V be a real Hilbert space. Its dual V' is the set of continuous linear forms on V. By definition, the norm of an element $L \in V'$ is :*

$$||L||_{V'} = \sup_{x \in V, x \neq 0} \frac{||L(x)||}{||x||}.$$

Remark 1.7.1.

1. *The Dual Of $L^2(\Omega)$ is identified at $L^2(\Omega)$ itself .*
2. *We can define the dual of Sobolev spaces , for example , the dual of the space $H_0^1(\Omega)$ is very important in the next study .*

1.7.1 The Space $\mathbf{H}^{-1}$

Definition 1.7.2. *The Dual Space of $H_0^1(\Omega)$ is called H^{-1} , it can be defined by :*

$$H^{-1}(\Omega) = \left\{ f = v_0 + \sum_{i=1}^{N} \frac{\partial v_i}{\partial x_i} \text{ avec } v_0, v_1, \ldots, v_N \in L^2(\Omega) \right\}.$$

Also , every continued lineair form on $H_0^1(\Omega)$, named $L \in H^{-1}$ is :

$$L(\phi) = \int_\Omega \left(v_0\phi - \sum_{i=1}^{N} v_i \frac{\partial \phi}{\partial x_i} \right) dx$$

- with $v_0, v_1, \ldots, v_N \in L^2(\Omega)$, and $\phi \in H_0^1(\Omega)$.

1.8 Some Important Theorems and Inequalities

Dealing with partial diffirential equations and solving elliptiques problems requires many theorems and inequalities which we use to get the existence of the approximate solution or to prove th Lax Milgram Theorem .

1.8.1 Young's inequality

Let p , q $\in]1,+\infty[$ such that $\frac{1}{p} + \frac{1}{q} = 1$ then :

$$\forall a, b \in \mathbb{R}_+, ab \leq \frac{a^p}{p} + \frac{b^q}{q}.$$

1.8.2 Cauchy-Schwartz inequality

Let f,g $\in \mathrm{L}^2(\Omega)$, then we have :

$$||fg||_{L^1(\Omega)} \leq ||f||_{L^2(\Omega)} ||g||_{L^2(\Omega)}.$$

1.8.3 Holder's inequality

Let (Ω, τ, μ) be a measurable space , and p,q $\in]1,+\infty[$ Such that : $\frac{1}{p} + \frac{1}{q} = 1$ then :

$$\forall f \in L^p(\Omega), \forall g \in L^p(\Omega); fg \in L^1(\Omega).$$

Also That :

$$||fg||_{L^1(\Omega)} \leq ||f||_{L^p(\Omega)} ||g||_{L^q(\Omega)}.$$

1.8.4 Poincaré's inequality

Let $\Omega \subset \mathbb{R}^{\mathbb{N}}$ Bounded at least at one direction of the space , It exist a constant $C \geq 0$ Such that :

$$\forall \text{v} \in H_0^1 \ , \qquad \int_\Omega |v(x)|^2 dx \leq C \int_\Omega |\nabla v(x)|^2 dx \ .$$

1.8.5 Orthogonal

Definition 1.8.1. *Let (H,(.,.)) be a Hilbert Space on* $\mathbb{K}$ *, we say that u,v from H are orthogonal if and only if :*

$$(u, v) = 0$$

1.8.6 Perpendicular of a set

Definition 1.8.2. *Let ((.,.),H) be a Hilbert space on the field* $\mathbb{K}$*, and let F be a part of H. We call the perpendicular F the set* $F^\perp$ *Which defined by :*

$$F^\perp = \{x \in H; \forall y \in F, (x, y) = 0\}$$

Remark 1.8.1. $F^\perp$ *is a Subspace of* H*.*

1.8.7 Perpendicular projection onto a convex set :

Theorem 1.8.1. *Let H be a Hilbert Space On* $\mathbb{K}$ *,* Ω *is a non-empty Closed Convex On H , let* $x \in H$ *:*

$$\exists ! P_\Omega(x) \in \Omega; \quad \|x - P_\Omega(x)\| = \inf_{y \in \Omega} \|x - y\|.$$

Remark 1.8.2. P_Ω *is an injective Mapping.*

Definition 1.8.3. *Let H be a Hilbert space over the field* $\mathbb{K}$*, and F be a closed linear subspace of H. Let u be an element of F. The application PF is called an orthogonal projection onto F. We call the image of u by PF the orthogonal projection of u onto F.*

Definition 1.8.4. *Let H be a Hilbert space over the field K and let F bea closed linear subspace of H . then the orthogonal complement* ^{p}erp *satisfies :*

$$H = F \oplus F^\perp.$$

1.8.8 Riesz Representation Theorem :

Theorem 1.8.2. *Let H be a real Hilbert space, and let H' be its dual. For every continuous linear form $f \in V'$, there exists a unique $y \in V$ such that :*

$$\exists ! v \in H, \forall u \in H, f(u) = (u, v)$$

Proof. By taking $F = ker(f)$, f is linear and continuous form , so F is a closed subspace from H . By The Definition (**??**) : H=F $\oplus F^{\perp}$.
There are two possibilities :
if $f = 0$ so $F = H$, then it suffices to take $v = 0$.
if $f \neq 0$, then $F^{\perp} \neq H$ so $F^{\perp} \neq \{0\}$ this means that

$$\exists v0 \in F^{\perp}, v0 \neq 0.$$
$$f(v_0) \neq 0\text{:}\ v_0 \notin F.$$

Since u belongs to H, we have:

$$u = u - \frac{f(u)}{f(v_0)} v_0 + \tfrac{f(u)}{f(v_0)} v_0.$$

On the other hand:

$$f\left(u - \frac{f(u)}{f(v_0)} v_0\right) = f(u) - \tfrac{f(u)}{f(v_0)} f(v_0) = 0.$$

This means:

$$u - \frac{f(u)}{f(v_0)} v_0 \in F.$$

Therefore,

$$\left(u - \tfrac{f(u)}{f(v_0)} v_0, v_0\right) = 0,$$

and thus:

$$(u, v_0) = \tfrac{f(u)}{f(v_0)} (v_0, v_0).$$

Two cases arise:

- If $K = \mathbb{R}$, we set:
  $v = \dfrac{f(u)}{(v_0, v_0)} v_0.$
- If $K = \mathbb{C}$, we set:
  $v = \dfrac{\overline{f(u)}}{(v_0, v_0)} v_0$

  Consequently,
  $\forall u \in H, \quad f(u) = (u, v).$

■

Chapter 2

Finite Elements Method

Introduction

In This Chapter, we will see a very powerful and useful tools to solve a wide range of mathematical problems, which very hard or impossible to solve, it is numerical method called the finite elements, this method is strongly dependent on the **variational form** of the problem .
Historical overview: The very first use of this method was in 1940 by the mathematician Richard Courant without this denomination. But it was developed by the mechanicals, whose gave it her popular name and show it is efficacy in the $1950-1960$. After that, the mathematician well-considered this method and developed its foundations theoretical.

2.1 Important Definitions

2.1.1 Test functions

Definition 2.1.1. *Let Ω be a bounded, regular open set of $\mathbb{R}^n$, let f be a function defined from Ω to $\mathbb{R}$, we say that f is a test function if it is infinitely differentiable and compactly supported.*

Coercivity of a bilinear form

Definition 2.1.2. *Let a be a bilinear form on a Hilbert space H. So we said that a is coercive over H if there exists a constant $C \geq 0$ such that*

$$a(u,u) \geq C \|u\|_H^2 \quad \text{for all } u \in H \tag{2.1}$$

Continuity of a bilinear form

Definition 2.1.3. *Let a be a bilinear form on a Hilbert space H. Then we said that a is continuous over H if there exists a constant $M \geq 0$ verifies the following inequality*

$$|a(u,v)| \leq M\|u\|_H\|v\|_H \quad \textit{for all } u\,,\, v \in H \tag{2.2}$$

2.1.2 Green's formula

Theorem 2.1.1. *Let Ω be a regular open of class C^1. Let f be a function of class $C^1(\Omega)$ with bounded support in the closed $\overline{\Omega}$. Then it verifies the following Green's formula*

$$\int_\Omega \frac{\partial w}{\partial x_i}(x)dx = \int_\Omega w(x)n_i(x)ds \tag{2.3}$$

Where n_i is the i^{th} component of the unit external normal of Ω.

Remark 2.1.1. *We called that a regular function w has its support bounded in the closed set $\overline{\Omega}$ means that it vanishes at infinity if the closed set is not bounded.*

Integration by Parts Formula

Definition 2.1.4. *Let Ω be a regular open set of class C^1. Let u and v be two functions of $C^1(\Omega)$ with bounded support in the closed set $\overline{\Omega}$. Then they satisfy the integration by parts formula:*

$$\int_\Omega u(x)\frac{\partial v}{\partial x_i}(x)dx = -\int_\Omega v(x)\frac{\partial u}{\partial x_i}(x)dx + \int_{\partial\Omega} u(x)v(x)n_i(x)ds.$$

2.2 Variational form

Definition 2.2.1. *let a be a bilinear form , L is linear , we call a variational form on a Hilbert Space H the following problem: Find $u \in H$ such that*

$$a(u,v) = L(v), \quad \forall v \in H \tag{2.4}$$

The principle idea of the **Variational Form** is helping us to proving the existence and uniqueness of the solution to problem (**??**) , using **Lax-Milgram** theorem

2.3 Lax-Milgram theorem

Theorem 2.3.1. *Let V be a real Hilbert space, L a continuous linear form on V, a a continuous coercive bilinear form on V. Then the variational formulation (**??**) admits a unique solution. Moreover, this solution depends continuously on the linear form L.*

Proof. For all $w \in V$, the mapping $v \mapsto a(w,v)$ is a continuous linear form on V. Therefore, the Riesz representation theorem implies that there exists an element of V, denoted $A(w)$, such that

$$a(w,v) = (A(w),v), \quad \text{for all } v \in V$$

Furthermore, the bilinearity of $a(u,v)$ obviously implies the linearity of the mapping $w \mapsto A(w)$. Now, taking $v = A(w)$, the continuity (**??**) of $a(u\,,\,v)$ shows that

$$c\|A(w)\|^2 = a(w,A(w)) \leq M\|w\|\|A(w)\|$$

i.e., $\|A(w)\| \leq M\|w\|$ and thus $w \mapsto A(w)$ is continuous. Another application of the Riesz representation theorem implies that there exists an element of V, denoted f, such that

$$L(v) = (f,v), \quad \text{for all } v \in V \tag{2.5}$$

Finally, the variational problem (**??**) equivalent to

$$\text{find } u \in V, \qquad \text{such that } A(u) = f \tag{2.6}$$

To prove the theorem, we need to show that the operator A is bijective from V to V (which implies the existence and uniqueness of u) and that its inverse is continuous (which proves the continuous dependence of u with respect to L).

The coercivity of $a(u,v)$ shows that

$$\nu\|w\|^2 \leq a(w,w) = (A(w),w) \leq \|A(w)\|\|w\| \tag{2.7}$$

which gives $\nu\|w\| \leq \|A(w)\|$ for all $w \in V$. i.e., A is injective. To show that A is surjective, i.e., $\operatorname{Im}(A) = V$ (which is not obvious if V is infinite-dimensional), it sufficient to show that $\operatorname{Im}(A)$ is closed in V and that $\operatorname{Im}(A)^\perp = \{0\}$. Indeed, in this case, we have

$$V = \{0\}^\perp = (\operatorname{Im}(A)^\perp)^\perp = \operatorname{Im}(A)$$

which proves that A is surjective.

Let $A(w_n)$ be a sequence in $\operatorname{Im}(A)$ that converges to b in V we obtain

$$\nu\|w_n - w_p\| \leq \|A(w_n) - A(w_p)\|$$

which tends to zero as n and p tend to infinity. Thus, w_n is a Cauchy sequence in the Hilbert space V, i.e., it converges to a limit $w \in V$. Then, by the continuity of A, we deduce that $A(w_n)$ converges to $A(w) = b$, i.e., $b \in \operatorname{Im}(A)$ and $\operatorname{Im}(A)$ is closed. On the other hand, let $v \in \operatorname{Im}(A)^\perp$; the coercivity condition of $a(u,v)$ implies that

$$\nu\|v\|^2 \leq a(v,v) = (A(v),v) = 0$$

i.e., $v = 0$ and $\operatorname{Im}(A)^\perp = \{0\}$, which proves that A is bijective. Let A^{-1} be its inverse: inequality with $w = A^{-1}(v)$ proves that A^{-1} is continuous, thus u depends continuously on f. ■

2.4 Elliptic Problems

Let Ω be a bounded and regular open subset of $\mathbb{R}^n$, we introduce the following probelm :

$$\begin{cases} -\Delta u = f & , in \quad \Omega \\ u = 0 & , in \quad \delta\Omega \end{cases} \tag{2.1}$$

Definition 2.4.1. *We Call A Classical (Strong) solution for (??) every function $u \in C^2(\Omega) \cap C(\overline{\Omega})$ Which Satisfy (??).*

Proposition 2.4.1. *Let be $u \in C^2(\Omega)$ and a space H defined by :*

$$H = \{\phi \in C^1(\Omega); \phi = 0 \quad \boldsymbol{in}\partial\Omega\}$$

u is a classical solution for (??) if and only if $u \in H$ and satisfy :

$$\int_\Omega \nabla u(x)\nabla v(x)dx = \int_\Omega f(x)v(x)dx, \forall v \in H \tag{2.2}$$

Assume that u is a solution to problem (??). Multiply both sides of (??) by a test function $v \in H$ and integrate over Ω to obtain:

$$\int_\Omega -\Delta u(x)v(x)\,dx = \int_\Omega f(x)v(x)\,dx, \quad \forall v \in H.$$

Using integration by parts (Green's formula), we have:

$$\int_\Omega \Delta u(x)v(x)\,dx = -\int_\Omega \nabla u(x)\cdot\nabla v(x)\,dx + \int_{\partial\Omega} \frac{\partial u}{\partial \eta}(x)v(x)\,ds.$$

Since $v \in H$ vanishes on the boundary, we obtain the weak formulation:

$$\int_\Omega \nabla u(x)\cdot\nabla v(x)\,dx = \int_\Omega f(x)v(x)\,dx, \quad \forall v \in H.$$

Conversely:

If $u \in H$ satisfies the weak formulation:

$$\int_\Omega \nabla u(x)\cdot\nabla v(x)\,dx = \int_\Omega f(x)v(x)\,dx, \quad \forall v \in H,$$

we can apply integration by parts in reverse to obtain:

$$\int_\Omega (\Delta u(x) + f(x))\,v(x)\,dx = 0, \quad \forall v \in H.$$

Since $\Delta u + f$ is a continuous function on Ω, it follow that:

$$\Delta u(x) + f(x) = 0, \quad \forall x \in \Omega.$$

That is, u satisfies the original PDE:

$$-\Delta u = f \quad \text{in } \Omega.$$

Moreover, since $u \in H$, it satisfies the boundary condition:

$$u = 0 \quad \text{on } \partial\Omega.$$

Thus, u is indeed a solution to problem (??).

2.4.1 Variational Formulation

The formulation (**??**) in Proposition (**??**) is called the **variational formulation** of the boundary value problem (refch222). The variational formulation (**??**) can be written as:

$$\begin{cases} u \in H; \\ a(u,v) = L(v), \quad \forall v \in H, \end{cases}$$

Where:

$$a(u,v) = \int_\Omega \nabla u(x) \cdot \nabla v(x)\,dx; \quad L(v) = \int_\Omega f(x)v(x)\,dx.$$

In other words: Find $u \in H$ such that $a(u,v) = L(v)$ for all $v \in H$.

2.4.2 Homogeneous Dirichlet Boundary Value Problem

Let Ω be a bounded and regular open subset of $\mathbb{R}^n$. Consider the following problem:

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{2.3}$$

where $f \in L^2(\Omega)$. This problem is called the **homogeneous Dirichlet problem**. The variational approximation of this problem involves three fundamental steps, presented in order below.

1 Deriving the Variational Form

To obtain the variational form of problem (**??**), multiply both sides by a test function $v \in D(\Omega)$ and integrate over Ω:

$$-\int_\Omega \Delta u(x)v(x)\,dx = \int_\Omega f(x)v(x)\,dx.$$

Using integration by parts (Green's formula), we get:

$$\int_\Omega \nabla u(x) \cdot \nabla v(x)\,dx - \int_{\partial\Omega} \frac{\partial u}{\partial \eta}(x)v(x)\,ds = \int_\Omega f(x)v(x)\,dx.$$

2.4.3 Variational Formulation of the Dirichlet Problem

Since the solution satisfies the Dirichlet boundary condition $u = 0$ on the boundary $\partial\Omega$, we choose the Hilbert space H such that:

$$\forall v \in H,\ v = 0 \text{ on } \partial\Omega,$$

which means:

$$v \in \ker(\gamma_0) = H_0^1(\Omega).$$

Consequently, the weak formulation becomes:

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

Thus, the variational formulation of the homogeneous Dirichlet boundary value problem is given by:

$$\begin{cases} \text{Find} & u \in H_0^1(\Omega); \\ & \int_\Omega \nabla u(x) \cdot \nabla v(x)\, dx = \int_\Omega f(x)v(x)\, dx, \quad \forall v \in H_0^1(\Omega). \end{cases} \tag{2.4}$$

2 Existence and Uniqueness of the Weak Solution

In this step, we verify that the variational formulation (**??**) admits a unique solution. For this purpose, we apply the Lax-Milgram theorem.
Let $H = H_0^1(\Omega)$ be the Hilbert space. The bilinear form a is defined by:

$$\forall u, v \in H, \quad a(u,v) = \int_\Omega \nabla u(x) \cdot \nabla v(x)\, dx,$$

and the linear form L is defined by: $\forall v \in H, \quad L(v) = \int_\Omega f(x)v(x)\, dx.$

Continuity of the Bilinear Form

For any $u, v \in H$, by the Cauchy-Schwarz inequality, we have:

$$\begin{aligned} |a(u,v)| = \left|\int_\Omega \nabla u(x) \cdot \nabla v(x)\, dx\right| &\leq \|\nabla u\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} \\ &= \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} \\ &= \|u\|_{H_0^1(\Omega)} \|v\|_{H_0^1(\Omega)}. \end{aligned}$$

Thus, the bilinear form a is continuous on H **Continuity and Coercivity Analysis**

For any $v \in H$, we have:

$$|L(v)| = \left|\int_\Omega f(x)v(x)\, dx\right| \leq \|f\|_{L^2(\Omega)} \cdot \|v\|_{L^2(\Omega)} \leq \|f\|_{L^2(\Omega)} \cdot \|v\|_{H_0^1(\Omega)}.$$

Thus, L is continuous on H.

Coercivity: For any $v \in H$, by Poincaré's inequality, there exists $C > 0$ such that:

$$a(v,v) = \int_\Omega |\nabla v(x)|^2\, dx \geq C\|v\|_{H_0^1(\Omega)}^2.$$

Therefore, the bilinear form a is coercive on H.

Application of Lax-Milgram Theorem

The above results show that all conditions of the Lax-Milgram theorem are satisfied. Hence, the variational formulation (**??**) admits a unique solution $u \in H$. Moreover, since a is symmetric, this solution minimizes the energy functional:

$$J(u) = \min_{v\in H} J(v), \quad \text{where} \quad \forall v \in H,\ J(v) = \tfrac{1}{2}a(v,v) - L(v).$$

3 Equivalence Between Weak and Strong Forms

This step establishes that the solution of the weak formulation also solves the classical problem (**??**), i.e., we recover the classical solution from the weak formulation.

Let $u \in H_0^1(\Omega)$. Using integration by parts (Green's formula), we have:

$$\int_\Omega \nabla u(x)\cdot\nabla v(x)\,dx = -\int_\Omega \Delta u(x)v(x)\,dx + \int_{\partial\Omega} \frac{\partial u}{\partial\eta}(x)v(x)\,ds.$$

Since v vanishes on the boundary $\partial\Omega$, we obtain:

$$\int_\Omega f(x)v(x)\,dx = -\int_\Omega \Delta u(x)v(x)\,dx,$$

which implies:

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

Given that $\Delta u + f \in L^2(\Omega)$, , we conclude:

$$-\Delta u = f \quad \text{almost everywhere in } \Omega.$$

Trace Theorem and Variational Solution

Furthermore, by the trace theorem, since $u \in H_0^1(\Omega)$, we have:

$$u = 0 \quad \text{a.e. on } \partial\Omega.$$

Thus, we have fully characterized the boundary value problem (**??**).

Definition 2.4.2. *The solution $u \in H_0^1(\Omega)$ of the variational formulation (**??**) is called the* ***variational solution*** *of the boundary value problem (**??**).*

2.4.4 Non-Homogeneous Dirichlet Boundary Value Problem

Let $a, b \in \mathbb{R}$. Consider the following problem:

$$\begin{cases} -u'' + u = f, & \text{in } (0,1), \\ u(0) = a,\ u(1) = b. \end{cases} \tag{2.5}$$

These are called **non-homogeneous Dirichlet boundary conditions**, where a and b are not necessarily zero. For this problem, we seek a solution in $H^1(\Omega)$ rather than $H_0^1(\Omega)$.

Solution Strategy via Lifting Method

To apply the Lax-Milgram theorem directly, particularly to prove the coercivity of the bilinear form on $H^1(\Omega)$, we employ a lifting technique. Let:

$$u = u_0 + \overline{u}, \quad \text{where} \quad u_0(x) = a + (b-a)x.$$

(2.6)

Note that $u_0(0) = a$ and $u_0(1) = b$. The function $\overline{u}$ must then satisfy:

$$\begin{cases} -\overline{u}'' + \overline{u} = f - u_0, & \text{in } (0,1), \\ \overline{u}(0) = 0, \; \overline{u}(1) = 0. \end{cases} \tag{2.7}$$

This transforms the problem into one with homogeneous Dirichlet boundary conditions. The variational formulation is: Find $\overline{u} \in H_0^1(0,1)$ such that

$$\textstyle\int_0^1 \overline{u}'(x)v'(x)\,dx = \int_0^1 (f(x) - u_0(x))v(x)\,dx, \quad \forall v \in H_0^1([0,1]).$$

Since this problem admits a unique solution $\overline{u} \in H_0^1(\Omega)$, we conclude that the original boundary value problem (**??**) has a unique solution $u \in H^1(\Omega)$.

2.4.5 Neumann Problem

Variational Formulation

Let Ω be an open subset of $\mathbb{R}^n$, and consider the following boundary value problem:

$$\begin{cases} -\Delta u + cu = f, & \text{in } \Omega, \\ \dfrac{\partial u}{\partial \eta} = g, & \text{on } \partial\Omega, \end{cases} \tag{2.8}$$

Where:

- $x \in \Omega$ is the spatial variable,
- $c(x) > 0$ with $c \in C(\overline{\Omega})$,
- $f \in L^2(\Omega)$ and $g \in L^2(\partial\Omega)$.

The Neumann problem is distinctly different from the Dirichlet problem (**??**) previously discussed. For this reason, we will revisit the three-step approximation process.

1 Deriving the Variational Form

To derive the variational form, we multiply both sides of equation (**??**) by a regular test function v and integrate over Ω, obtaining:

$$\int_\Omega (-\Delta u(x) + c(x)u(x))\, v(x)\, dx = \int_\Omega f(x)v(x)\, dx.$$

Using Green's formula (**??**), we have:

$$\int_\Omega (\nabla u(x) \cdot \nabla v(x) + c(x)u(x)v(x))\, dx - \int_{\partial\Omega} \frac{\partial u}{\partial \eta}(x)v(x)\, ds = \int_\Omega f(x)v(x)\, dx.$$

From the Neumann boundary condition, $\dfrac{\partial u}{\partial \eta} = g$ on $\partial\Omega$, so:

$$\int_\Omega (\nabla u(x) \cdot \nabla v(x) + c(x)u(x)v(x))\, dx - \int_{\partial\Omega} g(x)v(x)\, ds = \int_\Omega f(x)v(x)\, dx. \tag{2.9}$$

For the terms in (**??**) to be well-defined, it suffices to choose the Sobolev space $H^1(\Omega)$ (using the trace theorem to justify the boundary integral on $\partial\Omega$).
Thus, the variational form of the boundary value problem (**??**) is:

$$\begin{cases} \text{Find } u \in H^1(\Omega) \text{ such that:} \\ \int_\Omega (\nabla u \cdot \nabla v + cuv)\, dx = \int_{\partial\Omega} gv\, ds + \int_\Omega fv\, dx, \quad \forall v \in H^1(\Omega). \end{cases} \tag{2.10}$$

Remark 2.4.1. *The main difference between the variational form (**??**) with Neumann boundary conditions and the variational form (**??**) with Dirichlet boundary conditions lies in how the boundary conditions are incorporated. Dirichlet conditions constrain the choice of the function space, while Neumann conditions appear naturally in the bilinear form.*

2 Existence and Uniqueness of the Variational Solution

We verify the conditions of the Lax-Milgram theorem:

- **Function Space**: The Hilbert space is $H^1(\Omega)$.
- **Bilinear Form**: The bilinear form a is defined by:

$$\forall u, v \in H^1(\Omega), \quad a(u,v) = \int_\Omega (\nabla u(x) \cdot \nabla v(x) + c(x)u(x)v(x))\, dx.$$

Linear Form

The linear form L is defined by:

$$\forall v \in H^1(\Omega), \quad L(v) = \int_\Omega f(x)v(x)dx + \int_{\partial\Omega} g(x)v(x)ds.$$

Continuity and Coercivity

- **Continuity of a**: The bilinear form a is continuous (bounded) since:

$$|a(u,v)| \leq \|\nabla u\|_{L^2(\Omega)}\|\nabla v\|_{L^2(\Omega)} + \|c\|_\infty \|u\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)} \leq (1+\|c\|_\infty)\|u\|_{H^1(\Omega)}\|v\|_{H^1(\Omega)}.$$

- **Continuity of L**: The linear form L is continuous because:

$$|L(v)| \leq \|f\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)} + \|g\|_{L^2(\partial\Omega)}\|v\|_{L^2(\partial\Omega)}.$$

  By the trace theorem, there exists a constant $M > 0$ such that:

$$|L(v)| \leq \left(\|f\|_{L^2(\Omega)} + M\|g\|_{L^2(\partial\Omega)}\right)\|v\|_{H^1(\Omega)}.$$

Conclusion

The bilinear form a is continuous and coercive (when $c(x) \geq c_0 > 0$), and the linear form L is continuous. Therefore, by the Lax-Milgram theorem, the variational problem admits a unique solution in $H^1(\Omega)$. **Analysis of the Variational Form**

Coercivity of the Bilinear Form The bilinear form a is coercive on $H^1(\Omega)$. For any $v \in H^1(\Omega)$, we have:

$$a(v,v) = \int_\Omega \left(|\nabla v(x)|^2 + c(x)|v(x)|^2\right) dx \geq \min(1, c_0)\|v\|^2_{H^1(\Omega)},$$

where $c_0 = \inf_{x\in\Omega} c(x) > 0$. This establishes the coercivity of a on $H^1(\Omega)$.

Since all conditions of the Lax-Milgram theorem are satisfied, the variational problem (**??**) admits a unique solution $u \in H^1(\Omega)$.

Remark 2.4.2. **a** *The zero-order term ($c(x)u(x)$) was added to ensure coercivity of the bilinear form.*

b *For the Neumann problem **??** with $g = 0$, we cannot naively choose the space:*

$$H = \left\{ v \in H^1(\Omega) \mid \frac{\partial v}{\partial \eta} = 0 \text{ on } \partial\Omega \right\},$$

because for arbitrary $v \in H^1(\Omega)$, the normal derivative $\frac{\partial v}{\partial \eta}$ is not well-defined on $\partial\Omega$.

3 **Equivalence of Problems** To establish equivalence between the problems, we assume the following theorem holds:

Theorem 2.4.1. *Let Ω be a bounded C^1 domain in $\mathbb{R}^n$, $f \in L^2(\Omega)$, and g be the trace of some $H^1(\Omega)$ function on $\partial\Omega$. Then the solution u of the variational problem (**??**) belongs to $H^2(\Omega)$.*

Using Theorem **??** and Green Formula (**??**), for $u, v \in H^1(\Omega)$ we have:

$$\int_\Omega \Delta u(x)v(x)dx = -\int_\Omega \nabla u(x)\cdot\nabla v(x)dx + \int_{\partial\Omega} \frac{\partial u}{\partial \eta}(x)v(x)ds. \tag{2.11}$$

Combining Theorem (**??**) and equation (**??**), we obtain for all $v \in H^1(\Omega)$:

$$\int_\Omega \left(-\Delta u(x) + c(x)u(x) - f(x)\right) v(x)dx = \int_{\partial\Omega} \left(g(x) - \frac{\partial u}{\partial \eta}(x)\right) v(x)ds. \tag{2.12}$$

Equivalence Proof and Main Theorem

Derivation of the PDE Taking $v \in D(\Omega)$ in equation (**??**) causes the boundary term to vanish, yielding:

$$\textstyle\int_\Omega \left(-\Delta u(x) + c(x)u(x) - f(x)\right) v(x)dx = 0 \quad \forall v \in H^1(\Omega).$$

By the fundamental lemma of calculus of variations , we conclude:

$$-\Delta u + cu - f = 0 \quad \text{a.e. in } \Omega.$$

Boundary Condition Recovery For arbitrary $v \in H^1(\Omega)$, we have:

$$\textstyle\int_{\partial\Omega} \left(g(x) - \frac{\partial u}{\partial \eta}(x)\right) v(x)ds = 0.$$

Since the trace operator maps $H^1(\Omega)$ densely to $L^2(\partial\Omega)$, this implies:

$$g - \frac{\partial u}{\partial \eta} = 0 \quad \text{a.e. on } \partial\Omega.$$

Main Existence and Regularity Theorem

Theorem 2.4.2. *Let Ω be a bounded C^1 domain in $\mathbb{R}^n$, $f \in L^2(\Omega)$, g be the trace of some $H^1(\Omega)$ function on $\partial\Omega$, and $c \in C(\overline{\Omega})$ with $c(x) > 0$ for all $x \in \Omega$. Then:*

1. *The variational problem (**??**) admits a unique solution $u \in H^1(\Omega)$.*

2. *Moreover, $u \in H^2(\Omega)$ solves the boundary value problem (**??**) in the weak sense:*

$$\begin{cases} -\Delta u + cu = f & a.e. \text{ in } \Omega, \\ \frac{\partial u}{\partial \eta} = g & a.e. \text{ on } \partial\Omega. \end{cases}$$

Proof. The equivalence follows from:

- The variational solution automatically satisfies the PDE in the distributional sense
- The boundary condition is recovered through the trace theorem
- The H^2 regularity comes from Theorem (**??**) assumptions

■

2.4.6 The Fourier Boundary Value Problem

Problem Statement Let Ω be a bounded open subset of $\mathbb{R}^n$ and consider the diffusion equation with Fourier (Robin) boundary conditions:

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ \frac{\partial u}{\partial \eta} + \lambda u = 0, & \text{on } \partial\Omega, \end{cases} \tag{2.13}$$

where:

- $f \in L^2(\Omega)$ is the source term,
- $\lambda > 0$ is a physical parameter (e.g., heat transfer coefficient),
- η is the outward unit normal vector to $\partial\Omega$.

Physical Interpretation This models phenomena like:

- Heat conduction with convective boundary heat transfer (u = temperature, λ = heat transfer coefficient)
- Diffusion processes with semi-permeable boundaries

Key Features

- Combines aspects of both Dirichlet and Neumann problems
- The boundary condition represents a balance between flux and solution value
- Well-posed for $\lambda > 0$ (coercive variational formulation)

1 Variational Formulation of the Fourier Problem

Derivation of the Weak Form Multiply both sides of (**??**) by a test function $v \in C^\infty(\overline{\Omega})$ and integrate over Ω to obtain:

$$-\int_\Omega \Delta u(x)v(x)dx = \int_\Omega f(x)v(x)dx.$$

Applying Green's formula (integration by parts) yields:

$$\int_\Omega \nabla u(x)\cdot\nabla v(x)dx - \int_{\partial\Omega}\frac{\partial u}{\partial\eta}(x)v(x)ds = \int_\Omega f(x)v(x)dx.$$

Substituting the Fourier boundary condition $\frac{\partial u}{\partial\eta} = -\lambda u$:

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

Variational Formulation The natural function space is $H^1(\Omega)$. The weak form becomes:

$$\begin{cases}\text{Find } u\in H^1(\Omega) \text{ such that:}\\ \int_\Omega \nabla u\cdot\nabla v dx + \int_{\partial\Omega}\lambda uv ds = \int_\Omega fv dx, \quad \forall v\in H^1(\Omega).\end{cases} \tag{2.14}$$

Remark 2.4.3. **a** *Under the assumptions $f\in L^2(\Omega)$ and $\lambda\in L^\infty(\partial\Omega)$, all integrals in (??) are well-defined thanks to the trace theorem.*

b *Recall that if $\varphi\in L^\infty(\Omega)$ and $\psi\in L^2(\Omega)$, then $\varphi\psi\in L^1(\Omega)$.*

2 **Existence and Uniqueness** We verify the conditions of the Lax-Milgram theorem:

- **Hilbert Space**: $H = H^1(\Omega)$
- **Bilinear Form**:

$$a(u,v) = \int_\Omega \nabla u\cdot\nabla v dx + \int_{\partial\Omega}\lambda uv ds \tag{2.15}$$

- **Linear Form**:

$$L(v) = \int_\Omega fv dx \tag{2.16}$$

Analysis of the Bilinear Form

Continuity of a For $u, v\in H^1(\Omega)$, using the Cauchy-Schwarz inequality:

$$\begin{aligned}|a(u,v)| &\le \int_\Omega |\nabla u(x)\cdot\nabla v(x)|dx + \int_{\partial\Omega}|\lambda(x)u(x)v(x)|ds\\ &\le \|\nabla u\|_{L^2(\Omega)}\|\nabla v\|_{L^2(\Omega)} + \|\lambda\|_{L^\infty(\partial\Omega)}\|u\|_{L^2(\partial\Omega)}\|v\|_{L^2(\partial\Omega)}.\end{aligned}$$

By the trace theorem, there exists $C > 0$ such that:

$$|a(u,v)| \leq M\|u\|_{H^1(\Omega)}\|v\|_{H^1(\Omega)}, \quad M = 1 + C^2\|\lambda\|_{L^\infty(\partial\Omega)}.$$

Thus, a is continuous on H.

Continuity of L For $v \in H$:

$$|L(v)| \leq \int_\Omega |f(x)v(x)|dx \leq \|f\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)} \leq \|f\|_{L^2(\Omega)}\|v\|_{H^1(\Omega)}.$$

Thus, L is continuous on H.

Coercivity of a

Theorem 2.4.3. *If there exists $k > 0$ such that $\lambda(x) \geq k$ almost everywhere on $\partial\Omega$, then the bilinear form a defined in (??) is coercive on H.*

Proof. To prove coercivity, we show there exists $\alpha > 0$ such that:

$$\forall v \in H^1(\Omega), \quad a(v,v) \geq \alpha\|v\|^2_{H^1(\Omega)}.$$

Since $v \in H^1(\Omega)$ (not necessarily in $H^1_0(\Omega)$), Poincaré's inequality doesn't apply directly. We proceed by contradiction: Assume a is not coercive. Then:

$$\forall\alpha > 0, \exists v \in H^1(\Omega) \text{ such that } a(v,v) < \alpha\|v\|^2_{H^1(\Omega)}.$$

Taking $\alpha = \frac{1}{n}$, we construct a sequence $(v_n) \subset H^1(\Omega)$ with:

$$a(v_n, v_n) < \tfrac{1}{n}\|v_n\|^2_{H^1(\Omega)} \quad \forall n \in \mathbb{N}.$$

This leads to a contradiction with the lower bound on λ, proving coercivity. ■

Proof of Coercivity and Equivalence

Compactness Argument Since (v_n) is a bounded sequence in $H^1(\Omega)$, by Rellich's theorem, there exists a subsequence (still denoted v_n) that converges strongly in $L^2(\Omega)$. We then have:

$$a(v_n, v_n) = \int_\Omega |\nabla v_n(x)|^2 dx + \int_{\partial\Omega} \lambda(x)|v_n(x)|^2 ds.$$

Taking the limit in the inequality yields:

$$\lim_{n\to+\infty} \int_\Omega |\nabla v_n(x)|^2 dx = \lim_{n\to+\infty} \int_{\partial\Omega} \lambda(x)|v_n(x)|^2 ds = 0. \tag{2.17}$$

Weak Convergence Analysis This implies:

- (∇v_n) converges weakly to 0 in $L^2(\Omega)$
- For all test functions $\varphi \in D(\Omega)$:

$$\lim_{n\to+\infty} \int_\Omega \frac{\partial v_n}{\partial x_i}\varphi(x)dx = 0 \quad \forall i \in \{1, ..., n\}$$

- By weak convergence of (v_n) in $L^2(\Omega)$:

$$\int_\Omega v(x)\frac{\partial \varphi}{\partial x_i}(x)dx = 0$$

Conclusion of Coercivity Proof This shows that the weak derivative of v is zero, hence v is constant. From the boundary condition $\lambda \geq k > 0$, we conclude $v = 0$ on $\partial\Omega$, and thus $v = 0$ on Ω by the trace theorem.

This contradicts the assumption that $\|v_n\|_{H^1(\Omega)} = 1$, proving the coercivity of a on $H^1(\Omega)$.

3 Equivalence of Problems

Assuming regularity, applying Green's formula (**??**) for $u, v \in H^1(\Omega)$:

$$\int_\Omega \Delta u(x)v(x)dx = -\int_\Omega \nabla u(x)\cdot\nabla v(x)dx + \int_{\partial\Omega} \frac{\partial u}{\partial \eta}(x)v(x)ds. \tag{2.18}$$

Theorem 2.4.4. *Under the given conditions, the variational problem (**??**) admits a unique solution $u \in H^1(\Omega)$, which weakly solves the original boundary value problem (**??**)*

Equivalence Proof for the Fourier Problem

Derivation of the PDE Substituting into (**??**) yields:

$$-\int_\Omega \Delta u(x)v(x)dx + \int_{\partial\Omega} \tfrac{\partial u}{\partial \eta}(x)v(x)ds + \int_{\partial\Omega} \lambda u(x)v(x)ds = \int_\Omega f(x)v(x)dx.$$

Rearranging terms gives:

$$\int_\Omega (\Delta u(x) + f(x))v(x)dx = \int_{\partial\Omega} \left(\frac{\partial u}{\partial \eta}(x) + \lambda u(x)\right)v(x)ds. \tag{2.19}$$

Recovery of the PDE Taking $v \in D(\Omega)$ (test functions with compact support in Ω) causes the boundary term to vanish:

$$\textstyle\int_\Omega (\Delta u + f)v dx = 0 \quad \forall v \in D(\Omega).$$

By the Fundamental Lemma of Calculus of Variations , we conclude:

$$\Delta u + f = 0 \quad \text{a.e. in } \Omega.$$

Recovery of Boundary Conditions For arbitrary $v \in H^1(\Omega)$, we now have:

$$\int_{\partial\Omega} \left(\frac{\partial u}{\partial \eta} + \lambda u \right) v ds = 0.$$

Since the trace operator maps $H^1(\Omega)$ densely to $L^2(\partial\Omega)$, this implies:

$$\frac{\partial u}{\partial \eta} + \lambda u = 0 \quad \text{a.e. on } \partial\Omega.$$

Theorem 2.4.5. *The solution $u \in H^1(\Omega)$ of the variational problem (??) satisfies:*

$$\begin{cases} -\Delta u = f & a.e. \ in\ \Omega, \\ \frac{\partial u}{\partial \eta} + \lambda u = 0 & a.e. \ on\ \partial\Omega, \end{cases}$$

thus solving the original Fourier problem (??) in the weak sense.

2.5 The Principles ideas of the finite elements method

The basic idea of the finite element method is to replace the Hilbert space V, on which the variational formulation is posed, with a finite-dimensional subspace V_h. The approximate problem posed on V_h reduces to simply solving a linear system, whose matrix is called the stiffness matrix. Furthermore, we can choose the method of constructing V_h in such a way that the subspace V_h is a good approximation of V and that the solution uh in V_h of the variational formulation is "close" to the exact solution u in V, So , Dealing with the approximated problem means we will get an internal variational approximation which posed on the Space V_h .

2.6 Internal Variational Approximation

Let a be a bilinear, continued and coercive form, L is continued and linear form, Considering the variational form on a Hilbert Space V the following problem:

$$\begin{cases} \text{Find}\, u \in V \,\text{such that} \\ a(u\,,\,v) = L(v), \quad \text{For all}\, v \in V \end{cases} \tag{2.20}$$

This Variational form is verifying the conditions of LaX-Milgram theorem, So , there exist a unique solution for it, by replacing the space V by a finite dimension subspace V_h we'll search for a solution for the following problem:

$$\begin{cases} \text{Find}\, u_h \in V_h \,\text{such that} \\ a(u_h\,,\,v_h) = L(v_h), \quad \text{For all}\, v_h \in V_h \end{cases} \tag{2.21}$$

Lemma 2.6.1. *Let V be a real Hilbert space, and V_n a finite-dimensional subspace. Let $a(u\,,\,v)$ be a continuous and coercive bilinear form on V, and $L(v)$ a continuous linear form on V. Then the internal approximation (??) admits a unique solution. Moreover, this solution can be obtained by solving a linear system with a positive definite matrix (and symmetric if $a(u,v)$ is symmetric).*

Proof. The existence and uniqueness of $u_h \in V_h$, solution of (??), follows from the Lax-Milgram theorem applied to V_h To put the problem under a In a simpler form, we introduce a basis $(\phi_j)_{1\leq j\leq N_h}$ of V_h. If $u_h = \sum_{j=1}^{N_h} u_j\phi_j$, we let $u_h = (u_1, ..., u_{N_h})$ be the vector in $\mathbb{R}^{N_h}$ of the coordinates of u_h .

Problem (??) is equivalent to:

$$\text{find } u_h \in \mathbb{R}^{N_h} \text{ such that } \textstyle\sum_{j=1}^{N_h} a(u_j\phi_j, \phi_i) = L(\phi_i) \quad \forall 1 \leq i \leq N_h$$

which can be written in the form of a linear system :

$$K_h u_h = b_h \tag{2.22}$$

with, for $1 \leq i, j \leq N_h$,

$$(K_h)_{ij} = a(\phi_j, \phi_i), \quad (b_h)_i = L(\phi_i) \tag{2.23}$$

The coercivity of the bilinear form $a(u, v)$ implies the positive definite character of the matrix K_h, and thus its invertibility. Indeed, for any vector $u_h \in \mathbb{R}^{N_h}$, we have :

$$\exists C > 0, \text{ such that } \quad \langle K_h u_h\, ,\, u_h\rangle_V \geq C\, \|u_h\|^2$$

since all norms are equivalent in finite dimension ($\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^{N_h}$). Similarly, the symmetry of $a(u, v)$ implies that of K_h. In mechanical applications, the matrix K_h is called the stiffness matrix. ■

Now , We will compare the error comitted on replacing the Space V by it's subspace V_h ,

2.6.1 Jean Céa's Lemma

Lemma 2.6.2. *We place ourselves under the hypotheses of lemma ?? Let u be the solution of (??) and u_h that of (??). We have*

$$\|u - u_h\| \leq \frac{M}{C} \inf_{v_h\in V_h} \|u - v_h\|, \quad \textit{for all} v_h \in V_h \tag{2.24}$$

Proof. Since $V_h \subset V$, we deduce, by subtracting the variational formulations in (??) and (??), that

$$a(u - u_h, w_h) = 0 \quad \forall w_h \in V_h$$

By choosing $w_h = u_h - v_h$ we obtain

$$\nu\|u - u_h\|^2 \leq a(u - u_h, u - u_h) = a(u - u_h, u - v_h) \leq M\|u - u_h\|\|u - v_h\|$$

from which we deduce the result . ■

2.6.2 Interpolation Operator

Definition 2.6.1. *We place ourselves under the hypotheses of lemma* **??** *We assume that there exists a subspace* $V_h \subset V$ *dense in* V *and an application* r_h *from* V *to* V_h *(called interpolation operator) such that*

$$\lim_{h\to 0} \|v - r_h(v)\| = 0 \quad \forall v \in V. \tag{2.25}$$

Then the internal variational approximation method converges, that is to say that

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

Proof. Let $\varepsilon > 0$. By density of V_h, there exists $v_h \in V_h$ such that $\|u - v_h\| \leq \varepsilon$. Moreover, there exists an $h_0 > 0$ (depending on ε) such that, for this element we have

$$\|v - r_h(v)\| \leq \varepsilon \quad \forall h \leq h_0.$$

By virtue of Lemma of Céa, we have

$$\|u - u_h\| \leq C\|u - r_h(v)\| \leq C(\|u - v\| + \|v - r_h(v)\|) \leq 2C\varepsilon,$$

from which we deduce the result. ■

2.6.3 Conditions On The subSpace V_h

The idea of this method is much clear, starting from the variational form (**??**), choosing a finite dimension subspace V_h, then solving a linear system associated with the internal variational (**??**) form . However, the choosing of the Subspace V_h is not random, it must verify this two conditions :

1. Being able to formulate an interpolation operator r_h from V in V_h which verify (**??**) .
2. The solution for the linear system $K_h u_h = b_h$ must be economic.

2.7 Weak formulation of Partial Differential Equations

2.7.1 Historical perspective

By " Finite Element method ", we denote a family of approaches developed to compute an approximate solution to a partial differential equation (PDE). The physics of phenomena encountered in engineering applications is often modeled under the form of a boundary value problem. Equations describing the evolution in time are called initial value problems and consist of the coupling of an ordinary differential equation (ODE) in time with a boundary value problem in space.

The study of equations involving derivatives of the unknown has lead to rethinking the concept of derivation: from the idea of variation, then the study of the Cauchy problem, finally to the generalization of the notion of derivative with the Theory of Distributions.

2.7.2 Weak solution to the Dirichlet problem

Let us consider the Poisson problem posed in a domain Ω, an open bounded subset of $\mathbb{R}^d$ where $d \geq 1$ supplemented with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\Delta u(x) = f(x), \quad x \in \Omega \\ u(s) = 0, \quad s \in \partial\Omega \end{cases} \tag{2.26}$$

with $f \in C(\overline{\Omega})$ and the Laplace operator defined by

$$\Delta u(x) = \sum_{k=1}^{d} \frac{\partial^2 u}{\partial x_k^2}(x)$$

thus involving second order partial derivatives of the unknown u with respect to the space coordinates.

Definition 2.7.1 (Classical solution). *A classical solution (or strong solution) of Problem (**??**) is a function $u \in C^2(\Omega)$ satisfying relations of the problem.*

Problem (**??**) can be reformulated so as to look for a solution in the distributional sense by testing the equation against smooth functions. Reformulating the problem amounts to relaxing the pointwise regularity (i.e. continuity) required to answer the existence of the classical derivative to the (weaker) existence of the distributional derivative which regularity is to be interpreted in term in terms of Lebesgue spaces: the obtained problem is a weak formulation and a solution to this problem (i.e. in the distributional sense) is called weak solution. Three properties of the weak formulation should be studied:

1. firstly that a classical solution is a weak solution,
2. secondly that such a weak solution is indeed a classical solution provided that it is regular enough.
3. thirdly that the well-posedness of this reformulated problem, i.e. existence and uniqueness of the solution, is ensured.

2.7.3 Formal passage from classical solution to weak solution

Let $u \in C^2(\overline{\Omega})$ be a classical solution to (**??**) and let us test the first equation of (**??**) against any smooth function $\varphi \in C_c^\infty(\Omega)$:

$$-\int_\Omega \Delta u(x)\varphi(x)dx = \int_\Omega f(x)\varphi(x)dx \tag{2.27}$$

Since $u \in C^2(\overline{\Omega})$, Δu is well defined. Integrating by parts, the left-hand side we obtain

$$-\int_\Omega \Delta u(x)\varphi(x)dx = -\int_{\partial\Omega} \frac{\partial u(x)}{\partial n}\varphi(x)ds + \int_\Omega \nabla u(x)\cdot\nabla\varphi(x)dx$$

Where $\frac{\partial u}{\partial n} = \langle \nabla u\,,\, n\rangle$. Now, for simplicity, we recall the one-dimensional case:

$$-\int_0^1 \frac{\partial^2 u(x)}{\partial x^2}\varphi(x)dx = -\left[\frac{\partial u(x)}{\partial x}\varphi(x)\right]_0^1 + \int_0^1 \frac{\partial u(x)}{\partial x}\frac{\partial\varphi(x)}{\partial x}dx$$

Since φ has compact support in Ω, it vanishes on the boundary $\partial\Omega$, consequently the boundary integral is zero, thus the distributional formulation reads

$$\int_\Omega \nabla u(x)\cdot\nabla\varphi(x)dx = \int_\Omega f(x)\varphi(x)dx, \quad \forall\varphi \in C_c^\infty(\Omega)$$

and we are lead to look for a solution u belonging to a functional space such that the previous relation makes sense.

A weak formulation of Problem (**??**) consists in solving: Find $u \in H$ and a function $f \in H'$ such that

$$\int_\Omega \nabla u\cdot\nabla v\,dx = \int_\Omega fv\,dx, \quad \forall v \in V \tag{2.28}$$

in which H and V are functional spaces yet to be defined, both satisfying regularity constraints and for H boundary condition constraints.

2.7.4 Formal passage from weak solution to classical solution

Provided that the weak solution to Problem (**??**) belongs to $C^2(\overline{\Omega})$ then the second derivatives exist in the classical sense. Consequently the integration by parts can be performed the other way around and the weak solution is indeed a classical solution.

2.7.5 About the boundary conditions

Boundary condition	Expression on $\partial\Omega$	Property
Dirichlet	$u = u_D$	"essential" boundary condition
Neumann	$\nabla u\cdot n = 0$	"natural" boundary condition

Essential boundary conditions are embedded in the functional space, while natural boundary conditions appear in the weak formulation as linear forms.

2.8 Weak and variational formulations

2.8.1 Functional setting

Hilbert-Sobolev spaces H^s are a natural choice to "measure" functions involved in the weak formulations of PDEs as the existence of the integrals relies on the fact that integrals of powers

$|\cdot|^p$ of u and weak derivatives $D^\alpha \overline{u}$ for some $1 \leq p < +\infty$ exist:

$$H^s(\Omega) = \{u \in L^2(\Omega) : D^\alpha u \in L^2(\Omega), 1 \leq |\alpha| \leq s\}$$

with the Lebesgue space of square integrable functions on Ω:

$$L^2(\Omega) = \left\{u : \quad \int_\Omega |u(x)|^2 dx < +\infty\right\}$$

endowed with its natural scalar product

$$(u, v)_{L^2(\Omega)} = \int_\Omega uv\, dx$$

Since Problem (**??**) involves first order derivatives according to relation,

$$\int_\Omega \nabla u \cdot \nabla v\, dx = \int_\Omega fv\, dx$$

Then we should consider a solution in $H^1(\Omega)$.

$$H^1(\Omega) = \{u \in L^2(\Omega) : Du \in L^2(\Omega)\}$$

with the weak derivative Du i.e. a function of $L^2(\Omega)$ which identifies with the classical derivative (if it exists) "almost everywhere", and endowed with the norm,

$$\|\cdot\|_{H^1(\Omega)} = (\cdot, \cdot)^{1/2}_{H^1(\Omega)}$$

defined from the scalar product,

$$(u, v)_{H^1(\Omega)} = \int_\Omega uv\, dx + \int_\Omega \nabla u \cdot \nabla v\, dx$$

Moreover, the solution should satisfy the boundary condition of the strong form of the PDE problem. The homogeneous Dirichlet condition is embedded in the functional space of the solution: u vanishing on the boundary $\partial\Omega$ yields that we should seek u in $H^1_0(\Omega)$.

2.8.2 Determination of the solution space

We will now establish that any weak solution lives in $H^1_0(\Omega)$.

Choice of test space

In order to give sense to the solution in a Sobolev space we need to choose the test function itself in the same kind of space. Indeed $C^\infty_c(\Omega)$ is not equipped with a topology which allows us to work properly. If we choose $\varphi \in H^1_0(\Omega)$ then by definition, we can construct a sequence $(\varphi^n)_{n\in\mathbb{N}}$ of functions in $C^\infty_c(\Omega)$ converging in $H^1_0(\Omega)$ to φ, i.e.

$$\|\varphi^n - \varphi\|_{H^1(\Omega)} \to 0, \text{ as } n \to +\infty$$

For the sake of completeness, we show that we can pass to the limit in the formulation, term by term for any partial derivative:

$$\int_\Omega \partial_i u \partial_i \varphi^n \to \int_\Omega \partial_i u \partial_i \varphi$$

as $\partial_i \varphi^n \to D_i \varphi$ in $L^2(\Omega)$, which denotes the weak convergence i.e. tested on functions of the dual space (which, in case of $L^2(\Omega)$ is $L^2(\Omega)$ itself).

$$\int_\Omega f \varphi^n \to \int_\Omega f \varphi$$

As $\varphi^n \to \varphi$ in $L^2(\Omega)$. Consequently, the weak formulation is satisfied if $\varphi \in H_0^1(\Omega)$.

Choice of solution space

The determination of the functional space is guided, firstly, by the regularity of the solution: if u is a classical solution then it belongs to $C^2(\overline{\Omega})$ which involves that $u \in L^2(\Omega)$ and $\partial_i u \in L^2(\Omega)$, thus $u \in H^1(\Omega)$, secondly by the boundary conditions: the space should satisfy the Dirichlet boundary condition on $\partial\Omega$. This constraint is satisfied thanks to the following trace theorem for the solution to the Dirichlet problem: since $\ker(\gamma) = H_0^1(\Omega)$, we conclude $u \in H_0^1(\Omega)$.

2.8.3 Well-posedness

In the usual sense, a well-posed problem admits a unique solution which is bounded in the V-norm by the data (forcing term, boundary conditions). In this particular case of the Poisson problem the bilinear form $a(\cdot,\cdot)$ is the natural scalar product in $H_0^1(\Omega)$, thus it defines a norm in $H_0^1(\Omega)$ (but only a seminorm in $H^1(\Omega)$ due to the lack of definiteness, not a norm!).

Theorem 2.8.1 (Riesz-Fréchet)**.** *Let H be a Hilbert space and H' its topological dual, $\forall \phi \in H'$ there exists a unique representant $u \in H$ such that for any $v \in H$,*

$$\phi(v) = (u, v)_H$$

and furthermore $\|u\|_H = \|\Phi\|_{H'}$

This result ensures directly the existence and uniqueness of a weak solution as soon as $a(\cdot,\cdot)$ is a scalar product and ϕ is continuous for $\|\cdot\|_a$. Now that we have derived a variational problem for which there exists a unique solution with V infinite dimensional (i.e. for any point $x \in \Omega$), we need to construct an approximate problem which is also well-posed.

2.9 Galerkin's Method

After seeing the basic idea of the finite elements method , we can get it more detailed , This method has two fundamental principles , The first one is **Galerkin's Method** which very useful in the theorique view, it depends on the internal variational form which previously defined .

We assume that the Hilbert space V is separable and of infinite dimension, which implies , that there exists a Hilbertian basis $(e_i)_{i\geq 1}$ of V. We then choose V as the subspace spanned by this Hilbertian basis (spanned by finite linear combinations), which is of course dense in V. By setting $h = 1/n$, we define V_h as the finite-dimensional subspace spanned by $B = (e_1, ..., e_n)$. Finally, the interpolation operator r_h is simply the orthogonal projection onto V_h (which is defined here in all of V and not just in V_h).
All the hypotheses of The 3 Lemma's are therefore satisfied, and we deduce that the approximate solution u_h converges to the exact solution u. Recall that u_h is calculated by solving the linear system $K_h u_h = b_h$, where u_h is the vector in $\mathbb{R}^n$ of the coordinates of u_h in the basis B.

2.9.1 Modeled Problem

Let's assume on the domain $]a, b[= \Omega$ two functions $\alpha : \mathbb{R} \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$. And let's look for the function u, where:

$$\begin{cases} -(\alpha u')' = f, & \text{in} \quad \Omega \\ u(a) = u(b) = 0, & \text{on} \quad \partial\Omega \end{cases} \tag{2.29}$$

Let u be a solution for both (**??**). Hence, by multiplying both sides of equation (**??**) by a test function $v \in V$, we find:

$$(-\alpha u')' = fv$$

Integrating by parts, we find:

$$\int_\Omega \alpha u'v'dx - [\alpha u'v]_a^b = \int_\Omega fvdx$$

Applying the boundary conditions specified in (**??**), and since v vanishes at a and b, we find:

$$\int_\Omega \alpha u'v'dx = \int_\Omega fvdx, \quad \forall v \in V$$

Let $f \in L^2(\Omega)$ such that the second equation of (**??**), and we seek an approximate solution to problem. Therefore, we replace the infinite-dimensional space H by a finite-dimensional subspace V_h. We note that $V_h \subset H$. This subspace is called the approximate space of problem. We find that the latter is posed in the space as follows:

Find $v_h \in V_h$ such that:

$$\int_\Omega \alpha u_h' v_h' dx = \int_\Omega f v_h dx, \quad \forall v_h \in V_h \tag{2.30}$$

And we say that problem (**??**) is the general internal approximation of (**??**) on V_h. The method described above is called the Galerkin method, and we have presented it in its simplest forms.

Let the family $(\varphi_i)_{i=1}^N$ be a basis for the space V_h, where N is a finite number representing the dimension of V_h. By decomposing the approximate solution u_h on this basis, we find:

$$u_h = \sum_{i=1}^{N} u_i \varphi_i \tag{2.31}$$

By setting: $\varphi_j = v_h$, where φ_j is an element of the basis of the approximate space, we find:

$$a\left(\sum_{i=1}^{N} u_i \varphi_i, \varphi_j\right) = L(\varphi_j)$$

And by taking:

$$\int_\Omega \alpha u_h' v_h' dx = a(u_h, v_h) \quad \text{and} \quad \int_\Omega f v_h dx = L(v_h)$$

And since $a(.,.)$ is a bilinear form, then:

$$\sum_{i=1}^{N} a(\varphi_i, \varphi_j) u_i = L(\varphi_j)$$

Hence, we obtain a system of linear equations with unknowns $u_1, u_2, ..., u_N$ And we associate with it the following matrix notation:

$$\sum_{i=1}^{N} a(\varphi_i, \varphi_j) u_i = L(\varphi_j), \quad j = \overline{1:N} \tag{2.32}$$

Equivalent to

$$\begin{pmatrix} a(\varphi_1,\varphi_1) & a(\varphi_2,\varphi_1) & \cdots & a(\varphi_N,\varphi_1) \\ a(\varphi_1,\varphi_2) & a(\varphi_2,\varphi_2) & \cdots & a(\varphi_N,\varphi_2) \\ \vdots & \vdots & \ddots & \vdots \\ a(\varphi_1,\varphi_N) & a(\varphi_2,\varphi_N) & \cdots & a(\varphi_N,\varphi_N) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} L(\varphi_1) \\ L(\varphi_2) \\ \vdots \\ L(\varphi_N) \end{pmatrix}$$

equivalent to

$$A_h u_h = B_h$$

Theorem 2.9.1. *If the bilinear form $a(\cdot,\cdot)$ is continuous and coercive, then the system $A_h u_h = B_h$ admits a unique solution. In addition, if $a(\cdot,\cdot)$ is symmetric and coercive, then the matrix A_h is symmetric and positive definite.*

Proof. Let $A_h u_h = 0$, then each equation of the system of equations vanishes, and hence:

$$\begin{aligned} \sum_{j=1}^{N} a(\varphi_j, \varphi_i) u_j = 0, i = \overline{1,N} &\Leftrightarrow a\left(\sum_{j=1}^{N} u_j \varphi_j, \varphi_i\right) = 0, && i = \overline{1,N} \\ &\Leftrightarrow a\left(\sum_{j=1}^{N} u_j \varphi_j, \sum_{i=1}^{N} u_i \varphi_i\right) = 0, && \forall i = \overline{1,N} \\ &\Leftrightarrow a\left(\sum_{j=1}^{N} u_j \varphi_j, \sum_{j=1}^{N} u_j \varphi_j\right) = 0, && \forall i = \overline{1,N} \\ &\Leftrightarrow a\left(\sum_{i=1}^{N} u_i \varphi_i, \sum_{i=1}^{N} u_i \varphi_i\right) = 0 : && \forall i = \overline{1,N} \end{aligned}$$

By taking $v = \sum_{i=1}^{N} u_i\varphi_i$ and $i = j$, we find $a(v, v) = 0$. And since $a(\cdot, \cdot)$ is coercive, we find:

$$0 \leq \alpha||v||^2 \leq a(v, v) = 0 \Rightarrow ||v|| = 0$$

implies

$$\sum_{i=1}^{N} u_i\varphi_i = 0 \Rightarrow u_i = 0, \quad \forall i = \overline{1, N}$$

And hence, $\ker(A) = \{0\}$, and thus the matrix A_h is injective, and since the co-domain is the finite-dimensional subspace V_h, then it is surjective. Therefore, A_h is bijective, and hence invertible, which implies that the system $A_h u_h = B_h$ admits a unique solution.

Now, we have from $a(\cdot, \cdot)$ being symmetric that

$$a_{ij} = a(\varphi_i, \varphi_j) = a(\varphi_j, \varphi_i) = a_{ji}$$

and hence $A_h = A_h^T$. Therefore, A_h is symmetric.

Proof that positive definiteness is equivalent to proving that

$$v_h^T A_h v_h \geq 0, \quad \forall v_h \in \mathbb{R}^N$$

Where we have:

$$u_h = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} \quad \text{and} \quad A_h = [a(\varphi_i, \varphi_j)]_{i,j}, \quad 1 \leq i, j \leq N$$

Then it follows:

$$\begin{aligned} \langle A_h u_h \,,\, u_h \rangle = u_h^T A_h u_h &= \sum_{j=1}^{N}\sum_{i=1}^{N}(a_{ij}u_i)u_j \\ &= \sum_{j=1}^{N}\sum_{i=1}^{N}(a(\varphi_i, \varphi_j)u_i)u_j \\ &= \sum_{j=1}^{N} a\left(\sum_{i=1}^{N} u_i\varphi_i, \varphi_j\right) u_j \\ &= a\left(\sum_{i=1}^{N} u_i\varphi_i, \sum_{j=1}^{N} u_j\varphi_j\right) \\ &= a\left(\sum_{i=1}^{N} u_i\varphi_i, \sum_{i=1}^{N} u_i\varphi_i\right) \end{aligned}$$

By taking $v = \sum_{i=1}^{N} u_i\varphi_i$ and since $a(\cdot, \cdot)$ is coercive, we find:

$$u_h^T A_h u_h = a\left(\sum_{i=1}^{N} u_i\varphi_i, \sum_{i=1}^{N} u_i\varphi_i\right) \geq \alpha \left\|\sum_{i=1}^{N} u_i\varphi_i\right\|^2 \geq 0$$

Therefore, A_h is positive definite. Let the vector u_h from $\mathbb{R}^N$ formed by the components in this basis be $u_h = (u_i)_{1\leq i\leq N}$, and we call the matrix $A_h \in M_{N\times N}(\mathbb{R})$ the stiffness matrix, which is given by:

$$A_{ij} = (a_{ij}) = [a(\varphi_i, \varphi_j)] = \int_\Omega \alpha\varphi_i'\varphi_j' dx, \quad (i, j) \in \mathbb{N}^* \times \mathbb{N}^*$$

And let the vector $F_h \in \mathbb{R}^N$ be defined by:

$$L(\varphi_j) = \int_\Omega f\varphi_j = b_j, \quad j = \overline{1, N} \tag{2.33}$$

And it appears to us computationally that a solution to (**??**) is if and only if it is a solution to the linear system:

$$A_h u_h = B_h \tag{2.34}$$

■

Remark 2.9 *1 The Galerkin method allows replacing the posed problem with infinite dimension by a linear system or a finite dimension.*

2 The "Galerkin" method is a special case of the finite element method.

2.10 Finite Elements Method

The principle of the finite element method is to construct internal approximation spaces V_h, usual functional spaces $H^1(\Omega)$, $H_0^1(\Omega)$, $H^2(\Omega)$, ...
The definition is based on the geometric notion of meshing the domain Ω. A mesh is a paving of the space into very simple elementary volumes: triangles, tetrahedra, parallelepipeds .

In this context, the parameter h of V_h corresponds to the maximum size of the meshes or cells that compose the mesh. Typically, a basis of V_h will be constituted of functions whose support is localized on one or a few meshes. This will have two important consequences: on the one hand, in the limit $h \to 0$, the space V_h will be increasingly "large" and will better approximate the entire space V, and on the other hand, the stiffness matrix K_h of the linear system $K_hU_h = b_h$ will be sparse, that is to say, most of its coefficients will be zero (which will limit the cost of numerical resolution).

The finite element method is one of the most effective and popular methods for numerically solving boundary value problems. It is the basis of countless industrial calculation software.

2.11 Finite Element spaces

In the previous lectures we have studied the properties of coercive problems in an abstract setting and described Galerkin methods for the approximation of the solution to a PDE, respectively in the case of symmetric and non-symmetric bilinear forms.

The abstract setting reads:

$$\text{Find } u_h \in V_h \subset H \text{ such that:} \quad a(u_h, v_h) = L(v_h), \quad \forall v_h \in V_h$$

such that:

- V_h is a finite dimensional approximation space characterized by a discretization parameter h,
- $a(\cdot,\cdot)$ is a continuous bilinear form on $V_h \times V_h$, coercive w.r.t $\|\cdot\|_{V_h}$,
- $L(\cdot)$ is a continuous linear form.

Under these assumptions existence and uniqueness of a solution to the approximate problem holds owing to the Lax-Milgram Theorem and u_h is called discrete solution. Provided this abstract framework which allows us to seek approximate solutions to PDEs, we need to chose the approximate space V_h and construct a basis $\mathcal{B} = \{\varphi_1, \ldots, \varphi_{N_h}\}$ of V_h on which the discrete solution is decomposed:

$$u_h = \sum_{j=1}^{N_{V_h}} u_j \varphi_j$$

with $N_{V_h} = \dim(V_h)$, $\{u_j\}$ a family of N_{V_h} real numbers called *global degrees of freedom* and $\{\varphi_j\}$ a family of N_{V_h} elements of V_h called *global shape functions.*

To construct the approximate space V_h, we need two ingredients:

1 An admissible mesh $\mathcal{T}_h$ generated by a tesselation of domain Ω.

2 A reference finite element $(\hat{K}, \hat{P}, \hat{\Sigma})$ to construct a basis of V_h.

2.11.1 Admissible mesh

Let Ω be polygonal ($d = 2$) or polyhedral ($d = 3$) subset of $\mathbb{R}^d$, we define $\mathcal{T}_h$ (a triangulation in the simplicial case) as a finite family $\{K_i\}$ of disjoints non-empty subsets of Ω named cells. Moreover $\mathcal{N}_h = \{N_i\}$ denotes the set a vertices of $\mathcal{T}_h$ and $\mathcal{E}_h = \{\sigma_{KL} = K \cap L\}$ denotes the set of edges.

Also The Mesh size is :

$$h_{\mathcal{T}} = \max_{K \in \mathcal{T}_h}(\operatorname{diam}(K))$$

Definition 2.11.1. *A mesh is said geometrically conforming if two neighbouring cells share either exactly one vertex, exactly one edge, or in the case $d = 3$ exactly one facet.*

The meaning of the previous condition is that there should not be any “hanging node” on a facet. Moreover some theoretical results require that the mesh satisfies some regularity condition: for example, bounded ratio of equivalent ball diameter, Delaunay condition on the angles of a triangle, ...

2.11.2 Numerical integration

The contributions are integrated numerically, usually using quadrature rules.

2.11.3 Finite elements method

Solving a problem by a Finite Element Method is defined by the following procedure:

1. Choose a reference finite element $(\hat{K}, \hat{P}, \hat{\Sigma})$.
2. Construct an admissible mesh $\mathcal{T}_h$ such that any cell $K \in \mathcal{T}_h$ is in bijection with the reference cell $\hat{K}$.
3. Define a mapping to transport the reference finite element defined on $\hat{K}$ onto any $K \in \mathcal{T}_h$ to (K, P, Σ).
4. Construct a basis for V_h by collecting all the finite element basis of finite elements $\{(K, P, \Sigma)\}_{K \in \mathcal{T}_h}$ sharing the same degree of freedom.

Remark 2.11.1. *The Finite Element approximation is said H-conformal if* $V_h \subset H$ *and is said non-conformal if* $V_h \not\subset H$. *In this latter case the approximate problem can be constructed by building an approximate bilinear form*

$$a_h(\cdot, \cdot) = a(\cdot, \cdot) + s(\cdot, \cdot)$$

Chapter 3

A posteriori error estimates for Laplace problem

3.1 Numerical Methods

For all the model problems presented , it is typically impossible to find the exact solution (u or the couple u, p). Thus, numerical methods are used to find an approximate solution. Such methods rely on a notion of a spatial mesh, a partition of the domain Ω into elements that we call K. Herein, we suppose that the elements K are simplices (triangles in two space dimensions and tetrahedra in three space dimensions). We also suppose that the intersection of two elements K and K' is either an empty set, their common vertex, their common edge, or their common face if d = 3. The letter h stands for the maximal diameter of the elements and Th for the mesh itself. We will typically denote the approximate solutions uh for Some problems .

3.2 A priori error estimates

Traditionally, the quality of numerical solutions is expressed with the aid of a priori error estimates. These estimates have typically, for steady-state problems, the form

$$\|u - u_h\| \leq Ch^k \tag{3.1}$$

Where $C \geq 0$ and $k \geq 0$ are constants and $\| . \|$ is some norm (recall that u is the exact solution, u_h the approximate solution of the problem , and h (the mesh size). It can be concluded that the error between u and u_h goes to zero as h goes to zero (with the order k), which justifies the numerical method in question: When we refine the mesh T_h (add elements and decrease the maximal element size h), the approximate solution approaches the exact one. Unfortunately, the constant $C = C(u)$ typically depends on the exact solution u and is unknown. Thus, C_h^k cannot be evaluated in general in practice and one cannot obtain a computable upper bound on the error. In particular, property from the Introduction cannot be achieved. Remark also that Ch^k can in fact be evaluated prior to the calculation, whence the name of this estimate.

3.3 A Posteriori error estimates

A posteriori error estimates aim at giving bounds on the error between the known numerical approximation and the unknown exact solution that can be computed in practice, once the approximate solution is known. For a stationary problem, they typically take the form :

$$\|u - u_h\| \leq \left(\sum_{K \in T_h} \eta_K^2 \right)^{\frac{1}{2}} \tag{3.2}$$

Where $\eta_K = \eta_K(u_h)$ is a quantity linked to the mesh element K, computable from u_h; This quantity is called an element estimator.

One may formulate the following five properties describing an optimal a posteriori error estimate for a stationary problem:

1 ensure that (**??**) holds and that η_K is fully computable from u_h (guaranteed upper bound);

2 ensure that, for all $K \in T_h$, η_K represents, up to a generic constant, a lower bound for the actual error in the vicinity of K, i.e., that there exists a constant $C \geq 0$ such that

$$\eta_K \leq C \|u - u_h\|_{S(K)}, \quad \forall K \in T_h \tag{3.3}$$

Where $S(K)$ stands for the element K and it's neighbours. That is

$$S(K) = \{K' \in T_h, \quad K \cap K' \neq \emptyset\}$$

3 Ensure that the effectivity index, given as

$$I_{\text{eff}} = \frac{\left(\sum_{k \in \mathcal{T}_h} \eta_K^2\right)^{\frac{1}{2}}}{\|u - u_h\|} \tag{3.4}$$

i.e., as the ratio of the estimated and actual error, goes to one as the computational effort grows (asymptotic exactness);

4 guarantee the three previous properties independently of the parameters and of their variation (robustness);

5 give estimators η_K which can be evaluated locally (in the element K or on its neighborhood T_K (small evaluation cost).

3.4 The Laplace equation in one space dimension

Let us, for the sake of clarity, start with the Laplace equation in one space dimension , i.e., with Ω being an interval. Many concepts will be clear form this simple model case : Let $f \in L^2(\Omega)$

$$\begin{cases} -u'' = f, & \text{in}\,\Omega \\ u = 0, & \text{on}\,\partial\Omega \end{cases} \tag{3.5}$$

Let us first recall that (**??**) does not have a classical solution (i.e., $u \in C^2(\Omega)$ in general; existence and uniqueness of a solution to this problem can be ensured using the so-called variational formulation, and the Sobolev space $H_0^1(\Omega)$.

3.4.1 Variational formulation

Definition 3.4.1. *The variational formulation of (**??**) consists in finding $u \in H_0^1(\Omega)$ such that*

$$(\,u'\,,\,v'\,) = (\,f, v\,), \quad \forall v \in H_0^1(\Omega). \tag{3.6}$$

Recall from the previous chapter that there exists a unique solution for the equation (**??**) according to Lax-Milgram Theorem.

Definition 3.4.2. *Let u be the solution of (**??**) , We will call σ the flux if*

$$\sigma = -u' \tag{3.7}$$

Theorem 3.4.1. *Let u be the solution of (**??**) and σ the flux given by (**??**). Then*

$$v \in H_0^1(\Omega), \quad \sigma \in H_0^1(\Omega)$$

3.4.2 The Finite elements method

Let us now introduce the finite element method for approximating the solution of (**??**). Let τ_h be a mesh of Ω, i.e., a division of the interval Ω into subintervals noted as K. Let $\mathbb{P}_k(K)$ stand for the set of polynomials of total degree less than or equal to k on the element K $\in \tau_h$. Let, finally,

$$V_h := \{v_h \in C^0(\Omega); v_h|_K \in \mathbb{P}_k(K), \forall K \in \tau_h; v_h|\partial\Omega = 0\} \tag{3.8}$$

Definition 3.4.3. *The formulation of the finite element method is deduced to : Find $u_h \in V_h$ Such that :*

$$(u_h', v_h') = (f, v_h), \forall v_h \in V_h \tag{3.9}$$

Recall that the existence and uniqueness of u_h follows by the same arguments as that of (**??**). In analogy with Definition , we introduce:

Definition 3.4.4. *Let u_h be the solution of (**??**). We will call $-u_h'$ the approximate flux .*

Theorem 3.4.2. *Let u_h be the solution of (**??**). Then :*

$$\begin{aligned} u_h &\in H_0^1(\Omega), \quad -u_h' \notin H_0^1(\Omega) \quad ingeneral \\ u_h &\in C^0(\Omega), \quad -u_h' \notin C^0(\Omega) \quad ingeneral \end{aligned}$$

3.4.3 Energy norm and dual norms

A primordial question for measuring the distance between u and u_h is the choice of the norm ($\|\cdot\|$ in (**??**) and (**??**). A prominent role between all different possibilities is played by the *energy norm*: this is the norm induced by the scalar product in (**??**) :

$$\|(u-u_h)'\| = \|(u-u_h)'\|_{L^2(\Omega)}. \tag{3.10}$$

This norm admits the following useful characterization:

Theorem 3.4.3. *[Energy norm for (**??**) as a dual norm] Let $v \in H_0^1(\Omega)$. Then*

$$\|v'\| = \sup_{\substack{\varphi \in H_0^1(\Omega) \\ \|\varphi'\|=1}} (v', \varphi'). \tag{3.11}$$

Proof. First remark that the proof in the case $v' = 0$ is trivial. Suppose that $v' \neq 0$. We will proceed in two steps.

1 Proof of (**??**) with the sign $\leq$.

By the properties of the $L^2(\Omega)$ scalar product, there holds

$$\|v'\|^2 = (v', v').$$

Thus

$$\|v'\| = \left(v', \frac{v'}{\|v'\|}\right).$$

Set $w\frac{v'}{\|v'\|}$ and remark that $w \in H_0^1(\Omega)$ and that $\|w'\| = 1$. Thus, passing to a supremum, we get

$$\|v'\| = (v', w') \leq \sup_{\substack{\varphi \in H_0^1(\Omega) \\ \|\varphi'\|=1}} (v', \varphi').$$

2 Proof of (**??**) with the sign $\geq$.
Using the Cauchy-Schwarz inequality, we can bound from above the supremum in (**??**), $\sup_{\substack{\varphi\in H_0^1(\Omega)\\ \|\varphi'\|=1}}(v',\varphi') \leq \sup_{\substack{\varphi\in H_0^1(\Omega)\\ \|\varphi'\|=1}}\{\|v'\|\|\varphi'\|\} = \|v'\|$. ■

3.5 A first a posteriori error estimate

With the notion of the flux reconstruction and of the characterization of the energy norm of Theorem (**??**) , we will now give our first a posteriori estimate on the error between u, the unknown solution of (**??**), and u_h, the known solution of (**??**). The last ingredient that we need is the Friedrichs inequality:

$$\|\varphi\| \leq \frac{h_\Omega}{\pi}\|\varphi'\| \quad \forall\varphi \in H_0^1(\Omega). \tag{3.12}$$

Theorem 3.5.1. *Let u be the weak solution given by (**??**). Let u_h be its finite element approximation . Let finally σ_h be a flux reconstruction . For any $K \in \mathcal{T}_h$, define the residual estimator by*

$$\eta_{R,K}\frac{h_\Omega}{\pi}\|f-\sigma_h'\|_K \tag{3.13}$$

and the diffusive flux estimator by

$$\eta_{DF,K}\|u_h'+\sigma_h\|_K. \tag{3.14}$$

Theorem 3.5.2. *Let u be the weak solution , Let u_h be its finite element approximation , Let finally σ_h be a flux reconstruction . For any $K \in \mathcal{T}_h$, define the residual estimator by*

$$\eta_{R,K}\frac{h_\Omega}{\pi}\|f-\sigma_h'\|_K \tag{3.15}$$

and the diffusive flux estimator by

$$\eta_{DF,K}\|u_h'+\sigma_h\|_K. \tag{3.16}$$

Then

$$\|(u-u_h)'\| \leq \left\{\sum_{K\in\mathcal{T}_h}\eta_{R,K}^2\right\}^{1/2} + \left\{\sum_{K\in\mathcal{T}_h}\eta_{DF,K}^2\right\}^{1/2}. \tag{3.17}$$

Proof. Recall Theorem (**??**) , where we set $vu-u_h$. Let $\varphi \in H_0^1(\Omega)$ with $\|\varphi'\| = 1$ be fixed. Using the characterization (**??**) of the weak solution, we have

$$((u-u_h)',\varphi') = (f,\varphi) - (u_h',\varphi').$$

Adding and subtracting (σ_h,φ') and using Green's theorem $(\sigma_h,\varphi') = -(\sigma_h',\varphi)$, we obtain

$$((u-u_h)',\varphi') = (f,\varphi) - (u_h',\varphi') + (\sigma_h,\varphi') - (\sigma_h,\varphi') = (f-\sigma_h',\varphi) - (u_h'+\sigma_h,\varphi').$$

We now bound these two terms separately.

- **Diffusive flux term:** By the Cauchy-Schwarz inequality,
$$(u_h' + \sigma_h, \varphi') \leq \|u_h' + \sigma_h\| \|\varphi'\| = \left\{ \sum_{K \in \mathcal{T}_h} \eta_{DF,K}^2 \right\}^{1/2},$$
where we used $\|\varphi'\| = 1$.
- **Residual term:** Applying the Cauchy-Schwarz inequality, Friedrichs inequality , and $\|\varphi'\| = 1$:
$$(f - \sigma_h', \varphi) \leq \|f - \sigma_h'\| \|\varphi\| \leq \|f - \sigma_h'\| \frac{h_\Omega}{\pi} \|\varphi'\| = \left\{ \sum_{K \in \mathcal{T}_h} \eta_{R,K}^2 \right\}^{1/2}.$$

Combining both bounds yields the desired result. ■

3.6 The Laplace equation in multiple space dimensions

Let us recall the Laplace equation : for $f \in L^2(\Omega)$, find u such that
$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{3.18}$$

As in The Previous Section , (**??**) does not have a classical solution (i.e., $u \in C^2(\Omega)$) in general. We are thus again led to the variational formulation.

3.6.1 Variational formulation

In order to define u, we set:

Definition 3.6.1. *The variational formulation of (**??**) consists in finding $u \in H_0^1(\Omega)$ such that*
$$(\nabla u, \nabla v) = (f, v) \quad \forall v \in H_0^1(\Omega). \tag{3.19}$$

Remark 3.6.1. *The existence and uniqueness of a solution of (**??**) is ensured by the Riesz representation theorem (or by the Lax–Milgram theorem).*

Definition 3.6.2 (Flux)**.** *Let u be the solution of (**??**). Set*
$$\sigma - \nabla u. \tag{3.20}$$
We will call σ the flux.

by using the definitions of the spaces $H_0^1(\Omega)$ and $H(\text{div}, \Omega)$, we have:

Theorem 3.6.1. *Let u be the solution of (**??**). Let σ be given by (**??**). Then $u \in H_0^1(\Omega)$, $\quad \sigma \in H(\text{div}, \Omega)$.*

Proof. The weak solution u belongs to $H_0^1(\Omega)$ by definition. To verify that $\sigma \in H(\mathrm{div}, \Omega)$, we check the three conditions of Weak divergence:

1. Condition 1 is satisfied since $u \in H_0^1(\Omega)$ implies $\sigma = -\nabla u$ is square-integrable.
2. For condition 2a, choose wf (with $f \in L^2(\Omega)$ by assumption).
3. Condition 2b follows from (**??**) and the inclusion $D(\Omega) \subset H_0^1(\Omega)$.

■

3.6.2 Approximate solution

In order to make the presentation general (not restricted to any particular numerical method), we assume that the approximate solution u_h satisfies:

$$u_h \in H^1(\mathcal{T}_h), \tag{3.21}$$

where $H^1(\mathcal{T}_h)$ is the broken Sobolev space .

Definition 3.6.3 (Approximate flux)**.** *Let u_h be the approximate solution of (**??**). We call*

$$\sigma_h - \nabla u_h \tag{3.22}$$

the approximate flux.

Remark 3.6.2. *Let u_h be the approximate solution (**??**). Then $u_h \notin H_0^1(\Omega)$, $-\nabla u_h \notin$* $\mathbf{H}(\mathrm{div}, \Omega)$ *in general.*

3.6.3 Energy (semi-)norm and dual norms

we measure the distance between u and u_h in the energy (semi-)norm induced by the scalar product in (**??**):

$$\|\nabla v\|, \quad v \in H^1(\mathcal{T}_h). \tag{3.23}$$

Theorem 3.6.2. *[Energy norm for (**??**) as a dual norm] Let $v \in H_0^1(\Omega)$. Then*

$$\|\nabla v\| = \sup_{\substack{\varphi \in H_0^1(\Omega) \\ \|\nabla \varphi\| = 1}} (\nabla v, \nabla \varphi). \tag{3.24}$$

3.6.4 A General a posteriori error estimate

We present an a posteriori error estimate for the distance between u (the unknown solution of (**??**) and u_h (the known approximate solution characterized by (**??**). This estimate provides a guaranteed upper bound and is applicable to any numerical method. [Potential and flux reconstructions for (**??**) We assume that s_h is a potential reconstruction and $\boldsymbol{\sigma}_h$ is an equilibrated flux reconstruction .

Theorem 3.6.3. *Let u be the weak solution and u_h satisfy (**??**). Under the previous Assumption , define for each $K \in \mathcal{T}_h$:*

$$\eta_{\mathrm{R},K} C_{\mathrm{P},K} h_K \|f - \nabla \cdot \boldsymbol{\sigma}_h\|_K, \tag{3.25}$$

$$\eta_{\mathrm{DF},K} \|\nabla u_h + \boldsymbol{\sigma}_h\|_K, \tag{3.26}$$

$$\eta_{\mathrm{NC},K} \|\nabla(u_h - s_h)\|_K. \tag{3.27}$$

Then:

$$\|\nabla(u - u_h)\|^2 \leq \sum_{K \in \mathcal{T}_h} (\eta_{\mathrm{R},K} + \eta_{\mathrm{DF},K})^2 + \sum_{K \in \mathcal{T}_h} \eta_{\mathrm{NC},K}^2. \tag{3.28}$$

Proof. Define $s \in H_0^1(\Omega)$ by:

$$(\nabla s, \nabla v) = (\nabla u_h, \nabla v) \quad \forall v \in H_0^1(\Omega). \tag{3.29}$$

Existence and uniqueness of s follow from the Riesz representation theorem, as $(\nabla u_h, \nabla \cdot)$ is a continuous linear form on $H_0^1(\Omega)$. This s is the orthogonal projection of u_h onto $H_0^1(\Omega)$ with respect to $(\nabla \cdot, \nabla \cdot)$, yielding:

$$\|\nabla(u - u_h)\|^2 = \|\nabla(u - s)\|^2 + \|\nabla(s - u_h)\|^2. \tag{3.30}$$

Moreover: $\|\nabla(s - u_h)\|^2 = \inf_{v \in H_0^1(\Omega)} \|\nabla(v - u_h)\|^2 \leq \|\nabla(s_h - u_h)\|^2 = \sum_{K \in \mathcal{T}_h} \eta_{\mathrm{NC},K}^2$. For the first term in (**??**), since $u - s \in H_0^1(\Omega)$, Theorem (**??**) gives:

$$\|\nabla(u - s)\| = \sup_{\substack{\varphi \in H_0^1(\Omega) \\ \|\nabla \varphi\| = 1}} (\nabla(u - u_h), \nabla \varphi).$$

■

Proof (continued). For fixed $\varphi \in H_0^1(\Omega)$ with $\|\nabla \varphi\| = 1$, the weak formulation **??** gives: $(\nabla(u - u_h), \nabla \varphi) = (f, \varphi) - (\nabla u_h, \nabla \varphi)$. Adding/subtracting $(\boldsymbol{\sigma}_h, \nabla \varphi)$ and applying Green's theorem :

$$(\nabla(u - u_h), \nabla \varphi) = (f - \nabla \cdot \boldsymbol{\sigma}_h, \varphi) - (\nabla u_h + \boldsymbol{\sigma}_h, \nabla \varphi). \tag{3.31}$$

- **Diffusive flux term:** By Cauchy-Schwarz,

$$-(\nabla u_h + \boldsymbol{\sigma}_h, \nabla \varphi) \leq \sum_{K \in \mathcal{T}_h} \|\nabla u_h + \boldsymbol{\sigma}_h\|_K \|\nabla \varphi\|_K = \sum_{K \in \mathcal{T}_h} \eta_{\mathrm{DF},K} \|\nabla \varphi\|_K.$$

- **Residual term:** Using Poincaré inequality , and $\varphi_K\frac{1}{|K|}\int_K\varphi$:

$$(f-\nabla\cdot\boldsymbol{\sigma}_h,\varphi)=\sum_{K\in\mathcal{T}_h}(f-\nabla\cdot\boldsymbol{\sigma}_h,\varphi-\varphi_K)_K\le\sum_{K\in\mathcal{T}_h}C_{\mathrm{P},K}h_K\|f-\nabla\cdot\boldsymbol{\sigma}_h\|_K\|\nabla\varphi\|_K=\sum_{K\in\mathcal{T}_h}\eta_{\mathrm{R},K}\|\nabla\varphi\|_K$$

Combining these in (**??**) and using $\|\nabla\varphi\|=1$: $\|\nabla(u-s)\|^2\le\left(\sum_{K\in\mathcal{T}_h}(\eta_{\mathrm{DF},K}+\eta_{\mathrm{R},K})\|\nabla\varphi\|_K\right)^2\le\sum_{K\in\mathcal{T}_h}(\eta_{\mathrm{DF},K}+\eta_{\mathrm{R},K})^2$. With (**??**) and the nonconformity bound, this proves (**??**). ■

3.7 Studying the Laplacien Model Case

in this section , we will use the previous Theorems and definitions to get the local estimator for this problem .

3.7.1 Encoding

- Here is Some symbols and their meanings that we'll face in this Section :

h_K = diameter of triangle K

h_T = length of side T

$\|v\|_{m,K}$ = Sobolev norm of v in $H^m(K)$

$|v|_{1,\beta,K}$ = Sobolev semi-norm of v in $W^{1,\beta}(K)$

$\|v\|_{1,\beta,K}$ = Sobolev norm of v in $W^{1,\beta}(K)$

$\|v\|$ = norm of v in the space where the problem is posed. For example, in §1:

$\|v\|=\|v\|_{1,\Omega}$

n = unit normal vector to the boundary, oriented outwards.

3.7.2 Position of the problem

- For a better comprehension we will start with this simple case :

$$\begin{cases}-\Delta u=f & ,in\quad\Omega\\ u=0 & ,in\quad\delta\Omega\end{cases}\tag{3.32}$$

- First , we have to find the **Variational form** of this problem , let $f\in L^2(\Omega)$, So by choosing a test function $v\in V=H_0^1(\Omega)$ and integrating the two sides , we find :

$$-\Delta u.v=f.v$$

$$\Rightarrow-\int_\Omega\Delta u.v=\int_\Omega f.v$$

- By using the integration by parts :

$$\Rightarrow \int_{\Omega} \Delta u.\Delta v dx + \int_{\delta\Omega} \frac{\delta u}{\delta x}.v ds = \int_{\Omega} f.v dx$$

- we have u=0 in $\delta\Omega$

$$\Rightarrow \int_{\delta\Omega} \frac{du}{dx}.v ds = 0.$$

- Suppose That :

$$\begin{cases} a(u,v) = \int_{\Omega} \Delta u.\Delta v dx \\ L(v) = \int_{\Omega} f.v dx \end{cases} \tag{3.33}$$

- We find the **Variational form** :
- find $u^* \in V$ Such that :

$$\begin{cases} a(u,v) = L(v) \qquad ,\forall v \in V = H_0^1 \end{cases}$$

- This Problem admits a unique solution according to **lAX-MILIGRAME** Theorem , we will use the **finits elements** method to approximate it .
- Let (T_h) be a regular triangulation of (Ω). We have in particular (with the classical notations): $\frac{h_K}{h_t} < \sigma \quad \forall K \in T_h, \quad \forall t \subset \partial K \quad (t \text{ side of } K)$
- With σ independent of $h = max(h_k, h)$ in T_h .

Definition 3.7.1. *Let V_h be a finite element subspace of V associated with T_h and π_h the interpolation operator of discontinuous functions , π_h satisfies (C being a constant independent of h):*

$$||v - \pi_h v||_{m,K} \leq C h_K^{1-m} \sum_{K' \in S_K} ||v||_{1,K'} \tag{3.34}$$

For all $v \in H^1(S_K)$ with $m = 0$ or 1 . Where : $S_K = \bigcup \{K', K \cap K' \neq \emptyset\}$

The approximate problem is then written:

Find $(u_h \in V_h)$ such that:

$$\int_{\Omega} \nabla u_h \cdot \nabla v_h \, dx = \int_{\Omega} f v_h \, dx \quad ,\forall v_h \in V_h. \tag{3.35}$$

3.7.3 Error Estimation :

- The residual is the continuous linear form on V defined by:

$$\langle R, v\rangle = \int_\Omega \nabla u_h \nabla v - \int_\Omega f v \quad \forall v \in V$$

- It is therefore an element of $H^{-1}(\Omega)$ (The Dual of the Space $H_0^1(\Omega)$).

- Since $< R, v >= a(u_h, v) - (f, v) = a(u_h - u, v)$ then by taking $v = u_h - u$ and using the coercivity and continuity of a :

$$\begin{aligned} < R, v >= a(u_h - u, v) &= a(u_h - u, u_h - u) \\ &\geq C||u_h - u||^2 \\ \Longrightarrow C||u_h - u||^2 &\leq < R, u - u_h > \\ \Longrightarrow ||u - u_h|| \leq \frac{< R, u - u_h >}{||u_h - u||} &\leq ||R||_* \end{aligned}$$

- where $||R||_*$ denotes the norm of R in $V' = H^-1(\Omega)$, So It suffices therefore to estimate $||R||_*$.

- We propose to estimate R in terms of the norm of its "restriction" to each triangle K. by using (**??**) for all $v \in V$:

$$-\Delta u_h = f \Longrightarrow \int_\Omega \nabla u_h \cdot \nabla v_h \, dx = \int_\Omega -\Delta u_h v \, dx$$

- Then by applying **Green's Formula** (**??**):

$$\langle R, v\rangle = \textstyle\sum_{K \in T_h} \int_K (\nabla u_h \nabla v - f v) = \sum_{K \in T_h} \left(\int_K (-\Delta u_h - f) v + \int_{\partial K} \frac{\partial u_h}{\partial n} v \right) .$$

- And since $< R, v_h >= 0$ for all $v_h \in V_h$, and by using the Interpolation Opertor (**??**) then we find :

$$\begin{aligned} \langle R, v\rangle &= \sum_K \left(\int_K (-\Delta u_h - f)(v - \pi_h v) + \int_{\partial K} \frac{\partial u_h}{\partial n} (v - \pi_h v) \right) \\ &= \sum_K \int_K (-\Delta u_h - f)(v - \pi_h v) + \sum_{t \in \Gamma_l} \int_t \left[\frac{\partial u_h}{\partial n} \right]_t (v - \pi_h v) \end{aligned}$$

- Where Γ_l is the union of all sides of triangles interior to Ω (sides not meeting the boundary Γ of Ω) and $[w]_l$ denotes the jump of w across the side l.

- We then have, by using **Hölder's inequality** (**??**) on each K:

$$\langle R, v\rangle \leq \sum_K |\Delta u_h + f|_{0,K} |v - \pi_h v|_{0,K} + \sum_{l \in \Gamma_l} \left| \left[\frac{\partial u_h}{\partial n} \right]_l \right|_{0,l} \|v - \pi_h v\|_{0,l}$$

- We can Use **Cauchy-Shwartz Inequality** to get :

$$\langle R, v\rangle \leq \left(\sum_K h_K^2 |\Delta u_h + f|_{0,K}^2\right)^{1/2} \left(\sum_K h_K^{-2} |v - \pi_h v|_{0,K}^2\right)^{1/2} + \left(\sum_{l\in\Gamma_l} h_t \left|\left[\frac{\partial u_h}{\partial n}\right]_l\right|_{0,t}^2\right)^{1/2} \left(\sum_{t\in\Gamma_i} h_t^{-1} \|v -$$

- By using The formula of The Interpolation Operator (**??**) we'll get :

$$\langle R, v\rangle \leq C \left(\sum_K h_K^2 |\Delta u_h + f|_{0,K}^2\right)^{1/2} \|v\|_{1,\Omega} + \left(\sum_{t\in\Gamma_t} h_t \left|\left[\frac{\partial u_h}{\partial n}\right]_l\right|_{0,t}^2\right)^{1/2} \left(\sum_{t\in\Gamma_l} h_t^{-1} \|v - \pi_h v\|_{0,t}^2\right)^{1/2}$$

- To estimate the Amount

$$\left(\sum_{t\in\Gamma_l} h_t^{-1} \|v - \pi_h v\|_{0,t}^2\right)^{1/2} \tag{3.36}$$

we introduce the next **Lemma** :

Lemma 3.7.1. *There exists a constant $C \geq 0$ such that for all $K \in T_h$ and for $v \in W^{1,\beta}(K)$ we have:*

$$h_K \|v\|_{0,\beta,\partial K} \leq C \left(\|v\|_{0,\beta,K} + h_K \|\nabla v\|_{1,\beta,K}\right). \tag{3.37}$$

The Amount (**??**) can be bounded using (**??**) :

$$\begin{aligned}
& \|v - \pi_h v\|_{2,K} \leq C h_K^{-1} \sum_{K\in T_h} \|v - \pi_h v\|_{0,K} \\
\Rightarrow \quad & h_K^{-1} \|v - \pi_h v\|_{2,K} \leq \sum_{K\in T_h} h_K^{-2} \|v - \pi_h v\|_{0,K} \\
\Rightarrow \quad & h_K^{-1} \|v - \pi_h v\|_{0,K} \leq \sum_{K\in T_h} h_K^{-2} \|v - \pi_h v\|_{0,K} \\
\Rightarrow \quad & h_K^{-1} \|v - \pi_h v\|_{0,\partial K} \leq \sum_{K\in T_h} h_K^{-2} \|v - \pi_h v\|_{0,K} \\
\Rightarrow \quad & \left(\sum_{K\in T_h} h_K^{-1} \|v - \pi_h v\|_{0,\partial K}^2\right)^{1/2} \leq \left(\sum_{K\in T_h} h_K^{-2} \|v - \pi_h v\|_{0,K}^2\right)^{1/2}
\end{aligned}$$

- Next , We Apply Lemma (**??**) for $\beta = 2$ we find :

$$\left(\sum_{K\in T_h} h_K^{-1}\|v-\pi_h v\|_{0,\partial K}^2\right)^{1/2} \leq C\left(\sum_{K\in T_h} h_K^2\left(\|v-\pi_h v\|_{0,K}^2 + h_K^2\|\nabla(v-\pi_h v)\|_{1,K}^2\right)\right)^{1/2}$$
$$\leq C\|v\|_{1,\Omega}$$

- Then becomes, by taking the supremum over $v \in V$:

$$\|R\|_* \leq C\left[\sum_{K\in T_h} h_K^2|\Delta u_h + f|_{0,K}^2\right]^{1/2} + C'\left[\sum_{t\in\Gamma_l} h_t\left|\left[\frac{\partial u_h}{\partial n}\right]_t\right|_{0,t}^2\right]^{1/2}$$

- and since $||u-u_h|| \leq C||R||_*$, we deduce that:

$$\|u-u_h\| \leq C\left[\sum_{K\in T_h} h_K^2|\Delta u_h + f|_{0,K}^2\right]^{1/2} + C'\left[\sum_{t\in\Gamma_t} h_t\left|\left[\frac{\partial u_h}{\partial n}\right]_t\right|_{0,l}^2\right]^{1/2}$$

- as $a^{1/2} + b^{1/2} \leq 2^{1/2}(a+b)^{1/2}$ for a and $b \geq 0$, then:

$$\|u-u_h\| \leq C\left[\sum_{K\in T_h}\left\{h_K^2|\Delta u_h + f|_{0,K}^2 + \sum_{t\in\partial K} h_t\left|\left[\frac{\partial u_h}{\partial n}\right]_t\right|_{0,t}^2\right\}\right]^{1/2}$$

- and thus:

Theorem 3.7.2. *There exists a constant* C *independent of* h *such that:*

$$\||u-u_h\| \leq C\left[\sum_{K\in T_h}\eta(K)^2\right]^{1/2}$$

- *Where :*

$$\eta(K) = \left\{h_K^2|\Delta u_h + f|_{0,K}^2 + \sum_{l\in\partial K} h_t\left|\left[\frac{\partial u_h}{\partial n}\right]_t\right|_{0,t}^2\right\}^{1/2} \tag{3.38}$$

Remark 3.7.1. 1 *For any side* l *contained in the boundary of* Ω*, we set by convention the jump across* l *to be 0.*

2 *The mapping* $\eta : T_h \to \mathbb{R}$ *is called the local error estimator .*

Chapter 4

Application On FreeFem++

4.1 Introduction

A partial differential equation is a relation between a function of several variables and its (partial) derivatives . Many problems in physics, engineering, mathematics and even banking are modeled by one or several partial differential equations.
Freefem is a software to solve these equations numerically. As its name says, it is a free software (see copyright for full detail) based on the Finite Element Method. This software runs on all unix OS (with g++ 2.95.2 or better and X11R6) , on Window95, 98, 2000, NT, XP, on MacOS 9 and X.
Many phenomena involve several different fields. Fluid-structure interactions, Lorenz forces in liquid aluminium and ocean-atmosphere problems are three such systems. These require approximations of different degrees, possibly on different meshes. Some algorithms such as Schwarz' domain decomposition method also require data interpolation on different meshes within one program. freefem++ can handle these difficulties, i.e. arbitrary finite element spaces on arbitrary unstructured and adapted meshes

The characteristics of freefem++ are:

- A large variety of finites elements : linear and quadratic Lagrangian elements, discontinuous P1 and Raviart-Thomas elements, elements of a non-scalar type, mini-element, ...

- Automatic interpolation of data on different meshes to an over mesh, store the interpolation matrix.

- Linear problems description (real or complex) thanks to a formal variational form, with access to the vectors and the matrix if needed.

- Includes tools to define discontinuous Galerkin formulations (please refer to the following keywords: “jump”, “average”, “intalledges”)

- Analytic description of boundaries. When two boundaries intersect, the user must specify the intersection points.

- Automatic mesh generator, based on the Delaunay-Voronoï algorithm. Inner points density is proportional to the density of points on the boundary.

- Metric- based anisotropic mesh adaptation. The metric can be computed automatically from the Hessien of a solution .

- Solvers : LU, Cholesky, Crout, CG, GMRES, UMFPACK linear solver, eigenvalue and eigenvector computation.

- Online graphics, C++-like syntax.

- Many examples: Navier- Stokes, elasticity, Fluid structure, Schwarz’s domain decomposition method, Eigen value problem, residual error indicator, ...

4.1.1 History

The project has evolved from MacFem, PCfem, written in Pascal. The first C version was freefem 3.4; it offered mesh adaptation on a single mesh. A thorough rewriting in C++ led to freefem+ 1.2.10, which also included interpolation over multiple meshes (functions defined on one mesh can be used on any other mesh). Implementing the interpolation from one unstructured mesh to another was not easy because it had to be fast and nondiffusive; for each point, one had to find the containing triangle. This is one of the basic problems of computational geometry (see Preparata & Shamos[13] for example). Doing it in a minimum number of operations was a challenge. Our implementation was O(n log n) and based on a quadtree. We are now introducing freefem++ , an entirely new program written in C++ and based on bison for a more adaptable freefem language.
The freefem language allows for a quick specification of any partial differential system of equations. The language syntax of freefem++ is the result of a new design which makes use of the STL [21], templates and bison for its implementation. The outcome is a versatile software in which any new finite element can be included in a few hours; but a recompilation is then necessary. The library of finite elements available in freefem++ will therefore grow with the

version number. So far we have linear and quadratic Lagrangian elements, discontinuous P1 and Raviart-Thomas elements

4.1.2 Getting Started

We explain how freefem++ solve the problem Poisson; For a given function f(x, y), find a function u(x, y) satisfying :

$$\Delta u(x,y) = f(x,y) \quad \text{if}(x,y) \quad \text{is in}\Omega, \Delta u = \frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y} \tag{4.1}$$

$$u(x,y) = 0 \quad \text{if}(x,y) \quad \text{is on} \quad \delta\Omega \tag{4.2}$$

The following example shows **freefem++** program solving u when $f(x,y) = xy$ (see 5th line) and Ω is the unit disk. The boundary $C = \partial\Omega$ is:

$$C = \left\{(x,y)|x = cos(t), y = sin(t), 0 \leq t \leq 2\pi\right\}, (seeline1) \tag{4.3}$$

The example shows **freefem++** covers easily all standard step in FEM (finite element method). We explain how they are done by **freefem++** in a step-by-step manner.

Listing 4.1: FreeFem++ example for solving Poisson equation

```
// Example 1 - Poisson equation on a unit circle

1: border C(t=0,2*pi){x=cos(t); y=sin(t);label=1;}
2: mesh Th = buildmesh (C(50));
3: fespace Vh(Th,P2);
4: Vh u,v;
5: func f= x*y;
6: problem Poisson(u,v,solver=LU) =
7:     int2d(Th)(dx(u)*dx(v) + dy(u)*dy(v))    // bilinear part
8:   - int2d(Th)( f*v)                         // right hand side
9:   + on(1,u=0)                               // Dirichlet boundary ...
    condition
10:
11: real cpu=clock();
12: Poisson;                                   // SOLVE THE PDE
13: plot(u);                                   // Visualize solution
14: cout << " CPU time = " << clock()-cpu << endl;
```

This example demonstrates:

- Line 1: Definition of a circular boundary with label 1
- Line 2: Mesh generation with 50 points on the boundary
- Line 3: Creation of a P2 finite element space
- Lines 4-5: Declaration of variables and forcing function
- Lines 6-9: Definition of the Poisson problem with:
  - Bilinear form (weak Laplacian)
  - Linear form (right-hand side)
  - Dirichlet boundary condition (u=0 on boundary 1)
- Lines 11-14: Solution process with timing and visualization

1 Mesh Generation

1st line: the boundary Γ are described analytically (by opposition to CSG) as stated before. In the case $\Gamma = \sum_{j=0}^{J} \Gamma_j$ with curves Γ_j
then the user must specify the intersection points
In case two boundaries intersect. By the use of the keyword `label` such as:

```
border Gamma1(t=a1,b1) { x=...; y=... ; label=1; }
:     :     :
border GammaJ(t=aj,bJ) { x=...; y=... ; label=1; }
```

the user can refer to Γ by the number “1” .
2nd line: the triangulation T_h of Ω is automatically generated by `buildmesh(C(50))` giving 50 points on C ,nAutomatic mesh generation is based on the Delaunay-Voronoi algorithm. Refinement of the mesh are done by increasing the points on Γ, for example, `buildmesh(C(100))`, because inner vertices are determined by the density of points on the boundary.

The symbol T_h (`Th` in `freefem++`) shows a family $\{T_k\}_{k=1,\cdots,n_t}$ of triangles, with the size h of the mesh. Here n_t stands for the number of triangles in T_h. If Ω is not polygonal domain, a “skin” remains between the exact domain Ω and its approximation $\Omega_h = \cup_{k=1}^{n_t} T_k$. However, we notice that all corners of $\Gamma_h = \partial\Omega_h$ are on Γ.

2 Making Finite Element Space

3rd line: `fespace Vh(Th,P2)` makes the continuous Finite Element SPACE $V_h(T_h, P_2) = \left\{ w(x,y) \middle| w(x,y) = \sum_{k=0}^{M-1} w_k\phi_k(x,y), w_k \text{ are real numbers} \right\}$
where P_2 indicate ϕ_k is a polynomial of degree ≤ 2, that is, in each T_k, $\phi_k(x,y) =$

$\alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2$ and the constants $\alpha_1, \cdots, \alpha_6$ are defined by its values at the vertices of T_k and their middle points that continuous in Ω. Here w_k are called the degree of freedom of w and M the number of the degree of freedom. Already `freefem++` implemented P0, P1, P2, RT0, P1nc, P1dc, P2dc, P1b, P2b elements. The user can easily add a part of arbitrary degree elements to `freefem++`, so the available finite elements will differ with the version.

3 Setting the Problem

4th line: `Vh u` declare that u is approximated through the use of the basis functions ϕ_k in V_h, that is, $u(x,y) \simeq u_h(x,y) = \sum_{k=0}^{M-1} u_k \phi_k(x,y)$
5th line: the given function f is defined analytically using the keyword `func`.

6th–9th lines: the formulation of (**??**) and (**??**) are done. Multiplying (**??**) by $v(x,y)$ and integrating the result over Ω, we have $-\int_\Omega v\Delta u \, dxdy = \int_\Omega vf \, dxdy$
Then, by Green's formula, the problem Poisson is translated into finding u such that

$$a(u,v) - \ell(f,v) = 0 \quad \text{for all } v \tag{4.4}$$

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dxdy \tag{4.5}$$

$$\ell(f,v) = \int_\Omega fv \, dxdy \tag{4.6}$$

satisfying $v = 0$ on $\partial\Omega$. In **freefem++** the problem Poisson is declared by

$\forall h \, u, v$; problem Poisson$(u,v) =$

and (**??**) is expressed by symbols $\mathbf{dx}(u) = \partial u/\partial x$, $\mathbf{dy}(u) = \partial u/\partial y$ and
$\int_\Omega \nabla u \cdot \nabla v \, dxdy \longrightarrow \mathrm{int2d(Th)}(\mathbf{dx}(u) * \mathbf{dx}(v) + \mathbf{dy}(u) * \mathbf{dy}(v))$

$\int_\Omega fv \, dxdy \longrightarrow \mathrm{int2d(Th)}(\mathbf{f} * \mathbf{v})$ (notice here, u is unused)
In **freefem++** the first and second formulas just above must be distinguished each other. Because, the linear system to be solved are created from substituting u_h for u and ϕ_i for v in (**??**) ,

$$A_{ij} u_j - F_i = 0 \quad i,j = 0, \cdots, M-1; \quad F_i = \int_\Omega f \phi_i \, dxdy \tag{4.7}$$

and the solution $u_h = \sum_{j=0}^{M-1} u_j \phi_j$ must satisfy "$u_h = 0$ on $\Gamma_h \simeq C^n$". The matrix $A = (A_{ij})$ is called *stiffness matrix* and is modified from

$$A^0 = \{A^0_{ij}\}, \, A^0_{ij} = \int_\Omega \nabla \phi_j \cdot \nabla \phi_i \, dxdy \quad i,j = 0, \cdots, M-1 \tag{4.8}$$

to ensure $u = 0$ on C by "+on$\{1, u = 0\}$" in 9th line. If you want use the symbol "C" such as "+on$\{C, u = 0\}$" (9th line), then the user do not use the keyword "label". We can create directly the stiffness matrix A in (**??**) by

```
varf a(u,v) = int2d(Th) (dx(u)*dx(v) + dy(u)*dy(v)) + on(C,u=0);
matrix A=a(Vh, Vh);  // stiffness matrix
```

The vector B in (**??**) is called *load matrix*, and also we get it:

```
varf b([v], [f]) = int2d(Th) (v*f);
matrix B=b(Vh, Vh);
```

The linear system (**??**) is solved by a Gauss LU factorisation. You can declare the solver of (**??**), for example,

```
Vh u,v; problem Poisson(u,v, solver=CG) =
```

means that (**??**) will be solved by Conjugate Gradient method.

4 Solving and Visualization

11th line: the current time is stored into the real-valued variable `cpu`.

12th line: the problem is solved by calling its name.

4.2 Features of FreeFem++

The language it defines is typed, polymorphic and reentrant with macro generation , Every variable must be typed and declared in a statement; each statement separated from the next by a semicolon ';'.

For purposes of explanation, we used T_h (`Th`), V_h (`Vh`), unknown function u (`u`), test functions v (`v`) and the problem `Poisson`, etc. (the term inside the parentheses are symbols in FreeFem++ programming), but you can use any name except reserved words and names already used. Reserved words are shown in blue. `pi`, `x`, `y`, `label`, `solver` are reserved variables. It is allowed (although not advisable) to redefine these variables, so they will not be highlighted again in the following example programs.

In each step, the independence in FreeFem++ programming is very high as stated below.

- For example, by changing 1st and 2nd lines as following, we can solve (**??**) and (**??**) in L-shape domain with $\Gamma = \sum_{j=1}^{6} \Gamma_j$.

```
border G1(t=0,1){x=t;y=0;label=1};     // $\Gamma_1$
border G2(t=0,0.5){x=1;y=t;label=1};    // $\Gamma_2$
border G3(t=0,0.5){x=1-t;y=0.5;label=1}; // $\Gamma_3$
border G4(t=0.5,1){x=0.5;y=t;label=1};  // $\Gamma_4$
border G5(t=0.5,1){x=1-t;y=1;label=1};  // $\Gamma_5$
border G6(t=0,1){x=0;y=1-t;label=1};    // $\Gamma_6$
mesh Th = buildmesh(G1(6)+G2(4)+G3(4)+G4(4)+G5(4)+G6(6));
```

- In Step 3, you can control where the solution will be approximated. If you write “`Vh(Th,P1)`” in 3rd line, you can get P_1-approximation. The machine time by P_1-element will be faster than P_2-element and the storage is less.
- In Step 4, you can change the equation and boundary conditions easily. For example, if you want solve

$$-\mathrm{div}(k(x,y)\nabla u(x,y)) = f(x,y) \quad \text{in} \quad \Omega$$

$$u(x,y) = 0 \quad \text{if} \quad (x,y) \quad \text{on} \quad \Gamma$$

you only write as follows:

```
func f= ...;
func k= 1+sin(2*pi*x)*cos(2*pi*y);
problem Poisson(u,v) =
int2d(Th)(k*(dx(u)*dx(v) + dy(u)*dy(v)))
- int2d(Th)(f*v)
+ on(1,u=0);
```

The user can use FE-function as the given function f , for example, obtained function u in Example 1. In FreeFem++ programming, the easy reuse of the obtained results is important feature.

The user can easily compare between mathematical modelling and FreeFem++ program.

4.2.1 Projection or Interpolation

For a finite element space V_h, the P_1-projection Π_h is defined by $\Pi_h f = f(q^1)\phi_1 + f(q^2)\phi_2 + \cdots + f(q^{n_v})\phi_{n_v}$ for all continuous functions f. In **FreeFem++** we can easily create the projection f_h (`fh`) by

```
Vh fh = f(x, y);
```

Π_h is also called P_1-interpolate.

4.2.2 Matrix and Vector

Here, we show how to get the stiffness matrix using freefem++. The first command varf is to define the *variational formula.*

Example 4.2.1. *Here we solve the same problem (1.1) and (1.2) using matrix. For purposes of explanation, we chose mesh size and use P_1-element:*

```
border C(t=0,2*pi){ x=cos(t); y=sin(t); }
mesh Th = buildmesh(C(7)); % changed from Example 1
fespace Vh(Th,P1);
Vh u,v,f,F;
varf a(u,v) = int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v))
+on(C,u=0); % see (1.7)
varf b([v],[f]) = int2d(Th)(v*f);
f = x*y; % interpolate (x,y) ⟵ x ∗ y function
matrix A = a(Vh,Vh); % stiffness matrix, see (1.6)
matrix B = b(Vh,Vh);
F[] = B*f[]; % load vector, see (1.6)
cout << "F=" << F[] << endl;
cout << "A=" << A << endl;
u[] = A^-1*F[]; % solve AU_h = F, see (1.6)
plot(u);
```

We get the mesh $\mathcal{T}_h = \{T_1, \cdots, T_7\}$ (see Fig. 1.3).

In what follows, we denote the vertices by q^i, $i = 1, \cdots, 8$, the number of vertices by n_v, the number of triangles by n_t. For each triangle $T_k \in \mathcal{T}_h$, we index the vertices by q^{k_1}, q^{k_2}, q^{k_3} and denote the edges by $[q^{k_1}, q^{k_2}]$, $[q^{k_2}, q^{k_3}]$, $[q^{k_3}, q^{k_1}]$, that is, $[q^i, q^j]$ is the segment connecting q^i and q^j. We denote the number of edges $[q^i, q^j]$ by n_e for all q^i, $q^j \in \partial\Omega_h$, $\Omega_h = \sum_{k=1}^{7} T_k$. Here $n_v = 8$, $n_t = 7$, $n_e = 7$.

The function v in "Vh" is expressed

$$v(x) = v_1\varphi_1(x) + \cdots + v_{n_v}\varphi_{n_v}(x)$$

using the hat functions φ_j, $j = 1, \cdots, n_v$ (see Fig. 1.4). Here the j-th hat function φ_j associated with j-th vertex q^j is defined in the following way: [1]note

a φ_j is continuous function on Ω_h.

b φ_j is linear on each triangle T_k, $k = 1, \cdots, n_t$ of "Th".

c $\varphi_j(q^i) = \delta_{ji}$ where q^i denotes the i-th vertex, for all $i = 1, \cdots, n_v$.

Here δ_{ij} is the Kronecker symbol.

Other finite element spaces in freefem++ are explained in Section 4. [2]note

For an element $v = v_1\phi_1 + \cdots + v_M\phi_M$ in a finite element space V_h, we get the column vector $\{v\}$

$$\{v\} = \begin{bmatrix} v_1 \\ \vdots \\ v_M \end{bmatrix} \qquad \{v\} = v[] \quad \text{in freefem++}$$

Theoretically, it is natural to use the finite element space

$$H^1_{0h} = \{v \in V_h(\mathcal{T}_h, P_1) \mid \phi_i(x) = 0 \quad \text{if } q^i \in \partial\Omega_h\}$$

Let I_Ω be the set of indices i of all internal vertices of the mesh Vh. In this example, $I_\Omega = \{6\}$. The stiffness matrix A in 10th line is:

$$\left[\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 10^{30} & -0.31 & 0 & 0 & -0.46 & -0.51 & 0 & 0 \\ -0.31 & 10^{30} & -0.23 & 0 & 0 & -0.71 & 0 & 0 \\ 0 & -0.23 & 10^{30} & -0.31 & 0 & -0.71 & 0 & 0 \\ 0 & 0 & -0.31 & 10^{30} & 0 & -0.51 & -0.46 & 0 \\ -0.46 & 0 & 0 & 0 & 10^{30} & -0.35 & 0 & -0.54 \\ -0.51 & -0.71 & -0.71 & -0.51 & -0.35 & 3.47 & -0.35 & -0.30 \\ 0 & 0 & 0 & -0.46 & 0 & -0.35 & 10^{30} & -0.54 \\ 0 & 0 & 0 & 0 & -0.54 & -0.30 & -0.54 & 10^{30} \end{array}\right] \tag{1.8}$$

[3]note that is

$$A_{ij} = \int_{\Omega_h} \nabla\varphi_j \cdot \nabla\varphi_i \quad \text{if } i \neq j,\ i = j \in I_\Omega \tag{1.9}$$

$$A_{ij} = E \quad (E = 10^{30}) \quad \text{if } j \notin I_\Omega. \tag{1.10}$$

The load vector F^T is:

$$\begin{pmatrix} -0.020 & -0.037 & 0.037 & 0.020 & 0.064 & 0 & -0.064 & 1. \times 10^{-17} \end{pmatrix}$$

For $i \notin I_\Omega$,

$$Eu_i + \sum_{i \neq j} A_{ij} u_j = b_i$$

which means that

$$u_i = (b_i - \sum_{i \neq j} A_{ij} u_j) \times E^{-1} \simeq 10^{-30} \simeq 0$$

Mathematical results indicate that the Poisson equation with Neumann boundary condition has no unique solution, whose weak form is the same as (1.1) except the boundary condition:

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v \, dx$$

without penalization E. Then the stiffness matrix is created by

```
varf a(u, v) = int2d(Th)(dx(u)*dx(v) + dy(u)*dy(v))   // stiffness matrix
```

and the obtained stiffness matrix is:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1.29 & -0.31 & 0 & 0 & -0.46 & -0.51 & 0 & 0 \\ -0.31 & 1.26 & -0.23 & 0 & 0 & -0.71 & 0 & 0 \\ 0 & -0.23 & 1.26 & -0.31 & 0 & -0.71 & 0 & 0 \\ 0 & 0 & -0.31 & 1.29 & 0 & -0.51 & -0.46 & 0 \\ -0.46 & 0 & 0 & 0 & 1.35 & -0.35 & 0 & -0.54 \\ -0.51 & -0.71 & -0.71 & -0.51 & -0.35 & 3.47 & -0.35 & -0.30 \\ 0 & 0 & 0 & -0.46 & 0 & -0.35 & 1.35 & -0.54 \\ 0 & 0 & 0 & 0 & -0.54 & -0.30 & -0.54 & 1.38 \end{bmatrix}$$

The determinant of this matrix is $-1.7082274230870981 \times 10^{-9} \approx 0$ (The matrix here differs from the original one by omitting digits beyond the third decimal point).

4.2.3 Matrix Operations

The multiplicative operators `*`, `/`, and `%` group left to right.

- `'` is unary right transposition of array or matrix
- `.*` is the term-by-term multiply operator
- `./` is the term-by-term divide operator

There are some compound operators:

- `^-1` is for solving the linear system (example: $b = A^{-1}x$)

- `'*` is the composition of transposition and matrix product, so it computes the dot product (example: `real DotProduct = a'*b`)

Example 4.2.2.

```
mesh Th = square(2,1);
fespace Vh(Th,P1);
Vh f,g;
f = x*y;
g = sin(pi*x);
Vh<complex> ff,gg;    // complex valued finite element function
ff = x*(y+1i);
gg = exp(pi*x*1i);
varf mat(u,v) =
int2d(Th)(1*dx(u)*dx(v)+2*dx(u)*dy(v)+3*dy(u)*dx(v)+4*dy(u)*dy(v))
+ on(1,2,3,4,u=1);
varf mati(u,v) =
int2d(Th)(1*dx(u)*dx(v)+2i*dx(u)*dy(v)+3*dy(u)*dx(v)+4*dy(u)*dy(v))
+ on(1,2,3,4,u=1);
matrix A = mat(Vh,Vh);
matrix<complex> AA = mati(Vh,Vh);    // complex sparse matrix

Vh m0; m0[] = A*f[];
Vh m01; m01[] = A'*f[];
Vh m1; m1[] = f[].*g[];
Vh m2; m2[] = f[]./g[];
cout << "f = " << f[] << endl;
cout << "g = " << g[] << endl;
cout << "A = " << A << endl;
cout << "m0 = " << m0[] << endl;
cout << "m01 = " << m01[] << endl;
cout << "m1 = "<< m1[] << endl;
cout << "m2 = "<< m2[] << endl;
cout << "dot Product = "<< f[]'*g[] << endl;
cout << "hermitian Product = "<< ff[]'*gg[] << endl;
```

This produces the following output:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & \\ 10^{30} & 1.5 & 0 & 3.0 & 4 - 2.5 & 0 & \\ 0.0 & 10^{30} & 0.5 & 0 & 0.5 & -2.5 & \\ 0 & 0.0 & 10^{30} & 0 & 0 & 0.5 & \\ 4 & 0.5 & 0 & 0 & 10^{30} & 0.0 & \\ 5 & -2.5 & 0.5 & 0 & 0.5 & 10^{30} & 0.0 \\ 6 & 0 & -2.5 & 0.0 & 0 & 0.5 & 10^{30} \end{bmatrix}$$

$$\{v\} = f[] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 1 \end{pmatrix}^T$$

$$\{w\} = g[] = \begin{pmatrix} 0 \\ 1 \\ 1.2 \times 10^{-16} \\ 0 \\ 1 \\ 1.2 \times 10^{-16} \end{pmatrix}^T$$

$$A * f[] = \begin{pmatrix} -1.25 \\ -2.25 \\ 0.5 \\ 0 \\ 5\times 10^{29} \\ 10^{30} \end{pmatrix}^T \qquad (= A\{v\})$$

$$A' * f[] = \begin{pmatrix} -1.25 \\ -2.25 \\ 0 \\ 0.25 \\ 5\times 10^{29} \\ 10^{30} \end{pmatrix}^T \qquad (= A^T\{v\})$$

$$f[].*g[] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 1.2\times 10^{-16} \end{pmatrix}^T \qquad = (v_1 w_1 \cdots v_M w_M)^T$$

$$f[]./g[] = \begin{pmatrix} NaN \\ 0 \\ 0 \\ NaN \\ 0.5 \\ 8.1\times 10^{15} \end{pmatrix}^T \qquad = (v_1/w_1 \cdots v_M/w_M)^T$$

$$f[]' * g[] = 0.5 \quad (= \{v\}^T\{w\} = \sum_{i=1}^{M} v_i w_i)$$

4.2.4 Data Types

Basically **FreeFem++** is a compiler, the language is typed, polymorphic and reentrant. Every variable must be typed, declared in a statement; each statement separated from the next by a semicolon ‘;’. The language allows the manipulation of basic types:

- Integers (`int`)
- Reals (`real`)
- Strings (`string`)
- Arrays (example: `real[int]`)

- Bidimensional (2D) finite element meshes (`mesh`)
- 2D finite element spaces (`fespace`)
- Definition of functions (`func`)
- Arrays of finite element functions (`func[basic_type]`)
- Linear and bilinear operators
- Sparse matrices
- Vectors, etc.

For instance:

```
int i, n = 20;       // i, n are integers
real[int] xx(n), yy(n);  // two arrays of size n
for (i = 0; i ≤ 20; i++) {  // which can be used in statements ...
    such as
xx[i] = cos(i*pi/10);
yy[i] = sin(i*pi/10);
}
```

The life of a variable is the current block `{...}`, except the `fespace` variable, and the variables local to a block are destroyed at the end of the block as follows.

Example 2

```
real r = 0.01;
mesh Th = square(10,10);      // unit square mesh
fespace Vh(Th,P1);            // P1 Lagrange finite element space
Vh u = x + exp(y);
func f = z * x + r * log(y);
plot(u, wait=true);
{                             // new block
  real r = 2;                 // not the same r
  fespace Vh(Th,P1);          // error because Vh is a global name
}                             // end of block
// here r back to 0.01
```

The type declarations are compulsory in **FreeFem++** because it is easy to make bugs in a language with many types. The variable name is just an alphanumeric string, the underscore character "_" is not allowed, because it will be used as an operator in the future.

4.2.5 List of Major Types

`bool`	Used for logical expressions and flow-control.
`int`	Declares an integer.
`string`	Declares a variable to store text enclosed within double quotes, such as: `"This is a string in double quotes."`
`real`	Declares a variable to store a floating-point number such as `12.345`.
`complex`	Complex numbers, such as $1 + 2i$, $i = \sqrt{-1}$.
`ofstream`	Creates an output file and its associated functions.
`ifstream`	Creates an input file and its associated functions.
`real[int]`	Declares an array of real numbers with integer indices.
`real[string]`	Declares an associative array of real numbers with string keys.
`string[string]`	Declares an associative array of strings with string keys.
`func`	Defines a function. When no arguments are specified, the independent variables are `x, y`.
`mesh`	Creates a triangulation .
`fespace`	Defines a finite element space.

Complex Number Example

```
complex a = 1i, b = 2 + 3i;
cout << "a + b = " << a + b << endl;
cout << "a - b = " << a - b << endl;
cout << "a * b = " << a * b << endl;
cout << "a / b = " << a / b << endl;
```

Output:

```
a + b = (2,4)
a - b = (-2,-2)
a * b = (-3,2)
a / b = (0.230769,0.153846)
```

Array Example

```
real[int] a(5);
a[0] = 1; a[1] = 2; a[2] = 3.3333333; a[3] = 4; a[4] = 5;
cout << "a = " << a << endl;
```

Output:

```
a = 5
1    2    3.33333    4    5
```

Function Example

```
func f = cos(x) + sin(y);
```

Remark: The function's type is determined by the expression's type. FreeFem++ supports mathematical operations such as `x^1` and `y^0.23`.

4.2.6 Problem Declarations and Global Variables

Problem Declaration Keywords

`problem`	Declares the weak form of a partial differential problem without solving it.
`solve`	Declares a problem and immediately solves it.
`varf`	Defines a full variational form.
`matrix`	Defines a sparse matrix.

Global Variables

These special variables link the language to finite element tools:

`x`	x-coordinate of current point (real value)
`y`	y-coordinate of current point (real value)
`z`	Reserved for future z-coordinate (real value)
`label`	Boundary label number (0 if not on boundary, int value)
`region`	Region number of current point (x, y) (int value)
`P`	Current point ($\mathbb{R}^2$ value). Access components with `P.x`, `P.y`
`N`	Outward unit normal vector at current boundary point ($\mathbb{R}^3$ value). Components: `N.x`, `N.y`
`lenEdge`	Length of current edge: $\|q^i - q^j\|$ for edge $[q^i, q^j]$
`hTriangle`	Size of current triangle
`nuTriangle`	Index of current triangle (integer)
`nuEdge`	Index of current edge in triangle (integer)
`nTonEdge`	Number of adjacent triangles to current edge (integer)
`area`	Area of current triangle (real)
`cout`	Standard output device (default: console). Limited functionality on Windows.
`cin`	Standard input device (default: keyboard). Currently non-functional on Windows.

Example Usage

```
// Accessing point coordinates and normal vectors
border C(t=0,2*pi){x=cos(t); y=sin(t); label=1;}
mesh Th = buildmesh(C(50));
fespace Vh(Th,P1);

// Using global variables in a variational form
varf a(u,v) = int1d(Th)((N.x*u + N.y*v)*lenEdge)
+ int2d(Th)((x+y)*u*v*area);

// Using region numbers
func f = (region == 1) ? x^2 : y^2;
```

Important Notes:

- The `z` component and `P.z`, `N.z` are reserved for future 3D support
- `cin` may not work as expected on all platforms
- Global variables are automatically available in all expressions evaluated on the mesh

4.2.7 Loops

FreeFem++ implements `for` and `while` loops with `break` and `continue` keywords.

For Loops

The `for` loop has three components:

[leftmargin=*]**Initialization**: Control variable setup **Condition**: Continuation condition (loop continues while true) **Change**: Control variable modification

```
for (INITIALIZATION; CONDITION; CHANGE) {
// BLOCK of calculations
}
```

Example: Sum from 1 to 10

```
int sum = 0;
for (int i = 1; i ≤ 10; i++)
sum += i;
```

While Loops

The `while` loop executes while its condition is true:

```
while (CONDITION) {
// BLOCK of calculations or change of control variables
}
```

Example: Sum from 1 to 10 using while

```
int i = 1, sum = 0;
while (i ≤ 10) {
sum += i;
i++;
}
```

Loop Control Statements

[leftmargin=*]`break`: Immediately exits the loop `continue`: Skips remaining code in current iteration

Example 3

```
// Count from 0 to 9
for (int i = 0; i < 10; i = i + 1)
cout << i << "\n";

// Halving loop with break condition
real eps = 1;
while (eps > 1e-5) {
eps = eps / 2;
if (i++ < 100) break;  // Safety break after 100 iterations
cout << eps << endl;
}

// Continue example
for (int j = 0; j < 20; j++) {
if (j < 10) continue;  // Skip first 10 iterations
// Code here executes only for j ≥ 10
cout << "Processing j = " << j << endl;
}
```

Important Notes:

- Loop variables declared in the initialization are local to the loop
- Floating-point numbers in conditions may lead to precision issues
- Infinite loops can be created with `while (true)`
- Nested loops are fully supported

4.2.8 Special Constants and System Commands

Predefined Constants

`end1`	End of line character for input/output devices
`true`	Boolean true value
`false`	Boolean false value
`pi`	Real approximation of π ($\approx 3.141592653589793$)

System Inspection

To display all types, operators, and functions:

```
dumptable(cout);  // Outputs complete language reference
```

System Command Execution

Execute shell commands (not implemented on Carbon MacOS):

```
exec("shell command");
```

Windows-specific Notes:

- Requires full path to executables
- Backslashes in paths must be escaped

Example for Windows:

```
exec("c:\\cygwin\\bin\\ls.exe");  // Note double backslashes
```

Example Usage

```
// Using constants
real circumference = 2*pi*10;
bool flag = true;

// Conditional based on boolean
if (flag) {
cout << "pi = " << pi << endl;
}

// System command to list files (Unix-like systems)
exec("ls -l");

// Windows alternative
exec("dir");
```

Important Notes:

- `endl` is used for line endings in output streams
- `pi` provides double-precision approximation of π
- System command execution may be platform-dependent
- Security considerations apply when using `exec()` with user input

4.2.9 Input/Output

FreeFem++ uses C++-style input/output syntax with `cout`, `cin`, `endl`, «, and ».

File I/O Operations

`ofile("filename")`	Create output file stream (overwrites existing)
`ofile("filename", append)`	Open file in append mode
`ifile("filename")`	Create input file stream

Note 7: Files are automatically closed when the variable goes out of scope (at the end of the enclosing block).

Example 4: File Operations

```
int i;
// Standard console output
cout << " std-out" << endl;
cout << " enter i= ? ";
cin >> i;

// File output (overwrite mode)
{
ofstream f("toto.txt");
f << i << "coucou'\n";
} // File automatically closed here

// File input
{
ifstream f("toto.txt");
f >> i;
} // File automatically closed here

// File output (append mode)
{
ofstream f("toto.txt", append);
f << i << "coucou'\n";
} // File automatically closed here

// Final output
cout << i << endl;
```

Key Features

- « operator for output (both console and file)
- » operator for input (both console and file)
- `endl` for line endings

- Block scope determines file lifetime
- Append mode preserves existing file content

Special Considerations

- Windows may require full paths for files
- File permissions follow operating system rules
- Binary mode not shown here (use `ofstream` with mode flags if needed)
- Error handling should be added for production code

14th line: the time in calculation is outputted into your console (= default of standard output) using C++-like syntax. The user need not study C++ for using **freefem**++, because C++-like syntax is used for input/output, loops, flow-controls etc.
article

4.3 Error Estimation

There exist well-known error estimates between the numerical solution u_h and the exact solution u of problems (**??**) and (**??**) . If the triangulations $\{T_h\}_{h\geq 0}$ are regular , then we have the following estimates:

Energy Norm Estimate

$$|\nabla u - \nabla u_h|_{0,\Omega} \leq C_1 h \tag{4.9}$$

L² Norm Estimate

$$\|u - u_h\|_{0,\Omega} \leq C_2 h^2 \tag{4.10}$$

where:

- C_1 and C_2 are constants independent of h
- These estimates hold when $u \in H^2(\Omega)$
- The solution u is guaranteed to be in $H^2(\Omega)$ if Ω is convex

Mathematical Notation:

- $|\cdot|_{0,\Omega}$ denotes the L^2 norm of the gradient (seminorm)
- $\|\cdot\|_{0,\Omega}$ denotes the standard L^2 norm
- $H^2(\Omega)$ is the Sobolev space of functions with square-integrable second derivatives

4.3.1 Error Verification Implementation

We verify the error estimates (**??**) and (**??**) numerically. To avoid numerical errors from derivatives, we use the following identity for (**??**) :

$$\begin{aligned}\int_\Omega |\nabla u - \nabla u_h|^2\, dxdy &= \int_\Omega \nabla u \cdot \nabla (u - 2u_h)\, dxdy + \int_\Omega \nabla u_h \cdot \nabla u_h\, dxdy \\ &= \int_\Omega f(u - 2u_h)\, dxdy + \int_\Omega f u_h\, dxdy\end{aligned}$$

The constants C_1 and C_2 depend on $\mathcal{T}_h$ and f. We compute them using **FreeFem++**. Since exact solutions are rarely available in elementary form , we use an approximate solution u_0 in $V_h(\mathcal{T}_h, P_2)$ with $h \sim 0$ as reference.

Example 5: Error Computation

```
// High-resolution reference solution
mesh Th0 = square(100,100);
fespace V0h(Th0,P2);
V0h u0,v0;
func f = x*y;  // sin(pi*x)*cos(pi*y);

solve Poisson0(u0,v0) =
int2d(Th0)( dx(u0)*dx(v0) + dy(u0)*dy(v0) )  // bilinear form
- int2d(Th0)( f*v0 )                          // linear form
+ on(1,2,3,4,u0=0);                           // boundary ...
    condition

plot(u0);

// Error analysis for different mesh sizes
real[int] errL2(10), errH1(10);

for (int i=1; i≤10; i++) {
mesh Th = square(5+i*3,5+i*3);
fespace Vh(Th,P1);
fespace Ph(Th,P0);
Ph h = hTriangle;  // get size of all triangles
Vh u,v;

solve Poisson(u,v) =
int2d(Th)( dx(u)*dx(v) + dy(u)*dy(v) )
- int2d(Th)( f*v )
+ on(1,2,3,4,u=0);
```

```
28
29      // Compute errors
30      errL2[i-1] = sqrt( int2d(Th)( (u0-u)^2 ) );
31      errH1[i-1] = sqrt( int2d(Th)( (dx(u0)-dx(u))^2 + ...
            (dy(u0)-dy(u))^2 ) );
32
33      cout << "h = " << h[].max << ", L2 error = " << errL2[i-1]
34      << ", H1 error = " << errH1[i-1] << endl;
35      }
```

Implementation Notes

- Line 1-11: Compute reference solution on fine mesh (P_2 elements)
- Line 15-28: Loop over progressively refined meshes
- Line 22-26: Solve Poisson equation on current mesh
- Line 29-32: Compute L^2 and H^1 errors against reference solution
- The L^2 error corresponds to estimate (**??**)
- The H^1 seminorm error corresponds to estimate (**??**)

4.3.2 Periodic Boundary Conditions

Error Constant Calculation

The previous error analysis yields the constants:

```
1      Cout <<  C1  =   << errL2.max <<   (   << errL2.min <<   )   ...
           << endl;
2      Cout <<  C2  =   << errH1.max <<   (   << errH1.min <<   )   ...
           << endl;
```

We estimate $C_1 = 0.0179253(0.0173266)$ and $C_2 = 0.0729566(0.0707543)$, where the numbers in parentheses represent the minimum values observed during computation.

Periodic Poisson Problem

We solve the Poisson equation with periodic boundary conditions: $-\Delta u = \sin\left(x + \frac{\pi}{4}\right) \cos(y + \frac{\pi}{4}$ On the square $[0, 2\pi]^2$ with:

- $u(0, y) = u(2\pi, y)$ for all y
- $u(x, 0) = u(x, 2\pi)$ for all x

Example 06 (periodic.edp)

```
Mesh Th = square(10,10,[2*x*pi,2*y*pi]);
// Define periodic finite element space
// Labels: 2 and 4 are left/right sides (y-curve abscissa)
//        1 and 3 are bottom/upper sides (x-curve abscissa)
Fespace Vh(Th,P2,periodic=[[2,y],[4,y],[1,x],[3,x]]);
Vh uh,vh;  // unknown and test function
Func f = sin(x+pi/4.)*cos(y+pi/4.);  // right-hand side

Problem laplace(uh,vh) =
Int2d(Th)( dx(uh)*dx(vh) + dy(uh)*dy(vh) )  // bilinear form
+ int2d(Th)( -f*vh )                        // linear form
;  // no boundary conditions needed (periodic)

Laplace;  // solve the problem
Plot(uh); // visualize solution
Plot(uh,ps= period.eps ,value=true);  // save to file
```

Non-Axis-Aligned Periodic Conditions

Periodic conditions can also be applied to non-axis-aligned boundaries:

Example 07 (periodic4.edp)

```
Real r = 0.25;
// Diamond-shaped domain with a hole
Border a(t=0,1){x=-t+1; y=t; label=1;}
Border b(t=0,1){x=-t; y=1-t; label=2;}
Border c(t=0,1){x=t-1; y=-t; label=3;}
Border d(t=0,1){x=t; y=-1+t; label=4;}

// Build mesh and define periodic pairs
Mesh Th = buildmesh(a(20) + b(20) + c(20) + d(20));
Fespace Vh(Th,P1,periodic=[[1,x+y],[3,x+y],[2,x-y],[4,x-y]]);
```

Figure 4.1: Solution of the periodic Poisson problem (Example 07)

4.3.3 Non-Axis-Aligned Periodic Boundary Conditions

Example 08 Continued (periodic4.edp)

```
// Circular hole inside the diamond
border e(t=0,2*pi){ x=r*cos(t); y=-r*sin(t); label=0; };

int n = 10;
mesh Th = buildmesh(a(n) + b(n) + c(n) + d(n) + e(n));
plot(Th, wait=1);

real r2 = 1.732;
func abs = sqrt(x^2 + y^2);

// Implementation notes for periodic conditions:
// For sides a (label 1) and c (label 3):
// - On side a: x     [0,1] or x - y     [-1,1]
// - On side c: x     [-1,0] or x - y     [-1,1]
// Common abscissa can be x and x+1 respectively
// or using curvilinear abscissa x-y and x-y

// First implementation option:
// fespace Vh(Th,P2,periodic=[[2,1+x],[4,x],[1,x],[3,1+x]]);

// Second implementation option (using curvilinear coordinates):
fespace Vh(Th,P2,periodic=[[2,x+y],[4,x+y],[1,x-y],[3,x-y]]);

Vh uh, vh;

// Right-hand side function
func f = (y+x+1)*(y+x-1)*(y-x+1)*(y-x-1);

// Compute mean value of f
real intf = int2d(Th)(f);
real mTh = int2d(Th)(1);
real k = intf/mTh;
cout << "Mean value k = " << k << endl;

// Define and solve the problem
problem laplace(uh,vh) =
int2d(Th)( dx(uh)*dx(vh) + dy(uh)*dy(vh) )
+ int2d(Th)( (k-f)*vh );

laplace;
plot(uh, wait=1, ps="periodic_diamond.eps");
```

Key Features of This Implementation

- Creates a diamond-shaped domain with circular hole
- Two approaches for handling periodic conditions:
  - **a** Using shifted coordinates (commented out)
  - **b** Using curvilinear coordinates (implemented)
- Computes and accounts for the mean value of the forcing function
- Visualizes both the mesh and final solution

Mathematical Notes

The periodic condition implementation ensures: $\begin{cases} u|_{\text{side 2}} = u|_{\text{side 4}} \\ u|_{\text{side 1}} = u|_{\text{side 3}} \end{cases}$ through the mapping: For sides 2 & 4: $x + y$ as common coordinate

For sides 1 & 3: $x - y$ as common coordinate

Conclusion

In this memory we reviewed some basic a posteriori error estimation techniques which broadly can be classi- fied into global error estimators for the energy norm and goal-oriented error estimators to provide error estimates and error bounds for linear quantities of interest. We also discussed the case when the goal quantity is a point value, which normally poses a difficulty since the dual solution is not in the solution space.

Abstract

In this memory we review the basic concepts to obtain a posteriori error estimates for the finite element solution of an elliptic linear model problem. We give the basic ideas to establish global error estimates for the energy norm as well as goal-oriented error estimates. While we show how these error estimation techniques are employed for our simple model problem, the emphasis of the memory is on assessing whether these procedures are ready for use in practical linear finite element analysis. We conclude that the actually practical error estimation techniques do not provide mathematically proven bounds on the error and need to be used with care. The more accurate estimation procedures also do not provide proven bounds that, in general, can be computed efficiently. We also briefly comment upon the state of error estimations in nonlinear and transient analyses and when mixed methods are used .

Bibliography

[1] Allaire.G,**Analyse numérique et optimisation :une introduction a la modélisation mathématique et a la simulation numérique**.Editions Ecole polytechnique,2005

[2] Giovanni Leonni , **A first course in Sobolev Spaces** , 2009

[3] DAVEAU CHRISTIAN ,**METHODE DES ELEMENTS FINIS**, 2005.

[4] Endre S¨uli ,**Lecture Notes on Finite Element Methods for Partial Differential Equations** , 2025

[5] Aurélien Larcher, Niyazi Cem De˜girmenci **Lecture Notes: The Finite Element Method** , 2013

[6] Jacques Baranger , Hassan El Amri , **Estimateurs A Posteriori D'erreur pour le calcul adaptatif d'écoulements quasi-newtoniens**,1991

[7] F. Hecht ,O. Pironneau, K. Ohtsuka ,**FreeFem++ Manual** , 2005