

Università degli Studi di Milano

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Matematica



TESI DI LAUREA

Discontinuous Galerkin Methods for Isogeometric Analysis

Candidato:
Federica Brunero
Matricola 790334

Relatori:
Prof. Luca Franco Pavarino
o.Univ.-Prof. Dr. Ulrich Langer
Correlatore:
Dr. Clemens Pechstein

Anno Accademico 2011–2012

Contents

Introduction	1
1 Isogeometric Analysis	3
1.1 B-splines	3
1.1.1 Knot vectors and basis functions	4
1.1.2 B-spline curves	7
1.1.3 B-spline surfaces	8
1.1.4 Knot insertion	9
1.2 Non-Uniform Rational B-Splines	10
1.2.1 NURBS basis functions	10
1.2.2 NURBS curves	11
1.2.3 NURBS surfaces	12
1.3 Multiple patches	13
1.4 NURBS as an analysis framework	13
2 A Discontinuous Galerkin formulation for multi-patch NURBS discretization	16
2.1 A Discontinuous Galerkin formulation	17
2.1.1 Motivations	17
2.1.2 The derivation of a DG formulation	17
2.2 Multi-patch NURBS discretization	22
2.3 Properties of the bilinear form $a_{DG}(\cdot, \cdot)$	23
2.3.1 Coercivity	26
2.3.2 Boundedness	29

2.3.3	Existence and uniqueness of the DG solution	31
3	Discretization error analysis	32
3.1	Error estimate in the DG-norm	33
3.2	Error estimate in the L^2 norm	36
4	Numerical results	39
4.1	GeoPDEs	39
4.1.1	Main steps to write a code with GeoPDEs	40
4.2	Continuous Galerkin (CG) methods	41
4.2.1	CG methods for a single-patch domain in 2D	41
4.2.2	CG methods for a single-patch domain in 3D	44
4.2.3	CG methods for a multi-patch domain in 2D	46
4.3	Ideas for a DG implementation	48
4.3.1	The linear system	48
4.3.2	Pseudocode for DG methods	50
5	Conclusions	53
	Bibliography	55

Introduction

Isogeometric Analysis (IGA) is a concept introduced for the first time by Tom Hughes and co-workers (see [11]). It establishes a link between the technologies of Computer Aided Design and numerical simulation via Finite Element Analysis. The main advantage of the IGA framework consists in the fact that the same function spaces can be used for both the geometric representation of the computational domain and for the approximation of the problem unknowns. Therefore, an exact representation of the computational domain is available at all times during the simulation process. We refer the reader to the monograph [7] for rather complete presentation of IGA.

Discontinuous Galerkin (DG) methods are becoming more and more popular for the approximation of PDEs. They are non-conforming methods, which allow jumps across the boundaries of the elements. There is a wide literature about this kind of methods with a Finite Element approach (e.g. see [1] and the monograph [13] for a discussion of DG methods for solving elliptic and parabolic equations and the paper [9] if interested in DG methods with discontinuous coefficients).

The aim of this work is to link these two fields together to build a solid theory about DG-IGA methods for elliptic problems with discontinuous coefficients and to give some hints about how to test the theoretical results in a Matlab implementation. The only other reference in literature we have found with DG methods in IGA setting consists in the paper [2] referred to elastic problems.

In this thesis, we consider the scalar diffusion problem with Neumann

and Dirichlet boundary conditions as model problem: Find $u : \bar{\Omega} \rightarrow \mathbb{R}$ such that

$$\begin{cases} -\operatorname{div}(\alpha \nabla u) &= f & \Omega \\ u &= g_D & \Gamma_D \\ \alpha \nabla u \cdot \mathbf{n} &= g_N & \Gamma_N \end{cases} \quad (\text{P})$$

where $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\alpha \in L^\infty(\Omega)$ is a piecewise constant and uniformly positive diffusion coefficient, and f , g_D and g_N are given functions with assigned regularity.

In Chapter 1, we recall the basics of Isogeometric Analysis.

In Chapter 2, we first introduce a partition of the domain Ω into subdomains and the definition of the *broken Sobolev spaces*. Then we introduce the concept of *jumps* and *averages* of functions along the interfaces of these subdomains. These definitions are used to derive a Discontinuous Galerkin formulation of the model problem (P) that reads as follows: Find $u \in \mathcal{V}$ such that

$$a_{DG}(u, v) = \langle F, v \rangle \quad \forall v \in \mathcal{V},$$

where $a_{DG}(\cdot, \cdot)$ is the so-called *DG* bilinear form, F is a linear functional and \mathcal{V} is a suitable broken space. We then introduce a Multi-Patch NURBS discretization and we define the *DG*-norm. Finally, we prove that the bilinear form is coercive and bounded in the discrete *DG*-space $\mathcal{V}_h \subset \mathcal{V}$. Therefore, due to Lax-Milgram's theorem, there exists a unique solution of the discrete variational problem.

Chapter 3 is devoted to the derivation of a priori discretization error estimates in the *DG*-norm and in L^2 -norm.

In Chapter 4 we first give some numerical results with a Continuous Galerkin method, by testing it with a benchmark problem for a single-patch and multi-patch domain. Then we give some suggestions and indications on how to write a code of a *DG* method. For the implementation part of this work, we use the library GeoPDEs of Matlab, which contains useful built-in functions for multi-patch domains in an Isogeometric setting.

Finally, in Chapter 5, we summarize our results and suggest possible future work.

Chapter 1

Isogeometric Analysis

Isogeometric Analysis (IGA) is a new method for the numerical solution of problems governed by partial differential equations (see [11]). The method shares some common features with the finite element method and with meshless methods, but it is more based on geometry and is inspired from Computer Aided Design (CAD) techniques. Some important goals are:

- to be geometrically exact no matter how coarse the discretization;
- to simplify mesh refinement by eliminating the need for communication with the CAD geometry once the initial mesh is constructed;
- to more tightly link the mesh generation process and CAD.

1.1 B-splines

Unlike in standard finite element analysis, the B-spline *parameter space* is local to *patches* rather than elements. That is, the reference element in FEA is mapped into a single element in the physical space and each element has its own such mapping. Alternatively, the B-spline mapping takes a patch of multiple elements in the parameter space into the physical space. This means that each element in the physical space is the image of the corresponding element in the parameter space, but the mapping itself is global to the whole

patch, rather than to the elements themselves. In this sense, patches play the role of *subdomains* within which element types are assumed to be uniform. In what follows, we will refer to ‘patches’ as subdomains comprised of many ‘elements’.

1.1.1 Knot vectors and basis functions

Definition 1. Let p be a non-negative **degree** and let $s = (s_1, \dots, s_m)$ be a **knot vector** a set of coordinates in the parametric space with $s_k \leq s_{k+1}$ for all k . If knots are equally-spaced in the parametric space, they are said to be **uniform**. An **open knot vector** is a knot vector s where the multiplicity of a knot is at most p , except for the first and the last knots which have multiplicity $p + 1$. We call **knot span** an interval $[s_k, s_{k+1}]$.

Open knot vectors are the standard in CAD literature.

Definition 2. The $n_s = m - p - 1$ univariate **B-spline basis functions** $B_{k,p}^s : (0, 1) \rightarrow \mathbb{R}$, $k = 1, \dots, n_s$ are defined recursively starting with piecewise constants ($p = 0$) as follows:

$$B_{k,0}^s(\xi) = \begin{cases} 1 & \text{for } s_k \leq \xi < s_{k+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{k,p}^s(\xi) = \frac{\xi - s_k}{s_{k+p} - s_k} B_{k,p-1}^s(\xi) + \frac{s_{k+p+1} - \xi}{s_{k+p+1} - s_{k+1}} B_{k+1,p-1}^s(\xi). \quad (1.1)$$

This is referred to as the Cox-de Boor recursion formula. When a zero denominator appears in the definition above, the corresponding function $B_{k,p}^s$ is zero and the whole term is considered to be zero.

The results of applying (1.1) to a uniform knot vector are presented in Figure 1.1. For $p = 0$ and $p = 1$, the basis functions are the same as for standard piecewise constant and linear finite element functions, respectively. On the contrary, for $p = 2$, B-spline basis functions differ from their FEA counterparts: they are identical but shifted relative to each other, whereas

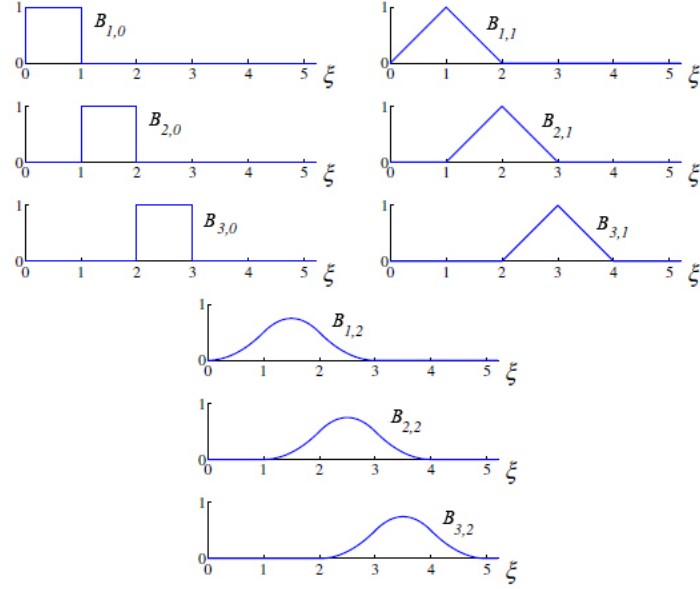


Figure 1.1: *Basis functions of degree 0, 1, 2 for uniform knot vector $s = \{0, 1, 2, 3, 4, \dots\}$. (See [11])*

the shape of a quadratic finite element function depends on whether it corresponds to an internal node or an end node. This fact holds also for higher degree.

Some important properties of B-spline basis functions are the following.

- The basis constitutes a partition of unity, i.e., $\forall \xi \quad \sum_{k=1}^{n_s} B_{k,p}^s(\xi) = 1$.
- The support of each basis function $B_{k,p}^s$ is compact and contained in the interval $[s_k, s_{k+p+1}]$ (that is, it is always $p + 1$ knot spans). Therefore, higher-degree functions have support over much larger portions of the domain than do classical FEA functions.
- Each basis function is non-negative, i.e., $B_{k,p}^s(\xi) \geq 0$ for all ξ . Therefore, all the coefficients of a mass matrix computed from a B-spline basis are non-negative.
- Basis functions of degree p have $p - 1$ continuous derivatives across the knots. This feature has some relevant consequences for the use of

splines as a basis for analysis and is one of the most distinctive features of isogeometric analysis.

In general, basis functions of degree p have $p - m_i$ continuous derivatives across the knot s_i , where m_i is the multiplicity of the value of s_i in the knot vector. When the multiplicity of a knot value is exactly p , the basis is interpolatory at that knot. When the multiplicity is $p + 1$, the basis becomes discontinuous and the patch boundary is formed.

A quadratic example of this aspect is presented in Figure 1.2 for the open non-uniform knot vector $\{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$. Note that the basis functions are interpolatory at the ends of the interval and also at $\xi = 4$, the location of a repeated knot. At this repeated knot, the functions are only C^0 -continuous, whereas elsewhere the functions are C^1 -continuous.

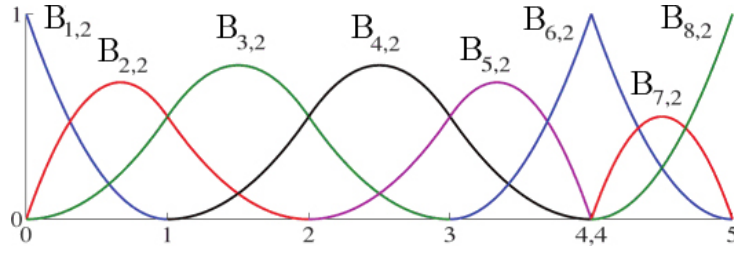


Figure 1.2: Quadratic basis functions for open, non-uniform knot vector $\{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$. (See [11])

Definition 3. The derivatives of B-spline basis functions are represented in terms of B-spline lower degree bases. For a given degree p and a knot vector s , the derivative of the k^{th} basis function is given by

$$\frac{d}{d\xi} B_{k,p}^s(\xi) = \frac{p}{s_{i+p} - s_i} B_{k,p-1}^s(\xi) - \frac{p}{s_{i+p+1} - s_{i+1}} B_{k+1,p-1}^s(\xi).$$

Again, if the denominator of these coefficients is zero, the term is defined to be zero.

1.1.2 B-spline curves

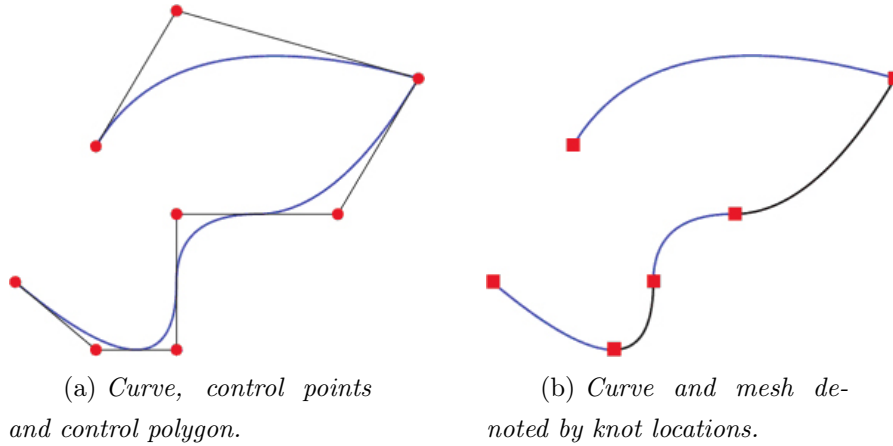
B-spline curves in \mathbb{R}^d are built by taking a linear combination of B-spline basis functions, exactly as in the classical FEA.

The vector-valued coefficients of the basis functions are called **control points** and piecewise linear interpolation of these points gives the **control polygon**. These are analogous to nodal coordinates in finite element analysis in the fact that they are the coefficients of the basis functions, but the non-interpolatory nature of the basis does not lead to a concrete interpretation of the control point values.

Definition 4. Given n_s basis functions, $B_{k,p}^s \forall k = 1, \dots, n_s$, and corresponding control points $C_k \in \mathbb{R}^d, \forall k = 1, \dots, n_s$, a piecewise-polynomial **B-spline curve** is defined as

$$C(\xi) = \sum_{k=1}^{n_s} B_{k,p}^s(\xi) C_k.$$

Piecewise linear interpolation of the control points gives the so-called **control polygon**.



Note in Figures 1.3a and 1.3b (see [11]) the difference between the **control points** and the **knot locations**, which are the images of the knots in the physical space and which partition the curve into **elements**.

Important properties of B-spline curves are the following.

- B-spline curves of degree p have continuous derivatives of order $p - 1$ in case of non-repeated knots or control points;
- *Locality*: due to the compact support of the B-spline basis functions, moving a single control point can affect the geometry of no more than $p + 1$ elements of the curve;
- *Convex hull property*: a B-spline curve is completely contained within the convex hull defined by its control points. For a curve of degree p , we define the convex hull as the union of all of the convex hulls formed by $p + 1$ successive control points.
- Repeating a knot or control point k times decreases the number of continuous derivatives by k ;
- Affine covariance: an affine transformation of a B-spline curve is obtained by applying the transformation directly to the control points.

1.1.3 B-spline surfaces

Definition 5. Let $\{B_{k,p}^s\}_{k=1}^{n_s}$ and $\{B_{l,q}^t\}_{l=1}^{n_t}$ be two families of univariate B-spline basis functions defined by the degrees p and q and the open knot vectors

$$\begin{aligned} s &= (s_1, \dots, s_{n_s+p+1}) \\ t &= (t_1, \dots, t_{n_t+q+1}) \end{aligned}$$

respectively. Given a **control net** $\{P_{(k,l)}\}$, a **tensor product B-spline surface** is defined by

$$S(\xi_1, \xi_2) = \sum_{k=1}^{n_s} \sum_{l=1}^{n_t} B_{k,p}^s(\xi_1) B_{l,q}^t(\xi_2) P_{(k,l)}.$$

Many of the properties of a B-spline surface are the result of its tensor product nature.

- The basis is pointwise non-negative and forms a partition of unity as,

$$\forall(\xi_1, \xi_2) \in [s_1, s_{n_s+p+1}] \times [t_1, t_{n_t+q+1}],$$

$$\sum_{k=1}^{n_s} \sum_{l=1}^{n_t} B_{k,p}^s(\xi_1) B_{l,q}^t(\xi_2) = \left(\sum_{k=1}^{n_s} B_{k,p}^s(\xi_1) \right) \left(\sum_{l=1}^{n_t} B_{l,q}^t(\xi_2) \right) = 1.$$

- The number of continuous partial derivatives in a given parametric direction may be determined from the associated one-dimensional knot vector and degree.
- The surface again possesses the property of affine covariance and has a strong convex hull property.
- The local support of the basis functions also follows directly from the one-dimensional functions that form them. The support of a given bi-variate function $\tilde{B}_{k,l;p,q}^{s,t}(\xi_1, \xi_2) = B_{k,p}^s(\xi_1) B_{l,q}^t(\xi_2)$ is exactly $[s_k, s_{k+p+1}] \times [t_l, t_{l+q+1}]$.

Associated to the knot vectors there is a **mesh** \mathcal{Q} , that is, a partition of the parametric domain $I = (0, 1)^2$ into two-dimensional open knot elements:

$$\mathcal{Q} := \{Q = (s_k, s_{k+1}) \otimes (t_l, t_{l+1}) \mid Q \neq \emptyset, 1 \leq k \leq n_s + p, 1 \leq l \leq n_t + q\}.$$

Definition 6. The mesh is **locally quasi-uniform** if the ratio of the sizes of two neighboring elements is uniformly bounded.

1.1.4 Knot insertion

The analogue of h -refinement of the finite element method is knot insertion. The main point here is that knots can be inserted without changing a curve geometrically or parametrically. Given a knot vector $s = \{s_1, \dots, s_{n_s+p+1}\}$, we want to insert a new knot $\bar{s} \in [s_l, s_{l+1}]$. The new $n_s + 1$ basis functions are formed recursively, using the definitions in (1.1) with the new knot vector $\{s_1, \dots, s_l, \bar{s}, s_{l+1}, \dots, s_{n_s+p+1}\}$. The new control points, $\{\bar{C}_1, \bar{C}_2, \dots, \bar{C}_{n_s+1}\}$ are constructed from the original control points $\{C_1, C_2, \dots, C_{n_s}\}$ by

$$\bar{C}_k = \alpha_k C_k + (1 - \alpha_k) C_{k-1}, \quad (1.2)$$

where

$$\alpha_k = \begin{cases} 1 & 1 \leq k \leq l - p \\ \frac{\bar{s} - s_k}{s_{k+p} - s_k} & l - p + 1 \leq k \leq l \\ 0 & l + 1 \leq k \leq n_s + p + 2. \end{cases}$$

Knot values already present in the knot vector may be repeated in this way, thereby increasing their multiplicity, but the continuity of the basis will be reduced. However, continuity of the curve is preserved by choosing the control points as in (1.2). Insertion of new knot values has similarities with the classical h -refinement strategy in finite element analysis as it splits existing elements into new ones. It differs, however, in the number of new functions that are created, as well as in the continuity of the basis across the newly created element boundaries.

1.2 Non-Uniform Rational B-Splines

The step from B-splines to NURBS is a relevant one because we gain the ability to exactly represent a wide range of objects that cannot be exactly represented by polynomials.

Some useful geometric entities in \mathbb{R}^d can be obtained by projective transformation of B-spline entities in \mathbb{R}^{d+1} . For instance, conic sections can be exactly represented with projective transformations of piecewise quadratic curves.

1.2.1 NURBS basis functions

We need to construct a basis for the NURBS space from knot vectors, and to build curves and surfaces from linear combinations of basis functions and control points. In this way, everything that we have said about B-splines will also be true of NURBS.

Definition 7. Let w_k , $k \in \{1 \dots, n_s\}$ be positive **weights**. The **univariate NURBS basis functions** are given by the piecewise rational functions

$$R_k(\xi) = \frac{B_{k,p}^s(\xi)w_k}{W(\xi)}, \quad (1.3)$$

where

$$W(\xi) = \sum_{\hat{k}=1}^{n_s} B_{\hat{k},p}^s(\xi) w_{\hat{k}}.$$

As the NURBS basis functions are constructed from the B-spline basis functions, the derivatives of rational functions will depend on the derivatives of their non-rational counterparts as well.

Definition 8. Simply applying the quotient rule to (1.3), the derivatives of the univariate NURBS basis functions are

$$\frac{d}{d\xi} R_k(\xi) = w_k \frac{W(\xi) B_{k,p}^s(\xi)' - W'(\xi) B_{k,p}^s(\xi)}{(W(\xi))^2},$$

where

$$W'(\xi) = \sum_{\hat{k}=1}^{n_s} B_{\hat{k},p}^s(\xi)' w_{\hat{k}}.$$

1.2.2 NURBS curves

We obtain the control points for the NURBS curve by performing a projective transformation to the control points of the B-spline curve. In this context, the B-spline, $C^w(\xi)$, is called the ‘projective curve’ with its associated ‘projective control points’, P_k^w , while the term ‘curve’ and ‘control points’ are reserved for the NURBS objects $C(\xi)$ and P_k , respectively.

With a given projective B-spline curve and its associated projective control points in hand, the **control points** for the NURBS curve are obtained by the following relations:

$$\begin{aligned} (P_k)_j &= \frac{(P_k^w)_j}{w_k}, \quad j = 1, \dots, d \\ w_k &= (P_k^w)_{d+1}, \end{aligned} \tag{1.4}$$

where $(P_k)_j$ is the j^{th} component of the vector P_k and w_k is referred to as the k^{th} weight.

Definition 9. Using the definition of NURBS basis functions and (1.4) leads to an equation for a NURBS curve, as follows

$$C(\xi) = \sum_{k=1}^{n_s} R_k(\xi) P_k. \tag{1.5}$$

1.2.3 NURBS surfaces

We denote the set of all double-indices (k, l) of NURBS basis functions by

$$\mathcal{R} = \{(k, l) : k \in \{1, \dots, n_s\}, l \in \{1, \dots, n_t\}\}.$$

Definition 10. Let $w_{(k,l)}$, $(k, l) \in \mathcal{R}$, be positive **weights**. We define the **bivariate NURBS basis functions** as follows:

$$R_{(k,l)}(\xi_1, \xi_2) = \frac{R_{(k,l)}^N(\xi_1, \xi_2)}{R^D(\xi_1, \xi_2)},$$

where the numerator and the denominator are given by

$$\begin{aligned} R_{(k,l)}^N(\xi_1, \xi_2) &= B_{k,p}^s(\xi_1) B_{l,q}^t(\xi_2) w_{(k,l)} \\ R^D(\xi_1, \xi_2) &= \sum_{(k',l') \in \mathcal{R}} R_{(k',l')}^N(\xi_1, \xi_2). \end{aligned}$$

The main advantage of the use of this kind of basis functions is that the fields considered in the problem (in the following, the temperature) are represented in terms of the same basis functions as the geometry.

Definition 11. Given a control net of control points $P_{(k,l)} \in \mathbb{R}^2$, the two dimensional **NURBS-surface** $G : I \rightarrow G(I)$ is defined by

$$G(\xi_1, \xi_2) = \sum_{(k,l) \in \mathcal{R}} R_{(k,l)}(\xi_1, \xi_2) P_{(k,l)}.$$

For brevity of notations, we collapse the double-indices (k, l) to \mathbf{k} . We define also the functions

$$\hat{R}_{\mathbf{k}} = R_{\mathbf{k}} \circ G^{-1}, \quad \forall \mathbf{k} \in \mathcal{R}.$$

The coefficients of the basis functions are the *degrees of freedom*.

Important properties of NURBS are the following.

- NURBS basis functions form a partition of unity.
- The continuity and support of NURBS basis functions are the same as for B-splines.

- NURBS have the property of affine covariance.
- If weights are equal, NURBS become B-splines.
- NURBS surfaces are the projective transformations of tensor product piecewise polynomial entities.

1.3 Multiple patches

In almost all practical circumstances, it will be necessary to describe domains with multiple NURBS patches. For instance, if different material or physical models are used in different parts of the domain, it might simplify things to describe these subdomains by different patches. Moreover, if different subdomains are to be assembled in parallel on a multiple processor machine, it is convenient from the point of view of data structures not to have a single patch split between different processors.

1.4 NURBS as an analysis framework

In the framework of NURBS, the basis functions are usually not interpolatory. There are two kinds of mesh: the control mesh and the physical mesh.

- The **control mesh** consists of multilinear elements and it is like a scaffold that controls the geometry. Namely, it interpolates the control points.
- The **physical mesh** is a decomposition of the actual geometry. There are two notions of elements in the physical mesh: the patch and the knot span.
 1. The **patch** may be thought of as a macro-element or subdomain. Each patch has two representations, one in parent domain and one in physical domain.

2. Each patch can be decomposed into **knot spans**. These define element domains where basis functions are smooth. They may be thought of as micro-elements because they are the smallest entities we deal with.

One other very important notion is the **index space**, which uniquely identifies each knot and discriminates among knots having multiplicity greater than one. A schematic illustration of the ideas is presented in Figure 1.3 for a NURBS surface in \mathbb{R}^3 .

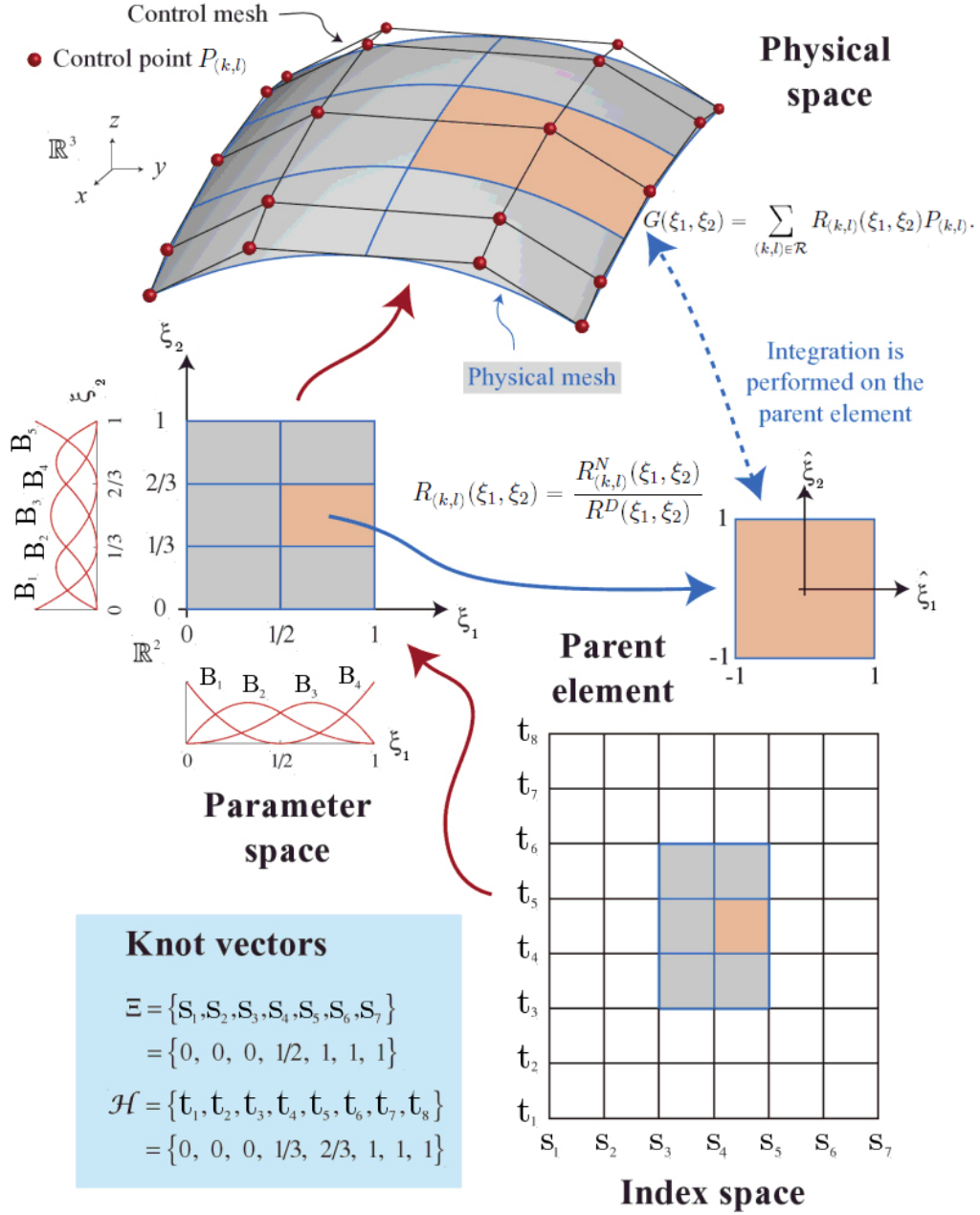


Figure 1.3: Schematic illustration of ideas of NURBS surfaces. (See [11])

Chapter 2

A Discontinuous Galerkin formulation for multi-patch NURBS discretization

Let $\Omega \subset \mathbb{R}^2$ be a bounded Lipschitz domain. Let $f \in L^2(\Omega)$, $g_D \in H^{\frac{1}{2}}(\Gamma_D)$ and $g_N \in L^2(\Gamma_N)$ be given functions. The classical boundary value problem for the scalar diffusion problem reads as follows.

Find $u : \bar{\Omega} \rightarrow \mathbb{R}$ such that

$$\begin{cases} -\operatorname{div}(\alpha \nabla u) &= f & \Omega \\ u &= g_D & \Gamma_D \\ \alpha \nabla u \cdot \mathbf{n} &= g_N & \Gamma_N \end{cases} \quad (2.1)$$

where $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\alpha \in L^\infty(\Omega)$ is a piecewise constant and uniformly positive diffusion coefficient, and \mathbf{n} is the unit outward normal vector in $\partial\Omega$.

The usual variational formulation reads:

Find $u \in H_D^1(\Omega) := \{w \in H^1(\Omega), w = g_D \text{ on } \Gamma_D\}$:

$$\int_{\Omega} \alpha \nabla u \cdot \nabla v = \int_{\Omega} f v + \int_{\Gamma_N} g_N v \quad (2.2)$$

$\forall v \in H_{0,D}^1(\Omega) := \{w \in H^1(\Omega) : w = 0 \text{ on } \Gamma_D\}$.

It can be shown that there exists a unique solution $u \in H_D^1(\Omega)$ of the problem (2.2) by using Lax-Milgram's theorem (see [14, Lemma A.34]), which, in general, states what follows.

Lemma 1 (Lax-Milgram). Let $a(\cdot, \cdot)$ be a symmetric, continuous and coercive bilinear form defined on a Hilbert space H and let F be a linear functional in the dual space H' . Consider the following problem: find $u \in H$ such that

$$a(u, v) = \langle F, v \rangle, \quad \forall v \in H. \quad (2.3)$$

Then (2.3) has a unique solution, satisfying

$$\|u\|_H \leq \frac{1}{\beta} \|F\|_{H'},$$

where β is the constant of coercivity.

2.1 A Discontinuous Galerkin formulation

2.1.1 Motivations

The ‘usual’ methods choose an approximation space $\mathcal{V}_h \subset \mathcal{V} := H_0^1$. Methods with this property are called *conforming*. *Non-conforming* means that $\mathcal{V}_h \not\subset \mathcal{V}$. DG is non-conforming.

In the framework of DG methods, we are not confined by continuity or differentiability requirements at element interfaces. Numerical fluxes will help us enforce the regularity requirements we are choosing. This will happen by means of penalty methods. These advantages come at a price, however. A calculation using DG methods typically has about twice the number of degrees of freedom of a conforming one.

2.1.2 The derivation of a DG formulation

Definition 12. Let $\mathcal{T}_H = \{\Omega^{(i)}\}_{i=1}^n$ be a partition of Ω into patches $\Omega^{(i)}$ such that $\bar{\Omega} = \bigcup_{i=1}^n \bar{\Omega}^{(i)}$ and $\Omega^{(i)} \cap \Omega^{(j)} = \emptyset$ for $i \neq j$. We assume in the following that $\alpha_{|\Omega^{(i)}} = \alpha_i = \text{const } \forall i = 1, \dots, n$. We define the **broken Sobolev space** as follows:

$$\mathcal{V} := H^s(\mathcal{T}_H) = \{v \in L^2(\Omega) : v|_{\Omega^{(i)}} \in H^s(\Omega^{(i)}) \quad \forall i = 1, \dots, n\}.$$

In what follows we will require $s > \frac{3}{2}$. The goal is to obtain a variational formulation of the problem (2.1) over Ω in the space \mathcal{V} . Since the continuity along the interfaces between patches is not required, first of all we multiply (2.1) by an arbitrary test function $v \in \mathcal{V}$ and integrate on each patch $\Omega^{(i)}$ separately, i.e., for $i = 1, \dots, n$,

$$-\int_{\Omega^{(i)}} \operatorname{div}(\alpha_i \nabla u) v = \int_{\Omega^{(i)}} f v.$$

By applying Green's theorem,

$$\int_{\Omega^{(i)}} \alpha_i \nabla u \cdot \nabla v - \int_{\partial\Omega^{(i)}} (\alpha_i \nabla u \cdot \mathbf{n}) v = \int_{\Omega^{(i)}} f v \quad \forall i = 1, \dots, n \quad \forall v \in \mathcal{V}. \quad (2.4)$$

By summing (2.4) over all patches, we obtain

$$\sum_{i=1}^n \int_{\Omega^{(i)}} \alpha_i \nabla u \cdot \nabla v - \sum_{i=1}^n \int_{\partial\Omega^{(i)}} (\alpha_i \nabla u \cdot \mathbf{n}) v = \int_{\Omega} f v. \quad (2.5)$$

We now want to rewrite the summation in (2.5), by taking into account the fact that, along the interfaces, the functions and the derivatives may present jumps. To do this, we introduce the following definitions.

Definition 13. Let $\Gamma^{(i,j)} = \partial\Omega^{(i)} \cap \partial\Omega^{(j)}$ be the interface between the two subdomains $\Omega^{(i)}$ and $\Omega^{(j)}$, $\Gamma^{(i,D)} = \partial\Omega^{(i)} \cap \Gamma_D$ and $\Gamma^{(i,N)} = \partial\Omega^{(i)} \cap \Gamma_N$. Then, for $v \in T(\Gamma) := \prod_{i=1}^n L^2(\partial\Omega^{(i)})$, we define respectively the **average** and the **jump** $\{v\}, [v] \in L^2(\Gamma_S)$, where $\Gamma_S = \bigcup_{i=1}^n \partial\Omega^{(i)}$, by

$$\left. \begin{aligned} \{v\} &:= \frac{v_{i|\Gamma^{(i,j)}} + v_{j|\Gamma^{(i,j)}}}{2} \\ [v] &:= v_{i|\Gamma^{(i,j)}} - v_{j|\Gamma^{(i,j)}} \end{aligned} \right\} \quad \text{on } \Gamma^{(i,j)}, i > j$$

and for $b \in \{D, N\}$:

$$\left. \begin{aligned} \{v\} &:= v_{i|\Gamma^{(i,b)}} \\ [v] &:= v_{i|\Gamma^{(i,b)}} \end{aligned} \right\} \quad \text{on } \Gamma^{(i,b)}.$$

Definition 14. For simplicity of notation, we also define the following sets. The set of **coupling edges** is

$$\mathcal{E}_C := \{\Gamma^{(i,j)} : i > j, \operatorname{meas}_1(\Gamma^{(i,j)}) > 0\}.$$

The set of **Dirichlet edges** is

$$\mathcal{E}_D := \{\Gamma^{(i,D)} : i = 1, \dots, n\}.$$

The set of **Neumann edges** is

$$\mathcal{E}_N := \{\Gamma^{(i,N)} : i = 1, \dots, n\}.$$

With this notation, we have that $\mathcal{E}_C \cup \mathcal{E}_D \cup \mathcal{E}_N = \bigcup_{i=1}^n \partial\Omega^{(i)}$. Note that these edges are not necessarily straight lines.

Definition 15. We define the diffusion coefficients on $\Gamma^{(i,j)}$ and $\Gamma^{(i,D)}$ as follows:

$$\begin{aligned}\alpha_{(i,D)} &:= \alpha_i \\ \alpha_{(i,j)} &:= \frac{\alpha_i + \alpha_j}{2}.\end{aligned}$$

Thanks to these definitions, we can now rewrite the summation in (2.5) as

$$\begin{aligned}\sum_{i=1}^n \int_{\partial\Omega^{(i)}} (\alpha_i \nabla u \cdot \mathbf{n}_i) v &= \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \int_{\Gamma^{(i,D)}} (\alpha_i \nabla u \cdot \mathbf{n}_i) v + \sum_{\Gamma^{(i,N)} \in \mathcal{E}_N} \int_{\Gamma^{(i,N)}} g_N v \\ &\quad + \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \int_{\Gamma^{(i,j)}} \left((\alpha_i \nabla u - \alpha_j \nabla u) \cdot \mathbf{n}_i \right) v.\end{aligned}$$

The normal vector \mathbf{n} is defined on $\bigcup_{i=1}^n \partial\Omega^{(i)}$ such that $\mathbf{n} = \mathbf{n}_i$ on $\Gamma^{(i,j)}$ for $i > j$. Thus, we have the following:

$$\begin{aligned}[\alpha \nabla u \cdot \mathbf{n}] &= \alpha_i \nabla u_i \cdot \mathbf{n}_i - \alpha_j \nabla u_j \cdot \mathbf{n}_i \\ \{\alpha \nabla u \cdot \mathbf{n}\} &= \frac{1}{2} (\alpha_i \nabla u_i \cdot \mathbf{n}_i + \alpha_j \nabla u_j \cdot \mathbf{n}_i).\end{aligned}$$

We note that on $\Gamma^{(i,j)}$ the following holds:

$$\begin{aligned}
[(\alpha \nabla u \cdot \mathbf{n})v] &= \alpha_i \nabla u_i \cdot \mathbf{n}_i v_i - \alpha_j \nabla u_j \cdot \mathbf{n}_i v_j \\
&= \frac{1}{2} \alpha_i \nabla u_i \cdot \mathbf{n}_i v_i + \frac{1}{2} \alpha_j \nabla u_j \cdot \mathbf{n}_i v_i - \frac{1}{2} \alpha_i \nabla u_i \cdot \mathbf{n}_i v_j - \frac{1}{2} \alpha_j \nabla u_j \cdot \mathbf{n}_i v_j \\
&\quad + \frac{1}{2} \alpha_i \nabla u_i \cdot \mathbf{n}_i v_i + \frac{1}{2} \alpha_i \nabla u_i \cdot \mathbf{n}_i v_j - \frac{1}{2} \alpha_j \nabla u_j \cdot \mathbf{n}_i v_i - \frac{1}{2} \alpha_j \nabla u_j \cdot \mathbf{n}_i v_j \\
&= \frac{1}{2} \alpha_i \nabla u_i \cdot \mathbf{n}_i (v_i - v_j) + \frac{1}{2} \alpha_j \nabla u_j \cdot \mathbf{n}_i (v_i - v_j) \\
&\quad + \frac{1}{2} \alpha_i \nabla u_i \cdot \mathbf{n}_i (v_i + v_j) - \frac{1}{2} \alpha_j \nabla u_j \cdot \mathbf{n}_i (v_i + v_j) \\
&= \{\alpha \nabla u \cdot \mathbf{n}\}[v] + [\alpha \nabla u \cdot \mathbf{n}]\{v\}.
\end{aligned}$$

By the regularity of the solution $u \in \mathcal{V}$, since $s > \frac{3}{2}$ and hence $\nabla u \in H^{\frac{1}{2}}(\bigcup_{i=1}^n \partial\Omega^{(i)})$, we have that

$$[\alpha \nabla u \cdot \mathbf{n}] = 0,$$

and therefore

$$[(\alpha \nabla u \cdot \mathbf{n})v] = \{\alpha \nabla u \cdot \mathbf{n}\}[v].$$

So, we can now express (2.5) as

$$\begin{aligned}
\sum_{i=1}^n \int_{\Omega^{(i)}} \alpha_i \nabla u \cdot \nabla v - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\alpha \nabla u \cdot \mathbf{n}\}[v] \\
= \int_{\Omega} f v + \sum_{\Gamma^{(i,N)} \in \mathcal{E}_N} \int_{\Gamma^{(i,N)}} g_N v, \tag{2.6}
\end{aligned}$$

where with $\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D$ we mean $\Gamma^{(i,j)} \in \mathcal{E}_C$ or $\Gamma^{(i,D)} \in \mathcal{E}_D$.

Now we introduce the following consistent penalty terms in the left-hand side of (2.6) to guarantee symmetry:

$$- \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \int_{\Gamma^{(i,j)}} \{\alpha \nabla v \cdot \mathbf{n}\}[u] + \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \frac{\alpha_i + \alpha_j}{2} \delta^{(i,j)} \int_{\Gamma^{(i,j)}} [u][v].$$

Then we impose weakly the Dirichlet boundary condition on Γ_D . For this purpose, we add to the left-hand side of (2.6) the following expression

$$\epsilon \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \int_{\Gamma^{(i,D)}} (\alpha_i \nabla v \cdot \mathbf{n}_i)(u - g_D) + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \alpha_i \delta^{(i,D)} \int_{\Gamma^{(i,D)}} (u - g_D)v,$$

where the expression of $\delta^{(i,D)}$ will be specified later. These terms are consistent as they vanish for the solution u of (2.2). We choose $\epsilon = -1$ to guarantee the symmetry of the bilinear form of the resulting variational formulation (see [13]). Summarizing, we obtain:

$$\begin{aligned}
& \sum_{i=1}^n \int_{\Omega^{(i)}} \alpha_i \nabla u \cdot \nabla v - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\alpha \nabla u \cdot \mathbf{n}\} [v] \\
& - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\alpha \nabla v \cdot \mathbf{n}\} [u] + \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \alpha_{(i,-)} \delta^{(i,-)} \int_{\Gamma^{(i,D)}} [u][v] \\
& = \int_{\Omega} f v + \sum_{\Gamma^{(i,N)} \in \mathcal{E}_N} \int_{\Gamma^{(i,N)}} g_N v + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \int_{\Gamma^{(i,D)}} \left(-\alpha_i \nabla v \cdot \mathbf{n} + \alpha_i \delta^{(i,D)} v \right) g_D.
\end{aligned} \tag{2.7}$$

Therefore, the final variational formulation is:

Find $u \in \mathcal{V}$ such that

$$a_{DG}(u, v) = \langle F, v \rangle \quad \forall v \in \mathcal{V}, \tag{2.8}$$

where:

- $a_{DG}(u, v) = \sum_{i=1}^n a_i(u, v) + b(u, v) + c(u, v)$ is a bilinear form, where

$$\begin{aligned}
a_i(u, v) &:= \int_{\Omega^{(i)}} \alpha_i \nabla u \cdot \nabla v; \\
b(u, v) &:= - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \left(\{\alpha \nabla u \cdot \mathbf{n}\} [v] + \{\alpha \nabla v \cdot \mathbf{n}\} [u] \right); \\
c(u, v) &:= \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \alpha_{(i,-)} \delta^{(i,-)} \int_{\Gamma^{(i,D)}} [u][v].
\end{aligned}$$

- F is a linear functional defined as

$$\langle F, v \rangle = \int_{\Omega} f v + \sum_{\Gamma^{(i,N)} \in \mathcal{E}_N} \int_{\Gamma^{(i,N)}} g_N v + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \int_{\Gamma^{(i,D)}} \left(-\alpha_i \nabla v \cdot \mathbf{n} + \alpha_i \delta^{(i,D)} v \right) g_D.$$

Note that the solution u of (2.2) solves (2.8). Furthermore, it can be shown that the solution of (2.8) is unique (see [6]).

2.2 Multi-patch NURBS discretization

To define precisely the discrete problem associated to (2.8), we use the definitions from Isogeometric Analysis that we introduced in Chapter 1.

In our specific case the physical domain $\Omega \subset \mathbb{R}^2$ is represented by n single patch NURBS mappings $G^{(i)}$, $i = 1, \dots, n$, each of which maps the parameter domain I to the patch $\Omega^{(i)} = G^{(i)}(I)$. We use the superscript (i) to indicate that knot vectors, degrees, NURBS basis functions, index sets, etc. are associated with a mapping $G^{(i)}$. We assume that the mesh $\mathcal{Q}^{(i)}$ underlying to the NURBS mapping $G^{(i)}$ is quasi-uniform for all $i = 1 \dots, n$. We denote by h_i the element size of $\mathcal{Q}^{(i)}$.

We also define the following space:

$$\mathcal{V}_h = \{v \in L^2(\Omega) : \forall i = 1, \dots, n \quad v|_{\Omega^{(i)}} \in X^{h_i}(\Omega^{(i)})\},$$

where

$$X^{h_i}(\Omega^{(i)}) = \text{span}\{\hat{R}_{\mathbf{k}}^{(i)}\}_{\mathbf{k} \in \mathcal{R}^{(i)}}$$

denotes the space of NURBS functions on $\Omega^{(i)}$. We see that $\mathcal{V}_h \subset \mathcal{V}$.

We introduce the family of meshes $\{\mathcal{Q}_{h_i}^{(i)}\}_{h_i}$, such that there exists a coarsest mesh $\mathcal{Q}_{h_{i0}}^{(i)}$ of which all the other meshes are refinement and such that the description of the geometry is fixed at the level of $\mathcal{Q}_{h_{i0}}^{(i)}$. Associated to this family, we introduce the nested family of spaces $\{\mathcal{V}_{h_i}^{(i)}\}_{h_i}$ and the family of meshes on the physical domain $\{\mathcal{K}_{h_i}^{(i)}\}_{h_i}$. When the mesh and the spaces are refined, the weights $w_{(k,l)}$ are selected such that $R^D(\xi_1, \xi_2)$ stays fixed and the control points are adjusted such that $G^{(i)}$ remains unchanged. Thus the geometry and its parametrization are held fixed in the refinement process (see [3]).

The discrete problem associated to (2.8) reads as follows:

Find $u_h \in \mathcal{V}_h$ such that

$$a_{DG}(u_h, v_h) = \langle F, v_h \rangle \quad \forall v_h \in \mathcal{V}_h. \quad (2.9)$$

2.3 Properties of the bilinear form $a_{DG}(\cdot, \cdot)$

We now want to prove that the bilinear form is coercive and bounded on \mathcal{V}_h , in order to apply the Lax-Milgram theorem and state existence and uniqueness of the solution of the discrete problem (2.9).

For this purpose, we state some results from previous literature in order to prove a useful inverse trace inequality.

Lemma 2. Let m be a non-negative integer, $Q \in \mathcal{Q}_h$ and $\omega = \mathbf{G}(Q)$. For all functions $v \in H^m(\omega)$, it holds that

$$\begin{aligned} |v \circ \mathbf{G}|_{H^m(Q)} &\leq C_{\text{shape}} \|\det \nabla \mathbf{G}^{-1}\|_{L^\infty(\omega)}^{1/2} \sum_{j=0}^m \|\nabla \mathbf{G}\|_{L^\infty(Q)}^j |v|_{H^j(\omega)} \\ |v|_{H^m(\omega)} &\leq C_{\text{shape}} \|\det \nabla \mathbf{G}\|_{L^\infty(Q)}^{1/2} \|\nabla \mathbf{G}\|_{L^\infty(Q)}^{-m} \sum_{j=0}^m |v \circ \mathbf{G}|_{H^j(Q)}, \end{aligned}$$

where C_{shape} is a uniform constant which depends only on the shape of ω .

Proof. See [3, Lemma 3.5]. □

Lemma 3. (See [3, Theorem 4.1]) It holds that

$$|v|_{H^2(\omega)} \leq C_{\text{shape}} h_\omega^{-1} |v|_{H^1(\omega)} \quad \forall \omega \in \mathcal{K}_h, \forall v \in \mathcal{V}_h, \quad (2.10)$$

where h_ω is the size of the physical element ω .

Proof. Lemma 2 yields, for $m = 2$,

$$|v|_{H^2(\omega)} \leq C_{\text{shape}} \|\det \nabla \mathbf{G}\|_{L^\infty(Q)}^{1/2} \|\nabla \mathbf{G}\|_{L^\infty(Q)}^{-2} \|v \circ \mathbf{G}\|_{H^2(Q)}, \quad (2.11)$$

where $\omega = \mathbf{G}(Q)$. Moreover

$$\|v \circ \mathbf{G}\|_{H^2(Q)} \leq C \left\| \frac{1}{R^D} \right\|_{W^{2,\infty}(Q)} \|R^D(v \circ \mathbf{G})\|_{H^2(Q)}.$$

Since $R^D(v \circ \mathbf{G})$ is polynomial, for a usual inverse inequality we have

$$\|v \circ \mathbf{G}\|_{H^2(Q)} \leq C h_Q^{-1} \left\| \frac{1}{R^D} \right\|_{W^{2,\infty}(Q)} \|R^D(v \circ \mathbf{G})\|_{H^1(Q)}. \quad (2.12)$$

We now have, using again Lemma 2,

$$\begin{aligned} \|R^D(v \circ \mathbf{G})\|_{H^1(Q)} &\leq C \|R^D\|_{W^{1,\infty}(Q)} \|v \circ \mathbf{G}\|_{H^1(Q)} \\ &\leq C_{\text{shape}} \|R^D\|_{W^{1,\infty}(Q)} \|\det \nabla \mathbf{G}^{-1}\|_{L^\infty(\omega)}^{1/2} \sum_{j=0}^1 \|\nabla \mathbf{G}\|_{L^\infty(Q)}^j |v|_{H^j(\omega)}. \end{aligned} \quad (2.13)$$

Joining the above bounds (2.11), (2.12) and (2.13), we finally obtain

$$|v|_{H^2(\omega)} \leq C_{\text{shape}} h_Q^{-1} \sum_{j=0}^1 \|\nabla \mathbf{G}\|_{L^\infty(Q)}^{j-2} |v|_{H^j(\omega)}. \quad (2.14)$$

Let now v_ω represent the constant function equal to the average of v on ω ; note that $v_\omega \in \mathcal{V}_h$. Therefore, applying (2.14), classical polynomial interpolation results and recalling that $\|\nabla \mathbf{G}\|_{L^\infty(Q)}^{-1} h_\omega = h_Q$, it easily follows that

$$\begin{aligned} |v|_{H^2(\omega)} &= |v - v_\omega|_{H^2(\omega)} \\ &\leq C_{\text{shape}} h_Q^{-1} \sum_{j=0}^1 \|\nabla \mathbf{G}\|_{L^\infty(Q)}^{j-1} |v - v_\omega|_{H^j(\omega)} \\ &\leq C_{\text{shape}} h_\omega^{-1} |v|_{H^1(\omega)}. \end{aligned}$$

□

Now, we are ready to state and prove the following significant result, which will be essential for the proofs of the properties of the bilinear form.

Lemma 4 (Inverse trace inequality). There exists a uniform constant $C_t > 0$ such that, for all $u \in \mathcal{V}_h$, $i = 1, \dots, n$ and for all edges $E \in \mathcal{E}_C \cup \mathcal{E}_D \cup \mathcal{E}_N$ with $E \subset \partial\Omega^{(i)}$,

$$\|\nabla u_i\|_{L^2(E)}^2 \leq \frac{C_t}{h_i} \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 \quad (2.15)$$

where $u_i = u|_{\Omega^{(i)}}$ and h_i is the element size within the patch $\Omega^{(i)}$.

Proof. Let $e \subset E$ be the image of an edge of $Q \in \mathcal{Q}^{(i)}$ under the NURBS map $G^{(i)}$, and observe that, due to our quasi-uniformity assumption, there exist uniform constants C_1, C_2 only depending on $\Omega^{(i)}$ such that

$$C_1 h_i \leq h_e := \text{diam}(e) \leq C_2 h_i.$$

Let $\omega_e = G^{(i)}(Q)$ be the physical element attached to e . Since ω_e is a Lipschitz domain, the usual trace inequality implies that (see [14, (3.16) page 67]) there exists $\tilde{C} > 0 \forall w \in H^1(\omega_e)$ such that

$$\|w\|_{L^2(e)}^2 \leq \tilde{C} \left(\frac{1}{h_e} \|w\|_{L^2(\omega_e)}^2 + h_e |w|_{H^1(\omega_e)}^2 \right).$$

Since e and ω_e can be transformed to their reference counterparts in $\mathcal{Q}^{(i)}$, it can be argued that \tilde{C} is indeed a uniform constant.

Since $u|_{\omega_e} \in H^2(\omega_e)$, this implies

$$\|\nabla u_i\|_{L^2(e)}^2 \leq \tilde{C} \left(\frac{1}{h_e} \|\nabla u_i\|_{L^2(\omega_e)}^2 + h_e |\nabla u_i|_{H^1(\omega_e)}^2 \right). \quad (2.16)$$

Moreover, we use the inverse inequality proved in Lemma 3, which is related to ω_e ,

$$|\nabla u_i|_{H^1(\omega_e)} \leq C_{\text{shape}} h_i^{-1} \|\nabla u_i\|_{L^2(\omega_e)}, \quad (2.17)$$

where C_{shape} is a uniform constant depending only on the shapes of $\Omega^{(i)}$. Combining the above estimates, we are able to prove the following:

$$\begin{aligned} \|\nabla u_i\|_{L^2(E)}^2 &= \sum_{e \in E} \|\nabla u_i\|_{L^2(e)}^2 \\ &\stackrel{(2.16)}{\leq} \tilde{C} \sum_{e \in E} \left(\frac{1}{h_e} \|\nabla u_i\|_{L^2(\omega_e)}^2 + h_e |\nabla u_i|_{H^1(\omega_e)}^2 \right) \\ &\stackrel{(2.17)}{\leq} \tilde{C} \sum_{e \in E} \left(\frac{1}{h_e} \|\nabla u_i\|_{L^2(\omega_e)}^2 + C_{\text{shape}}^2 h_e h_i^{-2} \|\nabla u_i\|_{L^2(\omega_e)}^2 \right) \\ &\stackrel{h_e \approx h_i}{\leq} \frac{C_t}{h_i} \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2, \end{aligned}$$

which completes the proof. \square

Thanks to the result of Lemma 4, in the following, we will prove the properties related to the bilinear form, by choosing

$$\delta^{(i,-)} = \frac{\delta}{h_i}, \quad (2.18)$$

where $\delta \in \mathbb{R}^+$ is a global parameter which has to be chosen sufficiently large (see Lemma 5). This means that, when changing the discretization space \mathcal{V}_h ,

we have to adapt $\delta^{(i,-)}$ and, therefore, the bilinear form a_{DG} and the linear functional F will change.

We also introduce the following definition (see [12]).

Definition 16. For (i, j) such that $\Gamma^{(i,j)} \neq \emptyset$, we collect the indices of those basis functions in $\Omega^{(i)}$ whose support intersects the interface $\Gamma^{(i,j)}$:

$$\mathcal{B}(i, j) = \{\mathbf{k} \in \mathcal{R}^{(i)} : \text{supp} \hat{R}_{\mathbf{k}}^{(i)} \cap \Gamma^{(i,j)} \neq \emptyset\}.$$

We say that two subdomains $\Omega^{(i)}$ and $\Omega^{(j)}$ are **fully matching** if the following two conditions are fulfilled:

1. The interface $\Gamma^{(i,j)}$ is the image of an entire edge of the respective parameter domains;
2. For each index $\mathbf{k} \in \mathcal{B}(i, j)$, there must be a unique index $\mathbf{l} \in \mathcal{B}(j, i)$, such that

$$\hat{R}_{\mathbf{k}}^{(i)}|_{\Gamma^{(i,j)}} = \hat{R}_{\mathbf{l}}^{(j)}|_{\Gamma^{(i,j)}}.$$

In what follows, we assume for simplicity the case of fully matching subdomains. This implies that on $\Gamma^{(i,j)}$ it holds $h_i \approx h_j$.

Definition 17. We define the **energy norm** for the Discontinuous Galerkin method as follows: for all $v \in \mathcal{V}$,

$$\|v\|_{DG}^2 = \sum_{i=1}^n \alpha_i \|\nabla v_i\|_{L^2(\Omega^{(i)})}^2 + \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\alpha_{(i,-)}}{h_i} \int_{\Gamma^{(i,-)}} [v]^2.$$

2.3.1 Coercivity

Lemma 5 (Coercivity of $a_{DG}(\cdot, \cdot)$). There exist $\delta_0 > 0$ and $D > 0$ such that for $\delta > \delta_0$ and $u \in \mathcal{V}_h$

$$a_{DG}(u, u) \geq D \|u\|_{DG}^2, \quad (2.19)$$

where D is independent of h_i , n and α_i .

Proof. Using (2.18) and the Definition 17, we easily obtain the identity

$$\begin{aligned} a_{DG}(u, u) &= \|u\|_{DG}^2 + b(u, u) + c(u, u) - \frac{1}{\delta}c(u, u) \\ &= \|u\|_{DG}^2 + b(u, u) + \frac{\delta - 1}{\delta}c(u, u). \end{aligned}$$

Note that the last term scales linearly with $(\delta - 1)$ (see the definition of $c(\cdot, \cdot)$ and (2.18)). The proof reduces to show that

$$-b(u, u) \leq \frac{1}{2} \sum_{i=1}^n a_i(u, u) + \frac{\delta_0}{\delta} c(u, u), \quad (2.20)$$

because, if (2.20) holds, then we have

$$b(u, u) \geq -\frac{1}{2} \sum_{i=1}^n a_i(u, u) - \frac{\delta_0}{\delta} c(u, u).$$

Therefore

$$a_{DG}(u, u) \geq \|u\|_{DG}^2 - \frac{1}{2} \sum_{i=1}^n a_i(u, u) + \frac{\delta - 1 - \delta_0}{\delta} c(u, u). \quad (2.21)$$

We know that $\delta > \delta_0$; this is equivalent to $\delta - 1 - \delta_0 \geq -1 + D$, for some $D \in (0, \frac{1}{2}]$. So, (2.21) becomes

$$\begin{aligned} a_{DG}(u, u) &\geq \|u\|_{DG}^2 - \frac{1}{2} \sum_{i=1}^n a_i(u, u) + \frac{D - 1}{\delta} c(u, u) \\ &= \sum_{i=1}^n a_i(u, u) + \frac{1}{\delta} c(u, u) - \frac{1}{2} \sum_{i=1}^n a_i(u, u) + \frac{D - 1}{\delta} c(u, u) \\ &= \frac{1}{2} \sum_{i=1}^n a_i(u, u) + \frac{D}{\delta} c(u, u) \\ &\stackrel{D \in (0, \frac{1}{2}]}{\geq} D \|u\|_{DG}^2. \end{aligned}$$

Now, let us prove (2.20). We recall that we have by definition

$$b(u, u) = -2 \sum_{\Gamma(i, -) \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma(i, -)} \{\alpha \nabla u \cdot \mathbf{n}\} [u].$$

We observe that, if $\Gamma^{(i,j)} \in \mathcal{E}_C$, then

$$\left| \int_{\Gamma^{(i,j)}} \{\alpha \nabla u \cdot \mathbf{n}\} [u] \right| \leq \frac{1}{2} (\alpha_i \|\nabla u_i\|_{L^2(\Gamma^{(i,j)})} + \alpha_j \|\nabla u_j\|_{L^2(\Gamma^{(i,j)})}) \|u_i - u_j\|_{L^2(\Gamma^{(i,j)})}.$$

If $\Gamma^{(i,D)} \in \mathcal{E}_D$, instead, we have

$$\left| \int_{\Gamma^{(i,D)}} \{\alpha \nabla u \cdot \mathbf{n}\} [u] \right| \leq \alpha_i \|\nabla u_i\|_{L^2(\Gamma^{(i,D)})} \|u_i\|_{L^2(\Gamma^{(i,D)})}.$$

Therefore, using Lemma 4,

$$\begin{aligned} |b(u, u)| &\leq 2 \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \left[\frac{1}{2} (\alpha_i \|\nabla u_i\|_{L^2(\Gamma^{(i,j)})} + \alpha_j \|\nabla u_j\|_{L^2(\Gamma^{(i,j)})}) \|u_i - u_j\|_{L^2(\Gamma^{(i,j)})} \right] \\ &\quad + 2 \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \alpha_i \|\nabla u_i\|_{L^2(\Gamma^{(i,D)})} \|u_i\|_{L^2(\Gamma^{(i,D)})} \\ &\stackrel{\text{Lemma 4}}{\leq} \sqrt{C_t} \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \left(\frac{\alpha_i}{\sqrt{h_i}} \|\nabla u_i\|_{L^2(\Omega^{(i)})} + \frac{\alpha_j}{\sqrt{h_j}} \|\nabla u_j\|_{L^2(\Omega^{(j)})} \right) \|u\|_{L^2(\Gamma^{(i,j)})} + \\ &\quad + \sqrt{C_t} \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} 2 \frac{\alpha_i}{\sqrt{h_i}} \|\nabla u_i\|_{L^2(\Omega^{(i)})} \|u\|_{L^2(\Gamma^{(i,D)})}. \end{aligned} \quad (2.22)$$

By using Young's inequality, i.e.

$$ab \leq \varepsilon a^2 + \frac{b^2}{4\varepsilon} \quad \forall \varepsilon > 0, \quad (2.23)$$

with

$$\begin{aligned} a &= \|\nabla u_i\|_{L^2(\Omega^{(i)})} \\ b &= \frac{1}{\sqrt{h_i}} \|u\|_{L^2(\Gamma^{(i,-)})} \end{aligned}$$

we obtain

$$\frac{\alpha_i}{\sqrt{h_i}} \|\nabla u_i\|_{L^2(\Omega^{(i)})} \|u\|_{L^2(\Gamma^{(i,-)})} \stackrel{(2.23)}{\leq} \alpha_i \left[\varepsilon \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 + \frac{1}{4\varepsilon h_i} \|u\|_{L^2(\Gamma^{(i,-)})}^2 \right],$$

and analogously for u_j . Therefore, using that $h_i \approx h_j$, we have the following,

$$\begin{aligned}
|b(u, u)| &\leq \sqrt{C_t} \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \left[\alpha_i(\varepsilon \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 + \frac{1}{4\varepsilon h_i} \| [u] \|_{L^2(\Gamma^{(i,j)})}^2) \right. \\
&\quad \left. + \alpha_j(\varepsilon \|\nabla u_j\|_{L^2(\Omega^{(j)})}^2 + \frac{1}{4\varepsilon h_j} \| [u] \|_{L^2(\Gamma^{(i,j)})}^2) \right] \\
&\quad + 2\sqrt{C_t} \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \alpha_i(\varepsilon \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 + \frac{1}{4\varepsilon h_i} \| [u] \|_{L^2(\Gamma^{(i,D)})}^2) \\
&= \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \left[\varepsilon \sqrt{C_t} (\alpha_i \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 + \alpha_j \|\nabla u_j\|_{L^2(\Omega^{(j)})}^2) \right. \\
&\quad \left. + \frac{\sqrt{C_t}}{4\varepsilon} \left(\frac{\alpha_i}{h_i} \| [u] \|_{L^2(\Gamma^{(i,j)})}^2 + \frac{\alpha_j}{h_j} \| [u] \|_{L^2(\Gamma^{(i,j)})}^2 \right) \right] \\
&\quad + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \left(\varepsilon \sqrt{C_t} \alpha_i \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 + \frac{\sqrt{C_t}}{2\varepsilon} \frac{\alpha_i}{h_i} \| [u] \|_{L^2(\Gamma^{(i,D)})}^2 \right) \\
&\leq 4\varepsilon \sqrt{C_t} \sum_{i=1}^n \alpha_i \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 + \frac{\sqrt{C_t}}{4\varepsilon} \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\alpha_{(i,-)}}{h_i} \| [u] \|_{L^2(\Gamma^{(i,-)})}^2 \\
&\stackrel{\varepsilon = \frac{1}{8\sqrt{C_t}}}{=} \frac{1}{2} \sum_{i=1}^n \alpha_i \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 + 2C_t \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\alpha_{(i,-)}}{h_i} \| [u] \|_{L^2(\Gamma^{(i,-)})}^2,
\end{aligned}$$

which is (2.20) with $\delta_0 = 2C_t$. \square

2.3.2 Boundedness

Lemma 6 (Boundedness of $a_{DG}(\cdot, \cdot)$). There exists $M > 0$ such that $\forall u, v \in \mathcal{V}_h$

$$|a_{DG}(u, v)| \leq M \|u\|_{DG} \|v\|_{DG}, \quad (2.24)$$

where M is independent of h_i , n and α_i .

Proof. Since $a_{DG}(u, v) = \sum_{i=1}^n a_i(u, v) + b(u, v) + c(u, v)$ we estimate each term separately. Let $u, v \in \mathcal{V}_h$ be arbitrary but fixed.

- With Cauchy-Schwarz in \mathbb{R}^m where $m = |\mathcal{E}_C \cup \mathcal{E}_D|$, we can easily

conclude that

$$\begin{aligned} \left| \sum_{i=1}^n a_i(u, v) \right| &\leq \sum_{i=1}^n \alpha_i \|\nabla u_i\|_{L^2(\Omega^{(i)})} \|\nabla v_i\|_{L^2(\Omega^{(i)})} \\ &\leq \left(\sum_{i=1}^n \alpha_i \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \cdot \left(\sum_{i=1}^n \alpha_i \|\nabla v_i\|_{L^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \\ &\leq \|u\|_{DG} \|v\|_{DG}. \end{aligned}$$

- For the third term we use again the Cauchy-Schwarz inequality in \mathbb{R}^m :

$$\begin{aligned} |c(u, v)| &\leq \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \alpha_{(i,-)} \frac{\delta}{h_i} \|[u]\|_{L^2(\Gamma^{(i,-)})} \|[v]\|_{L^2(\Gamma^{(i,-)})} \\ &\leq \delta \left(\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\alpha_{(i,-)}}{h_i} \|[u]\|_{L^2(\Gamma^{(i,-)})}^2 \right)^{\frac{1}{2}} \left(\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\alpha_{(i,-)}}{h_i} \|[v]\|_{L^2(\Gamma^{(i,-)})}^2 \right)^{\frac{1}{2}} \\ &\leq \delta \|u\|_{DG} \|v\|_{DG}. \end{aligned}$$

- Now we proceed with the estimate of the second term of the bilinear form:

$$|b(u, v)| = \left| \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \left(\{\alpha \nabla u \cdot \mathbf{n}\}[v] + \{\alpha \nabla v \cdot \mathbf{n}\}[u] \right) \right|.$$

We know, following the same steps as in the proof of coercivity, that

$$\begin{aligned} |b(u, v)| &\leq \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \left[\frac{1}{2} (\alpha_i \|\nabla u_i\|_{L^2(\Gamma^{(i,j)})} + \alpha_j \|\nabla u_j\|_{L^2(\Gamma^{(i,j)})}) \|v_i - v_j\|_{L^2(\Gamma^{(i,j)})} \right. \\ &\quad \left. + \frac{1}{2} (\alpha_i \|\nabla v_i\|_{L^2(\Gamma^{(i,j)})} + \alpha_j \|\nabla v_j\|_{L^2(\Gamma^{(i,j)})}) \|u_i - u_j\|_{L^2(\Gamma^{(i,j)})} \right] \\ &\quad + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \left[\alpha_i \|\nabla u_i\|_{L^2(\Gamma^{(i,D)})} \|v_i\|_{L^2(\Gamma^{(i,D)})} \right. \\ &\quad \left. + \alpha_i \|\nabla v_i\|_{L^2(\Gamma^{(i,D)})} \|u_i\|_{L^2(\Gamma^{(i,D)})} \right] \end{aligned}$$

Therefore,

$$\begin{aligned}
|b(u, v)| &\stackrel{\text{Lemma 4}}{\leq} \frac{\sqrt{C_t}}{2} \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \left[\left(\frac{\alpha_i}{\sqrt{h_i}} \|\nabla u_i\|_{L^2(\Omega^{(i)})} + \frac{\alpha_j}{\sqrt{h_j}} \|\nabla u_j\|_{L^2(\Omega^{(j)})} \right) \|v\|_{L^2(\Gamma^{(i,j)})} \right. \\
&\quad \left. + \left(\frac{\alpha_i}{\sqrt{h_i}} \|\nabla v_i\|_{L^2(\Omega^{(i)})} + \frac{\alpha_j}{\sqrt{h_j}} \|\nabla v_j\|_{L^2(\Omega^{(j)})} \right) \|u\|_{L^2(\Gamma^{(i,j)})} \right] \\
&\quad + \sqrt{C_t} \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \left[\frac{\alpha_i}{\sqrt{h_i}} \|\nabla u_i\|_{L^2(\Omega^{(i)})} \|v\|_{L^2(\Gamma^{(i,D)})} \right. \\
&\quad \left. + \frac{\alpha_i}{\sqrt{h_i}} \|\nabla v_i\|_{L^2(\Omega^{(i)})} \|u\|_{L^2(\Gamma^{(i,D)})} \right].
\end{aligned}$$

Now, using the Cauchy-Schwarz inequality, we apply on each sum above an argument similar to the following:

$$\begin{aligned}
&\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \sqrt{\alpha_{(i,-)}} \|\nabla u_i\|_{L^2(\Omega^{(i)})} \cdot \sqrt{\frac{\alpha_{(i,-)}}{h_i}} \|v\|_{L^2(\Gamma^{(i,-)})} \\
&\leq \left(\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \alpha_{(i,-)} \|\nabla u_i\|_{L^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \cdot \left(\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\alpha_{(i,-)}}{h_i} \|v\|_{L^2(\Gamma^{(i,-)})}^2 \right)^{\frac{1}{2}} \\
&\leq \|u\|_{DG} \|v\|_{DG}.
\end{aligned}$$

Hence,

$$|b(u, v)| \leq 4\sqrt{C_t} \|u\|_{DG} \|v\|_{DG}.$$

So the continuity of the bilinear form follows with $M = 1 + \delta + 4\sqrt{C_t}$. \square

2.3.3 Existence and uniqueness of the DG solution

Thanks to the results of the two previous subsections we are able to state existence and uniqueness of the solution of (2.9) applying the Lax-Milgram theorem (see Lemma 1).

Theorem 1. Let $f \in L^2(\Omega)$, $g_N \in L^2(\Gamma_N)$ and $g_D \in H^{\frac{1}{2}}(\Gamma_D)$ be given functions. Then there exists a unique solution $u \in \mathcal{V}_h$ such that

$$a_{DG}(u, v) = \langle F, v \rangle \quad \forall v \in \mathcal{V}_h$$

and there is a continuous dependence of the solution from the data.

Chapter 3

Discretization error analysis

Definition 18. Let $\Pi_{h_i} u \in \mathcal{V}_{h_{|\Omega^{(i)}}}$ be an interpolant of u on $\Omega^{(i)}$ and $\Pi_h u$ denote the interpolant on Ω such that $\Pi_h u = \Pi_{h_i} u$ on $\Omega^{(i)}$. Note that $\Pi_h u$ may be discontinuous across the interfaces $\Gamma^{(i,-)}$.

In order to provide error estimates in the DG norm and in the L^2 norm, we need the following lemmas.

Lemma 7. Let k and s be integer indices with $0 \leq k \leq s \leq p + 1$. We have, $\forall v \in H^s(\Omega^{(i)})$,

$$\begin{aligned} |v - \Pi_h v|_{H^k(\Omega^{(i)})}^2 &\leq C_{\text{shape}} \sum_{\omega \in \mathcal{K}_{h_i}} h_i^{2(s-k)} \sum_{t=0}^s \|\nabla G_i\|_{L^\infty(G_i^{-1}(\omega))}^{2(t-s)} |v|_{H^t(\omega)}^2 \\ &\leq C_{\text{shape}} h_i^{2(s-k)} \sum_{t=0}^s \|\nabla G_i\|_{L^\infty(I)}^{2(t-s)} |v|_{H^t(\Omega^{(i)})}^2. \end{aligned} \quad (3.1)$$

where we recall that $I = (0, 1)^2$ is the parameter domain.

Proof. See [3, Theorem 3.2]. □

Remark 1. Note that the NURBS space \mathcal{V}_h on the physical domain $\Omega^{(i)}$ delivers the optimal rate of convergence, as for the classical finite element spaces of degree p . Note, moreover, that a bound on the k^{th} -order seminorm of the error $v - \Pi_h v$ requires a control on the full s^{th} -order norm of v , unlike for finite elements, where only the s^{th} -order seminorm of v is involved in the

right-hand side of the estimate. This is due to the role played by the weighting function W and the geometrical map G_i .

Lemma 8. Let u be the solution of (2.8) and $\Pi_h u$ be the interpolant of u as in the definition 18. Then, for $s > \frac{3}{2}$, the following bound holds:

$$\|u - \Pi_h u\|_{DG}^2 \leq C_{\text{shape}} \sum_{i=1}^n \alpha_i h_i^{2(s-1)} \|u_i\|_{H^s(\Omega^{(i)})}^2, \quad (3.2)$$

where C_{shape} is independent of h_i , n and the jumps of α_i .

Proof. By definition, we have

$$\begin{aligned} \|u - \Pi_h u\|_{DG}^2 &= \sum_{i=1}^n \alpha_i \|\nabla(u_i - \Pi_h u_i)\|_{L^2(\Omega^{(i)})}^2 \\ &\quad + \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\alpha_{(i,-)}}{h_i} \| [u - \Pi_h u] \|_{L^2(\Gamma^{(i,-)})}^2. \end{aligned} \quad (3.3)$$

Since $u \in H^s \hookrightarrow C^0$ for $s > \frac{3}{2}$ and $d = 2, 3$, if we choose an interpolant $\Pi_h u$ such that $[\Pi_h u] = 0$, we have that the second sum in (3.3) is zero.

Therefore, we can state the following:

$$\begin{aligned} \|u - \Pi_h u\|_{DG}^2 &= \sum_{i=1}^n \alpha_i \|u_i - \Pi_h u_i\|_{H^1(\Omega^{(i)})}^2 \\ &\stackrel{\text{Lemma 7}}{\leq} C_{\text{shape}} \sum_{i=1}^n \alpha_i h_i^{2(s-1)} \|u\|_{H^s(\Omega^{(i)})}^2. \end{aligned}$$

□

3.1 Error estimate in the DG-norm

Theorem 2. Let u and u_h be the solutions of (2.8) and (2.9), respectively. For $s \geq 2$, the following bound holds:

$$\|u - u_h\|_{DG}^2 \leq D_{\text{shape}} \sum_{i=1}^n \alpha_i^2 h_i^{2(s-1)} \|u_i\|_{H^s(\Omega^{(i)})}^2, \quad (3.4)$$

where D_{shape} is independent of h_i , n and the jumps of α_i .

Proof. By the triangle inequality we have

$$\|u - u_h\|_{DG} \leq \|u - \Pi_h u\|_{DG} + \|\Pi_h u - u_h\|_{DG}. \quad (3.5)$$

Since $\Pi_h u - u_h \in \mathcal{V}_h$, we can use the coercivity of the bilinear form $a_{DG}(\cdot, \cdot)$ on $\mathcal{V}_h \times \mathcal{V}_h$:

$$\|\Pi_h u - u_h\|_{DG}^2 \leq D^{-1} a_{DG}(\Pi_h u - u_h, \Pi_h u - u_h).$$

Moreover, we observe that the following holds:

$$\begin{aligned} a_{DG}(\Pi_h u - u_h, \Pi_h u - u_h) &= a_{DG}(\Pi_h u, \Pi_h u - u_h) - a_{DG}(u_h, \Pi_h u - u_h) \\ &= a_{DG}(\Pi_h u, \Pi_h u - u_h) - \langle F, \Pi_h u - u_h \rangle \\ &= a_{DG}(\Pi_h u, \Pi_h u - u_h) - a_{DG}(u, \Pi_h u - u_h) \\ &= a_{DG}(\Pi_h u - u, \Pi_h u - u_h), \end{aligned}$$

and

$$\begin{aligned} a_{DG}(\Pi_h u - u, \Pi_h u - u_h) &= \sum_{i=1}^n \int_{\Omega^{(i)}} \alpha_i \nabla(\Pi_h u - u) \cdot \nabla(\Pi_h u - u_h) \\ &\quad - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\alpha \nabla(\Pi_h u - u) \cdot \mathbf{n}\} [\Pi_h u - u_h] \\ &\quad - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\alpha \nabla(\Pi_h u - u_h) \cdot \mathbf{n}\} [\Pi_h u - u] \\ &\quad + \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \alpha_{(i,-)} \frac{\delta}{h_i} [\Pi_h u - u] [\Pi_h u - u_h] =: A + B + \underbrace{C}_{=0} + \underbrace{D}_{=0}. \end{aligned}$$

It remains to estimate A and B . We have the following:

$$\begin{aligned}
|A| &= \left| \sum_{i=1}^n \int_{\Omega^{(i)}} \alpha_i \nabla(\Pi_h u - u) \cdot \nabla(\Pi_h u - u_h) \right| \\
&\leq \sum_{i=1}^n \alpha_i \|\nabla(\Pi_h u - u)\|_{L^2(\Omega^{(i)})} \|\nabla(\Pi_h u - u_h)\|_{L^2(\Omega^{(i)})} \\
&\leq \left(\sum_{i=1}^n \alpha_i \|\nabla(\Pi_h u - u)\|_{L^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \left(\sum_{i=1}^n \alpha_i \|\nabla(\Pi_h u - u_h)\|_{L^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \\
&\leq \|\Pi_h u - u\|_{DG} \|\Pi_h u - u_h\|_{DG} \\
&\stackrel{\text{Lemma 8}}{\leq} C_{\text{shape}} \left(\sum_{i=1}^n \alpha_i h_i^{2(s-1)} \|u_i\|_{H^s(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \|\Pi_h u - u_h\|_{DG}. \tag{3.6}
\end{aligned}$$

$$\begin{aligned}
|B| &= \left| \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\alpha \nabla(\Pi_h u - u) \cdot \mathbf{n}\} [\Pi_h u - u_h] \right| \\
&\leq \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \|\{\alpha \nabla(\Pi_h u - u)\}\|_{L^2(\Gamma^{(i,-)})} \|\Pi_h u - u_h\|_{L^2(\Gamma^{(i,-)})} \left(\frac{h_i}{\alpha_{(i,-)}} \right)^{\frac{1}{2}} \left(\frac{\alpha_{(i,-)}}{h_i} \right)^{\frac{1}{2}} \\
&\leq \left(\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \|\{\alpha \nabla(\Pi_h u - u)\}\|_{L^2(\Gamma^{(i,-)})}^2 \frac{h_i}{\alpha_{(i,-)}} \right)^{\frac{1}{2}} \\
&\quad \cdot \left(\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\alpha_{(i,-)}}{h_i} \|\Pi_h u - u_h\|_{L^2(\Gamma^{(i,-)})}^2 \right)^{\frac{1}{2}} \\
&\leq C \left(\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \|\{\alpha \nabla(\Pi_h u - u)\}\|_{L^2(\Gamma^{(i,-)})}^2 h_i \right)^{\frac{1}{2}} \|\Pi_h u - u_h\|_{DG}. \tag{3.7}
\end{aligned}$$

Now we have to estimate the last sum in (3.7).

$$\begin{aligned}
& \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} h_i \|\{\alpha \nabla(\Pi_h u - u)\}\|_{L^2(\Gamma^{(i,-)})}^2 \stackrel{z_h := \Pi_h u - u}{=} \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} h_i \alpha_i^2 \|\nabla z_h|_{\Omega^{(i)}}\|_{L^2(\Gamma^{(i,D)})}^2 \\
& \quad + \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} h_i \frac{\alpha_{(i,j)}^2}{4} \|\nabla z_h|_{\Omega^{(i)}} + \nabla z_h|_{\Omega^{(j)}}\|_{L^2(\Gamma^{(i,j)})}^2 \\
& \leq \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} h_i \alpha_i^2 h_i^{-1} (\|\nabla z_h\|_{L^2(\Omega^{(i)})}^2 + h_i^2 |\nabla z_h|_{H^1(\Omega^{(i)})}^2) \\
& \quad + \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} \frac{h_i \alpha_{(i,j)}^2 h_i^{-1}}{4} (\|\nabla z_h\|_{L^2(\Omega^{(i)})}^2 + \|\nabla z_h\|_{L^2(\Omega^{(j)})}^2 \\
& \quad + h_i^2 |\nabla z_h|_{H^1(\Omega^{(i)})}^2 + h_i^2 |\nabla z_h|_{H^1(\Omega^{(j)})}^2) \\
& \leq C \sum_{i=1}^n \alpha_i^2 (\|\nabla z_h\|_{L^2(\Omega^{(i)})}^2 + h_i^2 |\nabla z_h|_{H^1(\Omega^{(i)})}^2) \\
& \stackrel{\text{Lemma 7}}{\leq} C_{\text{shape}} \sum_{i=1}^n \alpha_i^2 (h_i^{2(s-1)} \|u\|_{H^s(\Omega^{(i)})}^2 + h_i^2 h_i^{2(s-2)} \|u\|_{H^s(\Omega^{(i)})}^2) \\
& \leq C_{\text{shape}} \sum_{i=1}^n \alpha_i^2 h_i^{2(s-1)} \|u\|_{H^s(\Omega^{(i)})}^2.
\end{aligned}$$

So, we have

$$a_{DG}(\Pi_h u - u, \Pi_h u - u_h) \leq C_{\text{shape}} \left(\sum_{i=1}^n \alpha_i^2 h_i^{2(s-1)} \|u_i\|_{H^s(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \|\Pi_h u - u_h\|_{DG}$$

and we obtain that

$$\|\Pi_h u - u_h\|_{DG} \leq \frac{C_{\text{shape}}}{D} \left(\sum_{i=1}^n \alpha_i^2 h_i^{2(s-1)} \|u_i\|_{H^s(\Omega^{(i)})}^2 \right)^{\frac{1}{2}}.$$

Because of this last inequality and of Lemma 7, (3.5) becomes

$$\|u - u_h\|_{DG} \leq C_{\text{shape}} \left(\sum_{i=1}^n \alpha_i^2 h_i^{2(s-1)} \|u_i\|_{H^s(\Omega^{(i)})}^2 \right)^{\frac{1}{2}}.$$

which is what we wanted to prove. \square

3.2 Error estimate in the L^2 norm

We consider the problem (2.8) in the particular case of $\alpha(x) = 1 \quad \forall x \in \Omega$ and $\Gamma_N = \emptyset$.

Theorem 3. Let u and u_h be the solutions of (2.8) and (2.9) respectively. For $s \geq 2$, the following bound holds:

$$\|u - u_h\|_{L^2(\Omega)} \leq E_{\text{shape}} \left(\max_{i=1}^n \{h_i\} \right)^{2(s-1)} \left(\sum_{i=1}^n \|u_i\|_{H^s(\Omega^{(i)})}^2 \right)^{\frac{1}{2}}, \quad (3.8)$$

where E_{shape} is independent of h_i and n .

Proof. In the following, we will apply the Aubin-Nitsche lift technique, which works well if the scheme is symmetric (see [13]).

We assume that the solution of the dual problem

$$\begin{cases} -\operatorname{div}(\nabla \phi) &= u - u_h & \Omega \\ \phi &= g_D & \Gamma_D \end{cases} \quad (3.9)$$

belongs to $H^2(\Omega)$ with continuous dependence on $u - u_h$:

$$\|\phi\|_{H^2(\Omega)} \leq C \|u - u_h\|_{L^2(\Omega)}. \quad (3.10)$$

Thus, we have

$$\|u - u_h\|_{L^2(\Omega)}^2 = \int_{\Omega} (u - u_h)^2 \stackrel{(3.9)}{=} - \int_{\Omega} (\operatorname{div} \nabla \phi)(u - u_h).$$

Denoting $e_h = u - u_h$ and integrating by parts on each element yields

$$\|e_h\|_{L^2(\Omega)}^2 = \sum_{i=1}^n \int_{\Omega^{(i)}} \nabla e_h \cdot \nabla \phi - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\nabla \phi \cdot \mathbf{n}\} [e_h]. \quad (3.11)$$

We now subtract the orthogonality equation

$$a_{DG}(u - u_h, v_h) = 0 \quad \forall v_h \in \mathcal{V}_h$$

from the equation (3.11):

$$\begin{aligned} \|e_h\|_{L^2(\Omega)}^2 &= \|e_h\|_{L^2(\Omega)}^2 - a_{DG}(e_h, v_h) \\ &= \sum_{i=1}^n \int_{\Omega^{(i)}} \nabla e_h \cdot \nabla (\phi - v_h) - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\nabla (\phi - v_h) \cdot \mathbf{n}\} [e_h] \\ &\quad + \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\nabla e_h \cdot \mathbf{n}\} [v_h] - \sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \frac{\delta}{h_i} \int_{\Gamma^{(i,-)}} [e_h] [v_h]. \end{aligned}$$

We choose $v_h = \Pi_h \phi$, a continuous interpolant of ϕ . We assume that such an interpolant exists. So we have that $[v_h] = 0$ and the last two terms on the right-hand side of the equation above vanish. Thus, we obtain

$$\|e_h\|_{L^2(\Omega)}^2 = \underbrace{\sum_{i=1}^n \int_{\Omega^{(i)}} \nabla e_h \cdot \nabla(\phi - \Pi_h \phi)}_{:=A} - \underbrace{\sum_{\Gamma^{(i,-)} \in \mathcal{E}_C \cup \mathcal{E}_D} \int_{\Gamma^{(i,-)}} \{\nabla(\phi - \Pi_h \phi) \cdot \mathbf{n}\} [e_h]}_{:=B}.$$

The first term is bounded using Cauchy-Schwarz's inequality and Lemma 7:

$$\begin{aligned} A &\leq \sum_{i=1}^n \|\nabla e_h\|_{L^2(\Omega^{(i)})} \|\nabla(\phi - \Pi_h \phi)\|_{L^2(\Omega^{(i)})} \\ &\leq \left(\sum_{i=1}^n \|\nabla e_h\|_{L^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \left(\sum_{i=1}^n \|\nabla(\phi - \Pi_h \phi)\|_{L^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \\ &\leq \|e_h\|_{DG} \left(\sum_{i=1}^n C_{\text{shape}} h_i^{2(s-1)} \|\phi_i\|_{H^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \\ &\stackrel{(3.10)}{\leq} \tilde{C}_{\text{shape}} \|e_h\|_{DG} \left(\max_{i=1}^n \{h_i\}^{s-1} \right) \|e_h\|_{L^2(\Omega)}. \end{aligned}$$

The second term is bounded by using the property of boundedness of the bilinear form:

$$\begin{aligned} B &\leq a_{DG}(\phi - \Pi_h \phi, e_h) \\ &\leq M \|\phi - \Pi_h \phi\|_{DG} \|e_h\|_{DG} \\ &\leq M \left(\sum_{i=1}^n C_{\text{shape}} h_i^{2(s-1)} \|\phi_i\|_{H^2(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \|e_h\|_{DG} \\ &\stackrel{(3.10)}{\leq} \tilde{C}_{\text{shape}} \|e_h\|_{DG} \left(\max_{i=1}^n \{h_i\}^{s-1} \right) \|e_h\|_{L^2(\Omega)}. \end{aligned}$$

Therefore,

$$\begin{aligned} \|e_h\|_{L^2(\Omega)} &\leq E_{\text{shape}} \left(\max_{i=1}^n \{h_i\} \right)^{s-1} \|e_h\|_{DG} \\ &\stackrel{\text{Theorem 2}}{\leq} E_{\text{shape}} \left(\max_{i=1}^n \{h_i\} \right)^{2(s-1)} \left(\sum_{i=1}^n \|u_i\|_{H^s(\Omega^{(i)})}^2 \right)^{\frac{1}{2}}. \end{aligned}$$

□

Chapter 4

Numerical results

In this chapter, we want to describe briefly the structure and the advantages of the built-in Matlab library GeoPDEs, in order to show some numerical results for the solution of the Poisson problem in the Continuous Galerkin case for single-patch and multi-patch domains. Then, we will outline the techniques for a Discontinuous Galerkin implementation, which, for reasons of time, we leave as a future work.

4.1 GeoPDEs

GeoPDEs is a software tool for research on Isogeometric Analysis of PDEs. It consists of a set of interrelated packages and it provides a flexible framework for implementing and testing new isogeometric methods in various application areas.

The main package, called `geopdes_base`, defines the basic data-structures and methods. Other packages deal with applications in linear elasticity, fluid mechanics and electromagnetism. There is also a specific package for multi-patch NURBS geometries and it is very useful for our purposes.

In this library, the computations for the geometry, the discrete basis functions and the matrices for the analysis are done separately. Moreover, all data needed for these computations is stored in independent structures, in such a

way that each part of the code can be modified without affecting the others.

For more information about the library GeoPDEs we suggest to read the paper [8], which explains in detail the main structures and the built-in functions to operate on them.

4.1.1 Main steps to write a code with GeoPDEs

In this paragraph we want to describe the main steps to write a code with the built-in functions and the structures within the library GeoPDEs.

There are three main structures that have to be built before starting to assemble the matrices and the vectors.

1. **The geometry structure.** The first step is to define the geometry of the physical domain. This is done by invoking the function `geo_load`, which takes as input the name of a file in Matlab binary format and gives as output a structure which contains the information to compute the geometry parametrization and its derivatives.
2. **The mesh structure.** The second step consists in defining the domain partition and to set the quadrature rule in each element, in order to compute the matrices and the right-hand side vector of the problem by numerical integration. The most simple possibility is to define a tensor product partition where the quadrature elements coincide with the knot spans in the geometry. For this purpose, we can use the function `msh_gauss_nodes` to compute the quadrature nodes and weights for a standard Gaussian quadrature rule.
3. **The space structure.** The most important structure of the implementation is the one which contains the information regarding the basis functions of the discrete space \mathcal{V}_h and their evaluation at the quadrature nodes, in order to numerically compute the integrals of the problem. Since the basis functions in IGA are locally supported, the integrals on each element of the partition are only computed for a reduced number

of basis functions. Similarly, only the values of some basis functions are stored on each element. A global numbering of the basis functions is introduced. Then, for each element of the quadrature partition, we give its connectivity, that is, the number associated to the basis functions whose support intersects the element.

Once the three basic data structures have been initialized, the next step is to assemble the stiffness matrix and the right-hand side of the linear system. GeoPDEs provides several functions that allow to compute these terms for different PDE problems. These functions all have a similar structure, which may be adapted with very little changes to handle different differential problems. There are also some built-in functions which allow the imposition of the boundary conditions.

4.2 Continuous Galerkin (CG) methods

We used the library GeoPDEs to solve the Poisson problem with a Continuous Galerkin method for some benchmark problems on two-dimensional and three-dimensional domains.

4.2.1 CG methods for a single-patch domain in 2D

First of all, we introduce a very simple benchmark problem with a single-patch domain.

We consider the Poisson problem (2.1) with $\alpha = 1$ and $\Gamma_N = \emptyset$. The domain $\Omega = \Omega^{(1)}$ is one quarter of a ring, with inner radius equal to 1 and outer radius equal to 2. We calculate the analytical expression of the functions f and g_D such that the exact solution in Ω of the problem is

$$u(x, y) = -(x^2 + y^2 - 1)(x^2 + y^2 - 4)xy^2.$$

We look for a numerical solution of the benchmark problem in the space of NURBS functions of degree 3 with an increasing number of subdivisions

and four quadrature nodes in each element of the patch. We represent the subdivision of the domain with different values of the element size h in Figure 4.1.

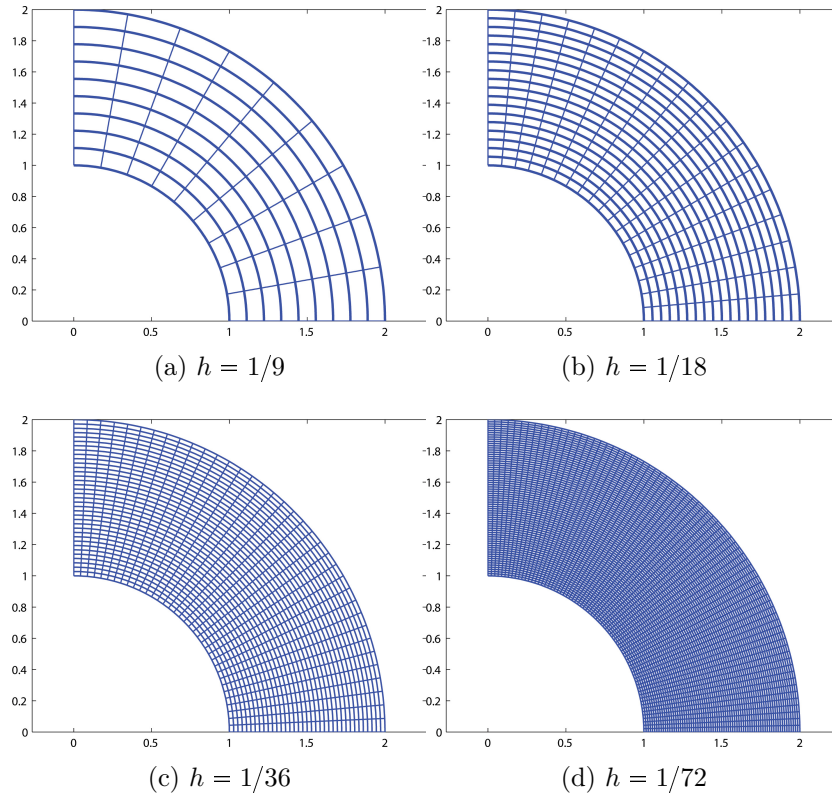


Figure 4.1: *Grid on the ring domain, with different values of the element size h .*

In Figure 4.2 we represent the exact solution and the numerical solution we found with a Matlab code in GeoPDEs.

Moreover, we study the convergence of the method in L^2 -norm and in H^1 -norm, using different values of h . The numerical values are in Table 4.1. For clarity's sake, in Figure 4.3 we represent the trend of the errors in both norms we are considering.

We can easily recognize from Figure 4.3 that the theoretical estimates for

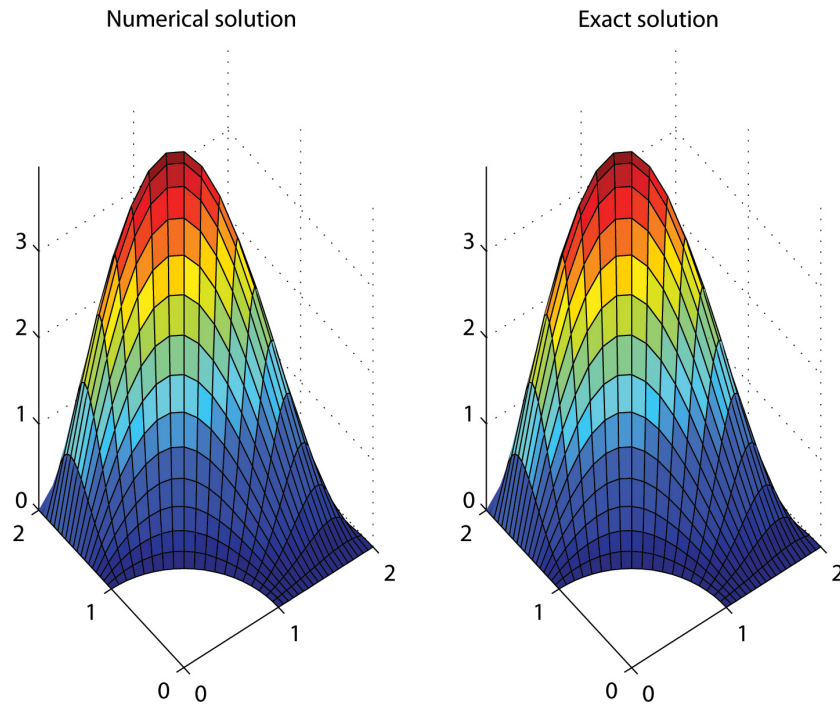


Figure 4.2: Comparison of exact solution and numerical solution.

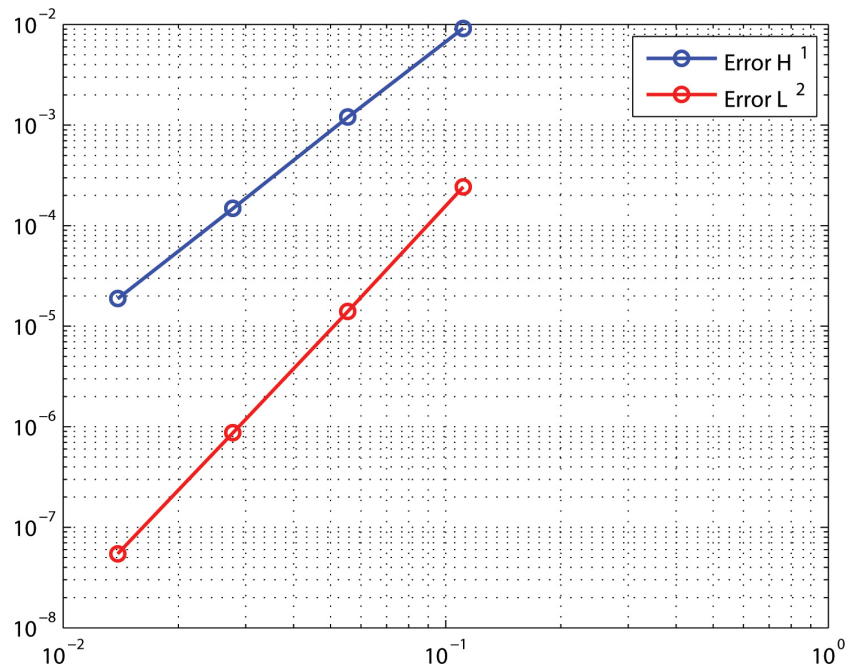


Figure 4.3: Trends of the errors in L^2 -norm and in H^1 -norm for the benchmark problem on a ring domain.

h	Error in L^2 -norm	Error in H^1 -norm
1/9	2.4274e-004	0.0092
1/18	1.4089e-005	0.0012
1/36	8.7255e-007	1.4849e-004
1/72	5.4632e-008	1.8746e-005

Table 4.1: *Errors in L^2 -norm and in H^1 -norm for the benchmark problem on a ring domain.*

the errors are verified, since

$$\begin{aligned} L^2 &\rightarrow O(h^4) \\ H^1 &\rightarrow O(h^3). \end{aligned}$$

4.2.2 CG methods for a single-patch domain in 3D

In this section, we show that it is also easily possible to solve Poisson problems with boundary conditions on a three-dimensional domain with the library GeoPDEs.

Namely, we consider a domain as in Figure 4.4, which reproduces a quarter of a thick ring.

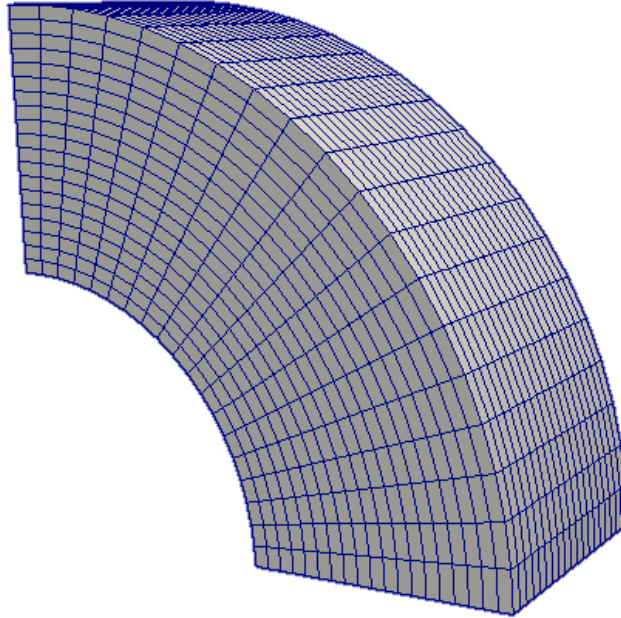
We assume that the exact solution of the benchmark problem is:

$$u(x, y, z) = e^x \sin(xy) \cos(z)$$

and we look for a numerical solution in the NURBS space of degree 2. As we have done in the preceding section, we let the element size h vary and we calculate the errors in H^1 - and in L^2 -norm. Then, we analyse the convergence of the numerical solution to the exact one in the two norms in Figure 4.5.

We note the estimated following trends:

$$\begin{aligned} L^2 &\rightarrow O(h^3) \\ H^1 &\rightarrow O(h^2). \end{aligned}$$

Figure 4.4: *Three-dimensional domain of a thick ring.*

h	Error in L^2 -norm	Error in H^1 -norm
1/4	0.1008	0.7770
1/8	0.0077	0.1422
1/16	7.2055e-004	0.0300
1/32	8.2405e-005	0.0072

Table 4.2: *Errors in L^2 -norm and in H^1 -norm for the benchmark problem on a thick ring domain.*

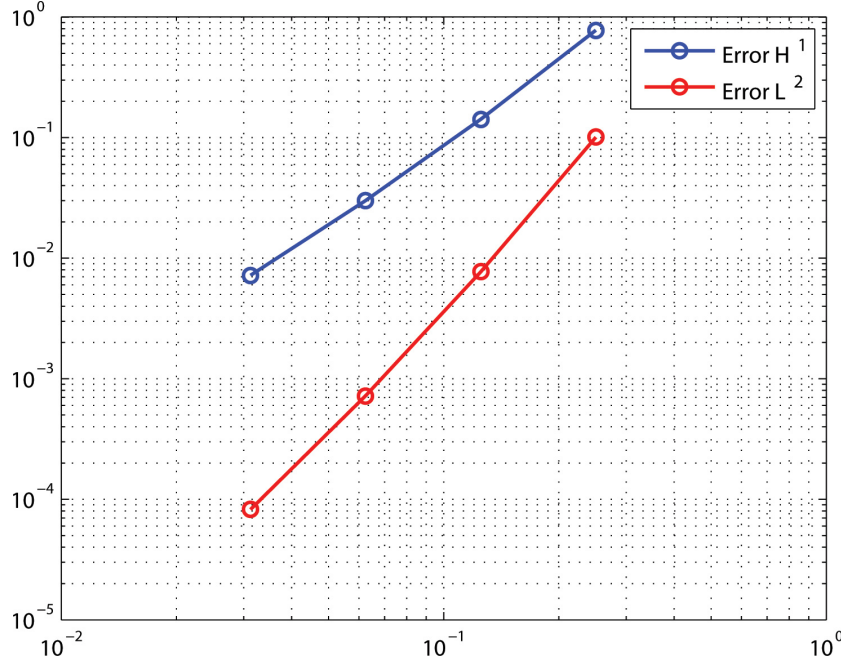


Figure 4.5: Trends of the errors in L^2 -norm and in H^1 -norm for the benchmark problem on a thick ring domain.

4.2.3 CG methods for a multi-patch domain in 2D

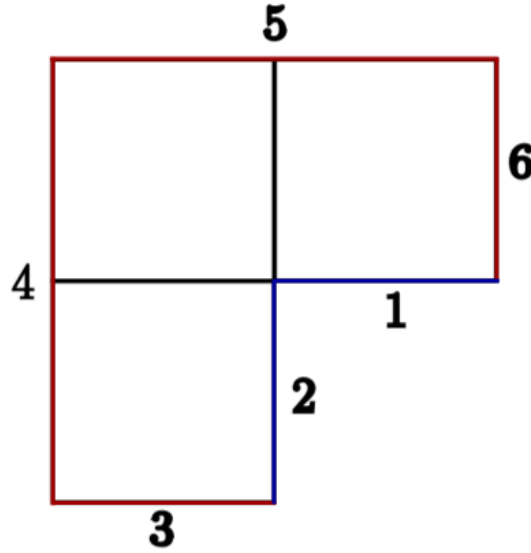
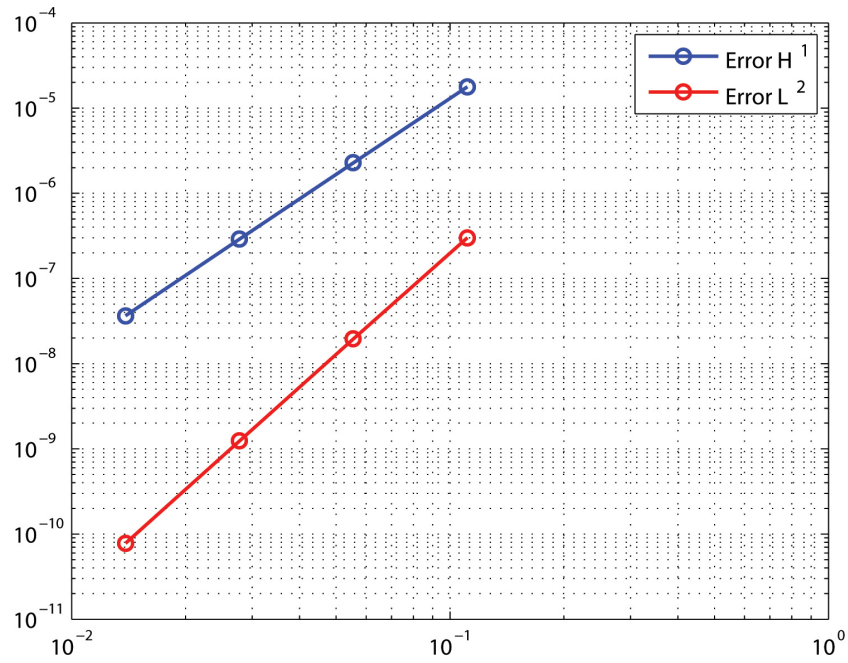
We consider now a benchmark problem with a multi-patch domain. Namely, we assume that the domain consists of three patches as in Figure 4.6, where $\Gamma_D = \{1, 2\}$ and $\Gamma_N = \{3, 4, 5, 6\}$.

The exact solution of the benchmark problem is given by

$$u(x, y) = e^x \sin(xy)$$

and we look for a solution in the NURBS space of degree 3.

We study the convergence of the method applied to this specific benchmark problem with the calculation of the errors in H^1 -norm and in L^2 -norm with different values of h (see Table 4.3). We represent the numerical errors of the Table 4.3 in Figure 4.7.

Figure 4.6: *L-shaped multi-patch domain.*Figure 4.7: *Trends of the errors in L^2 -norm and in H^1 -norm for the benchmark problem on a L-shaped domain.*

h	Error in L^2 -norm	Error in H^1 -norm
1/9	3.0064e-007	1.7752e-005
1/18	1.9452e-008	2.2797e-006
1/36	1.2372e-009	2.8919e-007
1/72	7.8004e-011	3.6428e-008

Table 4.3: *Errors in L^2 -norm and in H^1 -norm for the benchmark problem on a L -shaped domain.*

Therefore, we obtain the expected rates of convergence:

$$\begin{aligned} L^2 &\rightarrow O(h^4) \\ H^1 &\rightarrow O(h^3). \end{aligned}$$

4.3 Ideas for a DG implementation

Because of time constraints, we could not work extensively on the numerical results of the theory we developed in the previous chapters. Nevertheless, we would like to give some indications for a possible future work in this direction. In our opinion, the built-in library GeoPDEs is a useful tool that can be used as a starting point for the implementation of Discontinuous Galerkin methods. Therefore, we suggest to modify some built-in functions of this library in order to adapt them to the specific case of DG methods and to add new functions related to the jump and average terms using the geometry and data structures already available in GeoPDEs.

4.3.1 The linear system

First of all, it is useful to identify all the entries of the matrices and vectors involved in the discrete problem. We insert the following expressions

for u_h and v_h in the variational formulation of the discrete problem (2.9):

$$\begin{aligned} u_h &= \sum_{s=1}^n \sum_{\mathbf{l} \in \mathcal{R}^{(s)}} u_{\mathbf{l}}^{(s)} \hat{R}_{\mathbf{l}}^{(s)} \\ v_h &= \hat{R}_{\mathbf{k}}^{(t)} \quad \forall \mathbf{k} \in \mathcal{R}^{(t)}, \forall t = 1, \dots, n. \end{aligned}$$

The bilinear form and the linear functional can be rewritten as follows:

$$\begin{aligned} a_{DG}(u_h, v_h) &= \sum_{s=1}^n \sum_{\mathbf{l} \in \mathcal{R}^{(s)}} a_{DG}(\hat{R}_{\mathbf{l}}^{(s)}, \hat{R}_{\mathbf{k}}^{(t)}) u_{\mathbf{l}}^{(s)} \\ \langle F, v_h \rangle &= \langle F, \hat{R}_{\mathbf{k}}^{(t)} \rangle, \end{aligned}$$

$\forall \mathbf{k} \in \mathcal{R}^{(t)}$ and $\forall t = 1, \dots, n$. We can write explicitly every term of the previous expressions.

$$\begin{aligned} a_{DG}(\hat{R}_{\mathbf{l}}^{(s)}, \hat{R}_{\mathbf{k}}^{(t)}) &= \sum_{i=1}^n a_i(\hat{R}_{\mathbf{l}}^{(s)}, \hat{R}_{\mathbf{k}}^{(t)}) + b(\hat{R}_{\mathbf{l}}^{(s)}, \hat{R}_{\mathbf{k}}^{(t)}) + c(\hat{R}_{\mathbf{l}}^{(s)}, \hat{R}_{\mathbf{k}}^{(t)}) \\ &:= \sum_{i=1}^n A_{\mathbf{l}\mathbf{k}}^{ss} + \left(\sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} B_{\mathbf{l}\mathbf{k}}^{st} + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} B_{\mathbf{l}\mathbf{k}}^{sD} \right) + \left(\sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} C_{\mathbf{l}\mathbf{k}}^{st} + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} C_{\mathbf{l}\mathbf{k}}^{sD} \right), \\ \langle F, \hat{R}_{\mathbf{k}}^{(t)} \rangle &:= \mathbf{f}_{\mathbf{k}}^t, \end{aligned}$$

where

$$\begin{aligned} A_{\mathbf{l}\mathbf{k}}^{ss} &= \int_{\Omega^{(i)}} \alpha_i \nabla \hat{R}_{\mathbf{l}}^{(s)} \nabla \hat{R}_{\mathbf{k}}^{(s)} \\ B_{\mathbf{l}\mathbf{k}}^{st} &= - \int_{\Gamma^{(i,j)}} \left(\{ \alpha \nabla \hat{R}_{\mathbf{l}}^{(s)} \cdot \mathbf{n} \} [\hat{R}_{\mathbf{k}}^{(t)}] + \{ \alpha \nabla \hat{R}_{\mathbf{k}}^{(t)} \cdot \mathbf{n} \} [\hat{R}_{\mathbf{l}}^{(s)}] \right) \\ B_{\mathbf{l}\mathbf{k}}^{sD} &= - \int_{\Gamma^{(i,D)}} (\alpha_i \nabla \hat{R}_{\mathbf{l}}^{(s)} \cdot \mathbf{n}_i) \hat{R}_{\mathbf{k}}^{(t)} + (\alpha_i \nabla \hat{R}_{\mathbf{k}}^{(t)} \cdot \mathbf{n}_i) \hat{R}_{\mathbf{l}}^{(s)} \\ C_{\mathbf{l}\mathbf{k}}^{st} &= \alpha_{(i,j)} \delta^{(i,j)} \int_{\Gamma^{(i,j)}} [\hat{R}_{\mathbf{l}}^{(s)}] [\hat{R}_{\mathbf{k}}^{(t)}] \\ C_{\mathbf{l}\mathbf{k}}^{sD} &= \alpha_{(i,D)} \delta^{(i,D)} \int_{\Gamma^{(i,D)}} \hat{R}_{\mathbf{l}}^{(s)} \hat{R}_{\mathbf{k}}^{(t)} \\ \mathbf{f}_{\mathbf{k}}^t &= \int_{\Omega} f \hat{R}_{\mathbf{k}}^{(t)} + \sum_{\Gamma^{(i,N)} \in \mathcal{E}_N} \int_{\Gamma^{(i,N)}} g_N \hat{R}_{\mathbf{k}}^{(t)} \\ &\quad + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} \int_{\Gamma^{(i,D)}} (-\alpha_i \nabla \hat{R}_{\mathbf{k}}^{(t)} \cdot \mathbf{n}_i + \alpha_i \delta^{(i,D)} \hat{R}_{\mathbf{k}}^{(t)}) g_D. \end{aligned}$$

Note that on $\Gamma^{(i,j)}$ we assume the following:

$$\begin{aligned}\{\alpha \nabla \hat{R}_1^{(s)} \cdot \mathbf{n}\} &= \frac{1}{2} \left(\alpha_i (\nabla \hat{R}_1^{(s)})_{|\Omega^i} \cdot \mathbf{n}_i + \alpha_j (\nabla \hat{R}_1^{(s)})_{|\Omega^{(j)}} \cdot \mathbf{n}_i \right) \\ [\hat{R}_1^{(s)}] &= (\hat{R}_1^{(s)})_{|\Omega^{(i)}} - (\hat{R}_1^{(s)})_{|\Omega^{(j)}}.\end{aligned}$$

Therefore, we can say that, for **each** patch $\Omega^{(s)}$, we have to solve the following linear system:

$$\sum_{\mathbf{l} \in \mathcal{R}^{(s)}} \left[A_{\mathbf{l}\mathbf{k}}^{ss} + \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} B_{\mathbf{l}\mathbf{k}}^{st} + \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} B_{\mathbf{l}\mathbf{k}}^{sD} + \sum_{\Gamma^{(i,j)} \in \mathcal{E}_C} C_{\mathbf{l}\mathbf{k}}^{st} \sum_{\Gamma^{(i,D)} \in \mathcal{E}_D} C_{\mathbf{l}\mathbf{k}}^{sD} \right] u_1^{(s)} = \mathbf{f}_{\mathbf{k}}^t,$$

for all $\mathbf{k} \in \mathcal{R}^{(t)}$ and for all $t = 1, \dots, n$.

The implementation of this system is quite complex and a particular care with the jump and average terms is needed. Two tricky steps are the choice of the numbering of the shape functions that are non-zero along the interfaces and the implementation of the gradients of the shape functions on the boundaries of the patches.

4.3.2 Pseudocode for DG methods

In the following, we write the pseudocode for the resolution of the Poisson problem on a single-patch domain. In this case, all the terms related to jumps or averages are zero. Note that some built-in functions can be easily used for the terms which appear also in CG methods. The remaining terms, specific of DG variational formulation, can be evaluated with some built-in functions with the proper entries or with some functions that still have to be implemented (as `op_gradv_n` and `op_gradu_n_v`). Therefore, a first step of the implementation of DG methods on single-patch domains consists of building this missing functions.

With the two lines

```
stiff_mat = op_gradu_gradv_tp (sp,sp,msh,c_diff_fun)
rhs = op_f_v_tp (sp, msh, f)
```

we can evaluate, respectively, the stiffness matrix and the right-hand-side of the classical variational formulation:

$$\int_{\Omega} \nabla u \cdot \nabla v \text{ and } \int_{\Omega} f v.$$

With the loop over the Neumann boundary sides,

```
for iref = nmnn_sides
msh_side, sp_side, g_n_val
  rhs_nmnn = op_f_v (sp_side, msh_side, g_n_val);
  rhs(sp_side.dofs) = rhs(sp_side.dofs) + rhs_nmnn;
end
```

it is possible to evaluate the boundary integrals

$$\sum_{\Gamma^{(i,N)} \in \mathcal{E}_N} \int_{\Gamma^{(i,N)}} g_N v.$$

With the loop over the Dirichlet boundary side, all the remaining boundary integrals, except the ones over \mathcal{E}_C , can be evaluated and assembled in the stiffness matrix and in the right-hand-side.

```
for iref = drchlt_sides
msh_side, sp_side, g_d_val
  rhs_drchlt_1 = c_diff*delta/h.*op_f_v ( sp_side,
msh_side, g_d_val );
  rhs_drchlt_2 = -c_diff.*op_gradv_n ( sp_side,
msh_side, g_d_val );
  lhs_drchlt_1 = -c_diff.*op_gradu_n_v ( sp_side,
    sp_side, msh_side );
  lhs_drchlt_2 = delta/h.*op_u_v (sp_side, sp_side,
    msh_side, c_diff_fun( x, y ));

  global_dofs = gnum{iptc}(sp_side.dofs);
  stiff_mat(global_dofs,global_dofs) = stiff_mat
```

```
(global_dofs,global_dofs)+ lhs_drchlt_1 + ...  
    (lhs_drchlt_1)' + lhs_drchlt_2 ;  
rhs(global_dofs) = rhs(global_dofs)  
    + rhs_drchlt_1 + rhs_drchlt_2;  
end  
  
u = stiff_mat \ rhs;
```

The implementation of DG methods on multi-patch domains requires also the evaluation of the terms related to jumps and averages of functions. This is the trickiest step, since it is necessary to pay attention to the numbering of the shape functions in order to match correctly the corresponding values of the functions along the interfaces. For this purpose, it will be particularly useful the assumption of fully matching subdomains we made in Chapter 2.

Chapter 5

Conclusions

This Thesis consists in two parts: a theoretical and a numerical one.

In the first part we introduced the main ideas and concepts of Isogeometric Analysis and we used this new method of representing complex geometries in order to solve partial differential equations, namely the Poisson problem with Dirichlet and Neumann boundary conditions. We deduced the variational formulation of this problem and we verified the hypothesis of Lax Milgram's Lemma (the main steps are the proofs of the boundedness and the coercivity of the bilinear form). Therefore, we could state existence and uniqueness of the solution of the problem and continuous dependence from the data. Finally, we obtained the error estimates in the L^2 -norm and in the DG -norm, which we summarize in the following theorem.

Theorem 4 (Error estimates in L^2 - and in DG -norm). Let u and u_h be the solutions of (2.8) and (2.9), respectively. For $s \geq 2$, the following two bounds hold:

$$\begin{aligned} \|u - u_h\|_{DG}^2 &\leq D_{\text{shape}} \sum_{i=1}^n \alpha_i^2 h_i^{2(s-1)} \|u_i\|_{H^s(\Omega^{(i)})}^2 \\ \|u - u_h\|_{L^2(\Omega)} &\leq E_{\text{shape}} \left(\max_{i=1}^n \{h_i\} \right)^{2(s-1)} \left(\sum_{i=1}^n \|u_i\|_{H^s(\Omega^{(i)})}^2 \right)^{\frac{1}{2}} \end{aligned}$$

where D_{shape} is independent of h_i , n and the jumps of α_i and E_{shape} is independent of h_i and n .

In the second part, some numerical tests with CG methods have been discussed with a benchmark problem for a single-patch domain in 2D (a quarter of a ring) and in 3D (a quarter of a thick ring) and for a multi-patch domain (a L-shaped domain with three patches). We verified the theoretical estimates of the CG methods in IGA with a study of the convergence for all the benchmark problems. Moreover, we gave some suggestions and ideas about how to implement a DG code with the help of the built-in Matlab library GeoPDEs, which is, in our opinion, a very useful tool to approach this kind of issues. For this purpose, we wrote a pseudocode for a DG-method on a single-patch domain.

As a future work, an implementation of the DG method in Isogeometric Analysis could be interesting and useful for many industrial application, since it is more flexible and it seems more efficient than CG methods in Finite Element analysis. Once a complete implementation of this code has been reached, all the error estimates, that we have formulated as a theoretical result, can be tested and verified. Moreover, it could be particularly useful the development of specific DG solvers to reduce cost and running time of the implementation of these methods.

With DG methods, models of heat diffusion on domains with different materials can be described more in detail and, in general, all the problems with solutions with steep gradients along the interfaces can be implemented with some programming work. Finally, we hope that this Thesis can be a useful starting point for those who are interested in implementing a code with DG methods.

Bibliography

- [1] ARNOLD, D., BREZZI, F., COCKBURN, B. e MARINI, L. (2002), Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems, *SIAM Journal of Numerical Analysis*, vol. 39, p. 1749–1779.
- [2] AURICCHIO, F., BEIRAO DA VEIGA, L., BUFFA, A., LOVADINA, C., REALI, A. e SANGALLI, G. (2007), A fully "locking-free" isogeometric approach for plane linear elasticity problems: A stream function formulation, *Computer Methods in Applied Mechanics and Engineering*, vol. 197, p. 160–172.
- [3] BAZILEVS, Y., BEIRAO DA VEIGA, L., COTTRELL, J., HUGHES, T. e SANGALLI, G. (2006), Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes, *Math. Models Methods Appl. Sci.*, vol. 16, p. 1031–1090.
- [4] BEIRAO DA VEIGA, L., CHO, D., PAVARINO, L. e SCACCHI, S. (2012), BDDC preconditioners for Isogeometric Analysis, *To appear in Math. Mod. Meth. Appl. Sci.*
- [5] BEIRAO DA VEIGA, L., CHO, D., PAVARINO, L. e SCACCHI, S. (2012), Overlapping Schwarz methods for Isogeometric Analysis, *Siam J. Numer. Anal.*, vol. 50(3), p. 1394–1416.
- [6] BREZZI, F., COCKBURN, B., MARINI, L. e SULI, E. (2006), Stabilization mechanisms in discontinuous Galerkin finite element methods,

- Computer Methods in Applied Mechanics and Engineering*, vol. 195, p. 3293–3310.
- [7] COTTRELL, J., HUGHES, T. e BAZILEVS, Y. (2009), *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley and Sons.
- [8] DE FALCO, C., REALI, A. e VÁSQUEZ, R. (2011), GeoPDEs: a research tool for Isogeometric Analysis of PDEs, *Adv. Eng. Softw.*, vol. 42(12), p. 1020–1034.
- [9] DRYJA, M. (2003), On Discontinuous Galerkin Methods for Elliptic Problems with Discontinuous Coefficients, *Computational Methods in Applied Mathematics*, vol. 3, p. 76–85.
- [10] GAHALAUT, K. P. S., KRAUS, J. e TOMAR, S. K. (2012), Multi-grid Methods for Isogeometric Discretization, *Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences*.
- [11] HUGHES, T., COTTRELL, J. e BAZILEVS, Y. (2005), Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer Methods in applied mechanics and engineering*, vol. 194, p. 4135–4195.
- [12] KLEISS, S., PECHSTEIN, C., JUETTLER, B. e TOMAR, S. (2012), IETI - Isogeometric Tearing and Interconnecting, *To appear in Comput. Methods Appl. Mech. Engrg.*
- [13] RIVIÈRE, B. (2008), *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations - Theory and Implementation*.
- [14] TOSELLI, A. e WIDLUND, O. (2005), *Domain Decomposition Methods - Algorithms and Theory*, Springer.