EPFL - MSc in Computational Science and Engineering
Politecnico di Milano - MSc in Mathematical Engineering

Master Thesis

A Numerical Study of the Isogeometric Collocation Method

Supervisors:
Prof. Alfio Quarteroni
Prof. Luca Dedè

Student:
Ondine Chanon
ondine.chanon@epfl.ch
Student n°: 214211

Academic year 2016-2017
Submitted on June 23rd, 2017

Abstract

In this master project, we explore isogeometric collocation methods based on B-splines to solve partial differential equations. More specifically, we concentrate our work on one-dimensional problems of first and second order and we analyse the convergence of the error under h-refinement using several kind of collocation methods. Our analysis is motivated by the search of an isogeometric collocation method that converges with the same rate of convergence as the Galerkin isogeometric analysis.

Two different families of isogeometric collocation methods are explored. First, we mimic the equivalence between the Galerkin spectral element method with numerical integration and its collocation counterpart, using the isogeometric paradigm. However, we will show that this equivalence is still missing for the isogeometric analysis. Then, inspired from the Gauss-Lobatto Lagrange extraction of B-splines [Nguyen and Schillinger, 2017], a B-spline basis that is interpolatory at the Gauss-Legendre-Lobatto nodes is built, and collocation at some subsets of such nodes is studied. Subsets of Gauss-Legendre-Lobatto nodes are chosen so that the collocation methods derived from them converge with the best possible rate under h-refinement and so that the condition number of the collocation matrix is minimal. A good convergence of the error is obtained, however it is still not optimal: when the B-spline order p is even, the L^2- and the H^1-errors behave asymptotically as h^p where h is the mesh size; when p is odd, both errors behave asymptotically as h^{p-1}.

Acknowledgments

I would like to sincerely thank Professors Alfio Quarteroni and Luca Dedè for the opportunity to work on this project within the Chair of Modeling and Scientific Computing (CMCS) of the EPFL, and in collaboration with the Politecnico di Milano. Thank you for sharing your scientific expertise, for your help, your availability and your support.

I would also like to express my gratitude to MER Simone Deparis to have initiated and allowed me to take part in the double degree program between the EPFL and the Politecnico di Milano, and to have always helped me and supported me in front of the administrative difficulties.

Grazie a tutte le persone del gruppo CMCS e, in particolare, a Luca e a Stefano con cui ho condiviso l'ufficio, i momenti di disperazione e i momenti felici di questa tesi.

Un ringraziamento molto speciale ai miei cinque fedeli compagni di doppia laurea, in particolare Ste, e a Davide, entrato nel gruppo anche lui, senza i quali sarebbe stato molto difficile resistere fino ad oggi. Il vostro sorriso mi ha aiutata in ogni singolo giorno! Grazie anche a Laura e a Francesco, i miei migliori professori d'italiano, che mi hanno accolta a Milano con tantissima gentilezza. Merci aussi à ma petite Mél, avec qui j'ai partagé ces beaux moments italiens ainsi que tant de souvenirs, and thank you to the whole CLV group from Waterloo, for our wonderful Canadian dream.

Un grand merci à Lola, à Max et à PB, pour ces six années d'amitié, de soutient, de sourires et de folie! Notre petit groupe m'a apporté courage, persévérance et motivation.

Je voudrais aussi remercier de tout mon coeur ma famille, mes petits parents, ma Bibi et mon Titi, pour tout ce qu'ils ont fait pour moi depuis tant d'années, pour leur amour inconditionnel, leur écoute et leur soutient. Je dédie ce travail à mon papi et à ma petite grand-mère, qui sont tous les deux allés rejoindre les anges cette année.

E infine, grazie tantissimo a te, Mimò, per il tuo sorriso, il tuo sostegno in ogni istante, e anche semplicemente per la tua presenza al mio fianco.

Contents

List of Figures

List of Tables

Notation

We describe some notation and abbreviations that will be used throughout this document.

$\mathbb{P}_k(\Omega)$	Set of polynomials of total degree at most k, in Ω
$\mathbb{Q}_k(\Omega)$	Set of polynomials of degree at most k in each variable, in Ω
$\mathbb{P}_k^0(\Omega)$	Set of polynomials belonging to $\mathbb{P}_k(\Omega)$ that are 0 on $\partial\Omega$
$\mathbb{Q}_k^0(\Omega)$	Set of polynomials belonging to $\mathbb{Q}_k(\Omega)$ that are 0 on $\partial\Omega$
$f\|_\Omega$	Restriction of a function f to the space Ω
$k(A)$	Condition number of matrix A
$\#S$	Cardinality of a set S
$\mathbb{N}$	$\{0, 1, 2, \ldots\}$, the set containing the positive integers and zero
$\mathbb{B}_p^c(\Omega)$	Set of B-splines, piecewise polynomial of degree k on Ω and C^c-continuous between each element, such that $0 \leq c \leq p-1$
$f(x^+)$	The limit $\lim_{\xi \underset{>}{\to} x} f(\xi)$
$f(x^-)$	The limit $\lim_{\xi \underset{<}{\to} x} f(\xi)$
IGA	Isogeometric analysis
FEM	Finite element method
SEM	Spectral element method
SEM G-NI	Galerkin spectral element method with numerical integration
GLL	Gauss-Legendre-Lobatto
NURBS	Non-uniform rational B-spline
PDE	Partial differential equation

Chapter 1

Introduction

Many physical and Engineering fields, such as Mechanics, Biology or Chemistry among many others, are mostly described by a strong mathematical tool: the partial differential equations (PDE). Those equations can be very complex; one can think for example of the Navier-Stokes equations that describe the fluid dynamics. They can be so complex that mathematicians have not found a way to solve them analytically yet, and it is from this observation that computational mathematics and numerical approximation have been developed. The growing importance of computers and computer science in the last decades, together with the development of more powerful computing machines, have helped and fostered the development of efficient numerical and computational methods. Nowadays, scientific computing has become essential, and the most used and widely spread numerical method is the Finite Element Method [26], applied in most fields with success.

However, the performance of the Finite Element Method in many problems comes with an important drawback which shows the limit between the real world of application problems and the numerical tools. Indeed, the Finite Element Method requires the generation of a mesh on top of the geometrical domain on which the differential problem is defined. That is, even if Engineers and designers use an exact representation of the geometries they are working with, thanks to Computed Aided Design (CAD) tools for example, an additional step of geometry reconstruction by meshing is required in order to use the Finite Element Method. Not only does it approximate the geometrical domain instead of considering the exact one, and this can lead to accuracy problems, but it also requires a lot of additional computational time. More specifically, it can take up to 80% of the total computational time [10], including the time to solve the numerical problem, and an automatic mesh generation can only be done in some particular cases.

In order to overcome this difficulty, another numerical method has been developed in the last years, firstly introduced by Hughes, Cottrell and Bazilevs, and called Isogeometric Analysis [17]. The first idea was to develop a method that always preserves the geometry exactly, no matter how coarse the mesh, and that simplifies the refinement of the mesh. To satisfy this requirement, Isogeometric Analysis (IGA) has been directly designed from the basic components of any CAD: the B-splines, or more generally, the Non-Uniform Rational B-Splines (NURBS). To introduce briefly this method, IGA shares many properties with the Finite Element Method, but instead of piecewise polynomial functions, the basis functions of IGA are B-splines (or NURBS), that is, the exact same functions from which CAD geometries are built. Consequently, the basis functions used in IGA highly depend on the representation of the geometrical domain, and this fact is called isogeometric concept, hence the name Isogeometric Analysis.

IGA has given promising results on diverse applications such as in fluid mechanics [4, 5], wave propagation problems [13] or even in electromagnetism [6]. Moreover, comparisons of IGA with the classical Finite Element or Spectral Element methods have been performed [9, 18], and IGA shows superior approximation properties with respect to the total number of degrees of freedom while allowing more regular approximation. That is, the error obtained with IGA converges faster than the one obtained with the Finite Element or the Spectral Element method, when the number of degrees of freedom is fixed but the underlying polynomial degree or B-spline order is increased.

The most common way to solve a PDE is by using the Galerkin method, that is, the differential problem is formulated and treated in a weak sense: the whole equation is first multiplied by a test function and integrated over the computational domain, then the solution is sought in a suitable discrete space [26]. An alternative way to proceed is to work directly on the strong formulation of the PDE, and this can lead to the so-called collocation method. In this case, basis functions need to be sufficiently regular to be fed into the considered differential operator, but the advantage of collocation methods is to be cheaper with respect to the formation of the problem and the matrix assembly. IGA collocation methods have been first introduced by Auricchio and al. in [2] where the theoretical basis and a first insight on simple problems are given.

Nevertheless, isogeometric collocation methods proposed so far do not perform as well as the Galerkin isogeometric method, more specifically in terms of rate of convergence [1, 23, 24]. An interesting paper from Gomez and De Lorenzis [15] show however that there must exist a set of collocation points that reproduces exactly the Galerkin solution, and therefore such that the error of the collocation method built from those points has the same rate of convergence as the Galerkin isogeometric method. Such collocation points, called the Cauchy-Galerkin points and leading to the so-called variational collocation method, are determined thanks to a simple application of the mean value theorem of integral calculus [16] on the Galerkin formulation of the problem, but they highly depend on the Galerkin solution itself. Therefore, the variational collocation method cannot be used when the Galerkin solution is not known, so that attempts have been done to approximate the Cauchy-Galerkin points without knowing the Galerkin solution [1, 15]. But again, this does not always lead to an optimal isogeometric collocation method, that is, the error obtained with the Galerkin isogeometric method still converges in some cases with a greater rate of convergence.

Consequently, this project aims at analyzing and providing an insight into isogeometric collocation methods. Therefore, we explore the literature about collocation methods and select ideas from which we develop new isogeometric collocation methods. More precisely, this report is organized in the following way:

- After this first introductory chapter, we present in more details the isogeometric analysis in Chapter 2, and in particular the concept of isogeometric collocation method. Isogeometric collocation methods at Greville or Demko abscissae are presented as the most widely used such methods found in literature [14, 17].

- Then, it is now well known that the Galerkin Spectral Element Method with Numerical Integration (SEM G-NI) is equivalent to the spectral collocation method in which the collocation points are the Gauss-Legendre-Lobatto nodes on each element [8]. Those nodes are in particular the optimal quadrature nodes used in the numerical integration of the weak Galerkin spectral element problem. Therefore, in Chapter 3, an attempt to imitate this equivalence in the isogeometric framework is given. In particular, we find quadrature formulas that allow the exact integration of the mass and the stiffness matrices obtained when using the Galerkin isogeometric method on a second order differential problem. We hope to find suitable collocation points from the quadrature nodes, but instead we show that this is actually not possible to obtain.

- In Chapter 4, we develop an isogeometric collocation method based on the paper from Nguyen and Schillinger [24]. In this article, an extraction operator is introduced, linking Gauss-Lobatto Lagrange functions with B-splines. This transformation operator is used in order to define a basis of B-splines, still strongly dependent on the geometrical domain, but whose functions are interpolatory at the Gauss-Legendre-Lobatto nodes. We hope that a subset of the Gauss-Legendre-Lobatto nodes will be a good set of collocation sites when this basis is used, that leads to a well convergent isogeometric collocation method. It will indeed lead to a convergent method, but it will not always be optimal.

- Finally, we draw our conclusions in the last Chapter 5, together with some future possible developments and perspectives.

Finally, the scope of our research is the following:

- Only one dimensional problems will be considered since in higher dimensions, collocation points are defined by means of tensor product rules (see Chapter 2).

- Only B-spline functions and geometries are considered. The generalization to NURBS should be done in a complementary work but should not be complicated since all properties of B-splines are inherited to NURBS (see Chapter 2).

- Work is concentrated on B-spline spaces whose function have the same continuity between all elements. More precisely, we concentrate on spaces $\mathbb{B}_p^{p-1}$ where $p \in \mathbb{N} \setminus \{0\}$, that is, on spaces of B-splines of order p that are globally C^{p-1}-continuous, i.e. that are C^{p-1}-continuous between each pair of elements. Those are indeed the most widely used B-spline spaces found so far in the literature.

Chapter 2

Isogeometric Analysis

Domains of practical interest in which we aim at solving partial differential equations are often represented by B-splines, NURBS (Non-Uniform Rational B-Splines) or T-splines. Most of computed aided design (CAD) softwares are also based on those types of curves and can represent exactly the geometries that Engineers and designers want to reproduce. Isogeometric analysis (IGA) makes a link between CAD geometries and the classical finite element methods (FEM). Indeed, the FEM approach usually requires to build a mesh on top of the already existing geometry, and solves the problem on this mesh. This may affect the error of the numerical solution, and it can be computationally very expensive. Instead, in the IGA approach, both the underlying geometry and the space in which lies the numerical solution are represented by the same basis functions that are composed of B-splines or NURBS: a lot of computational time is saved and the solution is no more affected by the approximated geometry.

In this chapter, we introduce briefly what B-splines and NURBS are, the isogeometric analysis, and how it can be used to solve differential equations as a Galerkin or a collocation method. The structure of this introduction to IGA is inspired from [10].

2.1 B-splines and NURBS

2.1.1 Parametric space and knot vectors

In order to define B-splines and NURBS, we first need a parametric space. Recall that in finite element analysis, each element is represented in a parametric space and has its own mapping into the physical geometry. Unlike it, in the isogeometric analysis framework, sets of multiple elements are considered, called *patches* and representing subsets of the full considered geometry; each patch has its own mapping from a parametric space to the physical geometry. Patches are chosen so that they fulfill uniformity properties, from the point of view of element types or models used. In the following, we will always consider a single patch geometry.

B-splines are piecewise polynomial functions with high global regularity that compose a finite dimensional function space and that can then be defined from a finite basis. Let n be the dimension of this space, and p be the degree of the piecewise underlying polynomials. p is more generally called the *order* of the B-spline. Thanks to the parametric space considered, the B-spline basis is defined through the concept of knot vector introduced in the following definition. Let first consider a one-dimensional parametric space $\hat{\Omega}$.

Definition 2.1.1 (Knot vector) A finite subset of $\hat{\Omega}$ made of non-decreasing real numbers representing coordinates in this parametric space $\hat{\Omega}$ is called *knot vector* and is written

$$\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\} \subset \hat{\Omega}.$$

Each real ξ_i, $i = 1, \dots, n+p+1$ is a *knot*, hence the name knot vector.

Knots in a knot vector are not necessarily distinct, so we can define the multiplicity of a knot:

Definition 2.1.2 (Multiplicity of a knot) A knot has *multiplicity* k, $k \in \mathbb{N} \setminus \{0\}$, if it is repeated k times in the knot vector.

We will see that the multiplicity of each knot has important consequences in the properties of the B-spline basis defined from the knot vector considered. We now introduce the notion of knot span:

Definition 2.1.3 (Knot span) Each interval $[\xi_i, \xi_{i+1}]$ for $i = 1, \dots, n+p$ is called *knot span.*

Note that the knots in a knot vector are not necessarily distinct, so that a knot span can have length zero. Moreover, the interval $[\xi_1, \xi_{n+p+1}]$ contains $n+p$ knot spans that are not necessarily distinct neither. This notion of knot span should not be mixed with the notion of *element*:

Definition 2.1.4 (Element) Each internal knot span, that is each $[\xi_i, \xi_{i+1}]$ such that $\xi_{i+1} \neq \xi_1$ and $\xi_i \neq \xi_{n+p+1}$, $i = 1, \dots, n+p$, is called *element.*

This notion of element can be compared with the notion of element found in finite element analysis, but it will play a slight different role in IGA and one has to be careful not to mix them up. Note moreover that the case of a zero-length element is not excluded. In the following, only uniform open knot vectors will be considered:

Definition 2.1.5 (Uniform and open knot vector) A knot vector is said to be *uniform* if its distinct knots are uniformly spaced in $[\xi_1, \xi_{n+p+1}]$, that is, if they are equally distributed. It is said to be *open* if its first and last knots have multiplicity $p+1$, that is if $\xi_1 = \ldots = \xi_{p+1}$ and $\xi_{n+1}, \dots, \xi_{n+p+1}$.

We have now all the ingredients to be able to define the B-spline and the NURBS basis functions.

2.1.2 B-spline and NURBS basis functions

Let us first define univariate B-spline basis functions, that is B-spline basis functions that are defined from a one-dimensional parametric space $\hat{\Omega}$. As before, let p be the order of the B-splines considered, and n be the dimension of the B-spline space.

Definition 2.1.6 (Univariate B-spline basis functions) Let $\Xi = \{\xi_1, \dots, \xi_{n+p+1}\}$ be a knot vector of a parametric space $\hat{\Omega}$. Then, the n *univariate B-spline basis functions* of order p, say $\{N_{i,p} : \hat{\Omega} \to \mathbb{R}\}_{i=1}^n$, are recursively defined on the order by: $\forall i \in \{1, \dots, n\}$,

$$\begin{cases} N_{i,0}(\xi) = \begin{cases} 1 \text{ if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 \text{ otherwise,} \end{cases} \\ N_{i,k}(\xi) = \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} N_{i,k-1}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} N_{i+1,k-1}(\xi), \text{ for } k = 1, \dots, p. \end{cases} \tag{2.1}$$

This formula is called *Cox-de Boor recursion formula* from the names of C. de Boor and M.G. Cox that have first defined it [11]. By convention, we consider that $N_{n+1,k}$ is identically zero, for all $k > 0$, and that $\frac{0}{0} = 0$. Indeed, the latter case can happen in (2.1) whenever $\xi_{i+k} = \xi_i$ or if $\xi_{i+k+1} = \xi_{i+1}$. Remark that $N_{i,0}$, $i = 1, \dots, n$, are indicator functions corresponding to the first n knot spans of Ξ. In particular, they are piecewise constant functions. Since $N_{i,k}$, for $k > 0$ and $i = 1, \dots, n$, are linear combinations of the precedent $N_{i,k-1}$ and $N_{i+1,k-1}$, then they are piecewise polynomial functions as desired.

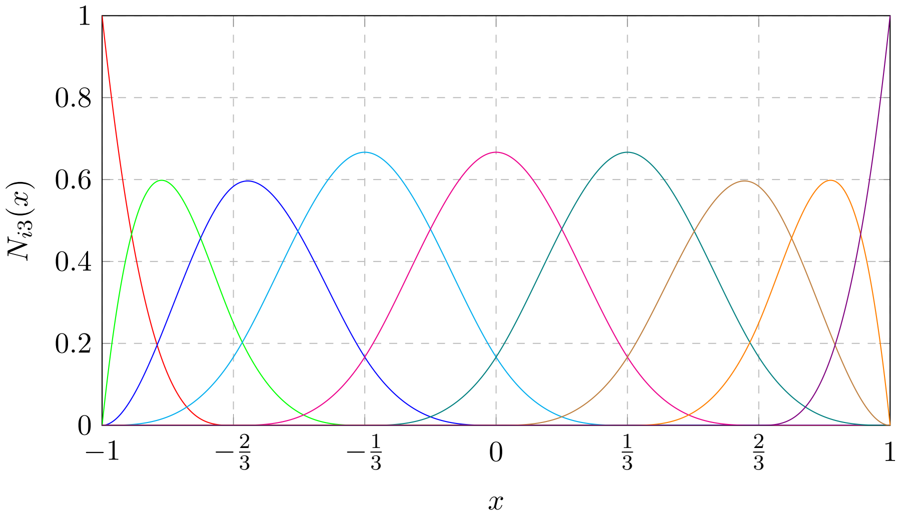


Figure 2.1: B-spline basis functions of order $p = 3$ defined by the uniform and open knot vector $\Xi = \{-1, -1, -1, -1, -\frac{2}{3}, -\frac{1}{3}, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$.

As an example, Figure 2.1 shows the B-spline basis functions of order $p = 3$ defined by the uniform open knot vector $\Xi = \{-1, -1, -1, -1, -\frac{2}{3}, -\frac{1}{3}, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$. Note that this knot vector is both open and uniform and in this case, $n = 9$. Moreover, the number of elements n_{el} is equal to 6, so we remark that $n = n_{\text{el}} + p$. This gives a first hint into the following proposition.

Proposition 2.1.7 *Let $\Xi = \{\xi_1, \dots, \xi_{n+p+1}\}$ be an open knot vector, such that $n \in \mathbb{N} \setminus \{0\}$ is the corresponding number of B-spline basis functions, and $p \in \mathbb{N} \setminus \{0\}$ their order. Let n_{el} be the number of elements of Ξ. Then $n = n_{\text{el}} + p$.*

Proof. Ξ being open, $p + 1$ knots are equal to ξ_1 and $p + 1$ other knots are equal to ξ_{n+p+1}. Consequently, if n_{int} is the number of internal knots, we have $(n+p+1)-2(p+1) = n_{\text{int}} = n_{\text{el}}-1$. Therefore, $n - p - 1 = n_{\text{el}} - 1$ and thus $n = p + n_{\text{el}}$. □

B-spline basis functions have some important properties stated in the following proposition:

Proposition 2.1.8 *Let p be the order of univariate B-spline basis functions $\{N_{i,p}\}_{i=1}^{n}$ defined from a knot vector $\Xi \subset \hat{\Omega}$.*

1. *If a knot has multiplicity k, the basis is C^{p-k}-continuous at that knot. In particular, if internal knots are not repeated, B-splines basis functions are C^{p-1}-continuous. If a knot has multiplicity p, then the basis is C^0-continuous and interpolatory at that location.*

2. *On the interior of each knot span, the basis functions are polynomials of order p, and thus in particular, they are C^∞-continuous.*

3. *The support of each basis function is compact and only consists of a small number of elements, corresponding to $p + 1$ knot spans.*

4. *Each knot span is in the support of $p + 1$ basis functions.*

5. *For all $i = 1, \dots, n$ and for all $\xi \in \hat{\Omega}$, $N_{i,p}(\xi) \geq 0$.*

6. *The B-spline basis functions form a partition of the unity, i.e. for all $\xi \in \hat{\Omega}$,*

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1.$$

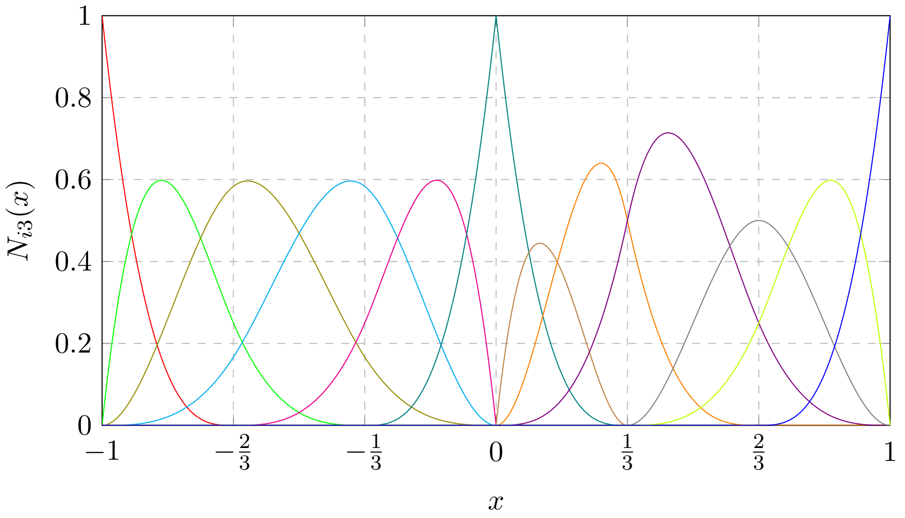


Figure 2.2: B-spline basis functions of order $p = 3$ defined by the uniform and open knot vector $\Xi = \{-1, -1, -1, -1, -\frac{2}{3}, -\frac{1}{3}, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$.

Proof. The proof can be found in the book [10] from J.A. Cottrell, T.J.R. Hughes and Y. Bazilevs. □

Note from property 1 that the B-spline basis functions are interpolatory at the boundary of the patch when an open knot vector is considered. Moreover, those properties can be easily checked in the specific example of Figure 2.1. Furthermore, Figure 2.2 shows the B-spline basis functions that are still of order $p = 3$, as in Figure 2.1, but that are defined by the uniform open knot vector $\Xi = \{-1, -1, -1, -1, -\frac{2}{3}, -\frac{1}{3}, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$. We remark that in this case, there are 12 basis functions instead of 9, and all the properties of Proposition 2.1.8 are still verified; the comparison with Figure 2.1 is interesting.

Furthermore, multivariate B-spline basis functions are built in a similar way, by means of tensor product rules, from the univariate B-spline basis functions. More precisely, a ν-variate B-spline basis, $\nu \in \mathbb{N} \setminus \{0\}$, is defined from a parametric space $\hat{\Omega}$ of dimension ν, that can be decomposed thanks to a cartesian product as $\prod_{i=1}^{\nu} \hat{\Omega}_i$, where each $\hat{\Omega}_i$ is a one-dimensional parametric space. From this fact, we give the following formal definition:

Definition 2.1.9 (Multivariate B-spline basis functions) Let $\Xi_i \subset \hat{\Omega}_i$, $i = 1, \dots, \nu$, be ν knot vectors that define n_i univariate B-spline basis functions of order p_i, $i = 1, \dots, \nu$, respectively, obtained from Cox-de Boor formula (2.1). Then the corresponding *ν-variate B-spline basis* is

$$\left\{ N_{\mathbf{j},\mathbf{p}} : \mathbf{j} = (j_1, \dots, j_\nu), 0 \le j_1 \le n_1, \dots, 0 \le j_\nu \le n_\nu;\ \mathbf{p} = (p_1, \dots, p_\nu) \right\}$$

such that for all $\xi = (\xi_1, \dots, \xi_\nu) \in \hat{\Omega}$,

$$N_{\mathbf{j},\mathbf{p}}(\xi) = \prod_{i=1}^{\nu} N_{j_i,p_i}(\xi_i).$$

Thanks to the tensor product structure of the multivariate B-spline basis functions, most properties of univariate B-spline basis functions given by Proposition 2.1.8 still hold: multivariate B-spline basis functions are piecewise polynomials of degree p_i, $i = 1, \dots, \nu$, respectively in each variable (i.e. in each direction of space), pointwise non-negative, form a partition of the unity and have compact support. Moreover, they are C^∞-continuous in each element, where a multi-dimensional element is also defined as the cartesian product of the corresponding one-dimensional elements of each Ξ_i, $i = 1, \ldots, \nu$. Moreover, their regularity in each direction of space is the same as the regularity of the underlying univariate B-spline basis corresponding to this direction. We will call *mesh* the set of multi-dimensional elements used to define some B-spline basis functions.

Finally, non-uniform rational B-spline (NURBS) basis functions are an extension of the B-spline basis functions, that is, they define a larger function space. Next section will explain more in details why this generalization is important. Let us first define what a univariate NURBS basis is.

Definition 2.1.10 (Univariate NURBS basis functions) Let $\hat{\Omega}$ be a one-dimensional parametric space, and let Ξ be a knot vector on $\hat{\Omega}$ that generates the n univariate B-spline basis functions of order p, $\{N_{i,p} : \hat{\Omega} \to \mathbb{R}\}_{i=1}^n$, by Cox-de Boor formula (2.1). Then given n weights $w_i \in \mathbb{R}$, $i = 1, \ldots, n$, we can define the set of *univariate NURBS basis functions* as $\{R_{i,p} : \hat{\Omega} \to \mathbb{R}\}_{i=1}^n$ such that for all $i \in \{1, \ldots, n\}$ and for all $\xi \in \hat{\Omega}$,

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi) w_i}{\sum_{j=1}^n N_{j,p}(\xi) w_j}.$$

We call *order* of NURBS basis functions the order of the underlying B-spline basis functions. Moreover, in general, positive weights are used to define NURBS basis functions, and most properties from the B-spline basis functions are kept:

Proposition 2.1.11 *Let p be the order of univariate NURBS basis functions $\{R_{i,p}\}_{i=1}^n$.*

- *The regularity of the NURBS basis functions is the same as the regularity of the underlying B-spline basis.*
- *The support of each NURBS basis function consists of only $p+1$ knot spans and is compact.*
- *Each knot span is in the support of $p+1$ NURBS basis functions.*
- *If the weights defining the NURBS basis are positive, then for all $i = 1, \ldots, n$ and for all $\xi \in \hat{\Omega}$, $R_{i,p}(\xi) \geq 0$.*
- *The NURBS basis functions form a partition of the unity.*

As in the B-spline case, the definition of multivariate NURBS basis functions follows naturally from a tensor product rule from the definition of a univariate NURBS basis.

Definition 2.1.12 (Multivariate NURBS basis functions) Let $\hat{\Omega} = \prod_{i=1}^{\nu} \hat{\Omega}_i$ be a parametric space of dimension $\nu \in \mathbb{N} \setminus \{0\}$, and let $\Xi_i \subset \hat{\Omega}_i$, $i = 1, \ldots, \nu$ be ν knot spans that generate $\prod_{i=1}^{\nu} n_i$ multivariate B-spline basis functions of multi-order $\mathbf{p}$, say $\{N_{\mathbf{j},\mathbf{p}}\}_{\mathbf{j} \in I}$ where

$$I = \{(i_1, \ldots, i_\nu) : 0 \leq i_1 \leq n_1, \ldots, 0 \leq i_\nu \leq n_\nu\}.$$

Then given $\prod_{i=1}^{\nu} n_i$ weights $w_{\mathbf{i}} \in \mathbb{R}$, $\mathbf{i} \in I$, we can define the set of *ν-variate NURBS basis functions* as $\{R_{\mathbf{i},\mathbf{p}} : \hat{\Omega} \to \mathbb{R}\}_{\mathbf{i} \in I}$ such that for all $\mathbf{i} \in I$ and for all $\xi \in \hat{\Omega}$,

$$R_{\mathbf{i},\mathbf{p}}(\xi) = \frac{N_{\mathbf{i},\mathbf{p}}(\xi) w_{\mathbf{i}}}{\sum_{\mathbf{j} \in I} N_{\mathbf{j},\mathbf{p}}(\xi) w_{\mathbf{j}}}.$$

Note that whenever the weights are constant, that is if $w_{\mathbf{i}} = c \in \mathbb{R}$ for all $\mathbf{i} \in I$, then for all $\xi \in \hat{\Omega}$, $R_{\mathbf{i},\mathbf{p}}(\xi) = N_{\mathbf{i},\mathbf{p}}(\xi)$ thanks to the partition of unity property of the B-spline basis functions, exposed in Proposition 2.1.8. This specific case show that B-splines are particular cases of NURBS.

The definition and properties of multivariate B-spline and NURBS basis functions have been given as a sake of completeness but in the following, only univariate B-spline functions will be considered.

2.1.3 B-spline and NURBS geometries

The previously introduced B-spline basis functions are used to build geometrical domains called *B-spline geometries.* As in the previous section, let us first define a one-dimensional B-spline geometry in $\mathbb{R}^d$, $d \in \mathbb{N} \setminus \{0\}$, called B-spline curve.

Definition 2.1.13 (B-spline curve) Let $\hat{\Omega}$ be a one-dimensional parametric domain and let Ξ be a knot span on $\hat{\Omega}$ that generates n univariate B-splines basis functions of order p, say $\{N_{i,p} : \hat{\Omega} \to \mathbb{R}\}_{i=1}^{n}$, thanks to Cox-de Boor formula (2.1). Given a set of n points $\{\mathbf{P}_i\}_{i=1}^{n} \subset \mathbb{R}^d$, $d \in \mathbb{N} \setminus \{0\}$, called *control points*, a *B-spline curve* embedded in $\mathbb{R}^d$ is parametrically described as

$$\mathbf{S} : \hat{\Omega} \to \Omega \subset \mathbb{R}^d, \;\; \mathbf{S}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\mathbf{P}_i \text{ for all } \xi \in \hat{\Omega}. \tag{2.2}$$

That is, Ω is the geometrical subspace of $\mathbb{R}^d$ corresponding to the B-spline curve, while $\mathbf{S}$ is its parametric description.

To sum up, a curve spline is parametrically defined as the linear combination of the univariate B-spline basis functions, with control points in the real geometrical space $\mathbb{R}^d$ as coefficients. Consequently, n is both the number of B-spline basis functions and the number of control points required to define a B-spline curve.

Even if only B-spline curves will be considered in the following, we also give the generalization of the definition of B-spline geometries of higher dimension as a sake of completeness. As in the case of multivariate B-spline basis functions, B-spline geometries of dimension $\nu > 1$ are defined by means of tensor product rules from the B-spline curves.

Definition 2.1.14 (B-spline geometry) Let $\hat{\Omega} = \prod_{i=1}^{\nu} \hat{\Omega}_i$ be a ν-dimensional parametric domain, and let $\Xi_i \subset \hat{\Omega}_i$, $i = 1, \dots, \nu$ be ν knot spans that generate $\prod_{i=1}^{\nu} n_i$ multivariate B-spline basis functions of order $\mathbf{p} = (p_1, \dots, p_\nu)$, say $\{N_{\mathbf{j},\mathbf{p}} : \mathbf{j} = (j_1, \dots, j_\nu), 0 \leq j_i \leq n_i,\ i = 1, \dots, \nu\}$. To simplify the notation, let $I := \{\mathbf{j} = (j_1, \dots, j_\nu), 0 \leq j_i \leq n_i,\ i = 1, \dots, \nu\}$. Then given a set of $\prod_{i=1}^{\nu} n_i$ points $\{\mathbf{P}_{\mathbf{i}}\}_{\mathbf{i} \in I} \subset \mathbb{R}^d$, $d \in \mathbb{N} \setminus \{0\}$, still called *control points*, a *B-spline geometry* of dimension ν embedded in $\mathbb{R}^d$ is parametrically described as

$$\mathbf{S} : \hat{\Omega} \to \Omega \subset \mathbb{R}^d, \;\; \mathbf{S}(\xi) = \sum_{\mathbf{i} \in I} N_{\mathbf{i},\mathbf{p}}(\xi)\mathbf{P}_{\mathbf{i}} \text{ for all } \xi \in \hat{\Omega}.$$

As in the one dimensional case, Ω is the geometrical subspace of $\mathbb{R}^d$ corresponding to the B-spline geometry while $\mathbf{S}$ is its parametric description.

When $\nu = 2$, B-spline geometries derived from bivariate B-spline basis functions are called *B-spline surfaces*; when $\nu = 3$, B-spline geometries are called *B-spline solids.* Moreover, we have the following proposition:

Proposition 2.1.15 *B-spline geometries inherit from all properties of the corresponding B-spline basis functions. In particular:*

- *the regularity of the B-spline geometry is determined by the regularity of the underlying B-spline basis;*

- *the B-spline basis is not interpolatory, in general, except at knots whose multiplicity is p. Consequently, the B-spline geometry does not pass through the control points, in general, except at points where the tangent is discontinuous;*

- *since B-spline basis functions have reduced support, then changing a single control point will only affect locally the B-spline geometry.*

However, even if B-splines can already describe a wide range of geometries, we need to use NURBS instead of B-splines to be able to define common geometrical objects such as circles or ellipses, or in general any object containing conic sections. Now, we can then define a NURBS curve/surface/solid, or more generally a NURBS geometry, similarly as a B-spline geometry. We directly give the general definition of a NURBS geometry that is obtained from the one of a NURBS curve by means of tensor product rules.

Definition 2.1.16 (NURBS geometry) Let $\hat{\Omega} = \prod_{i=1}^{\nu} \hat{\Omega}_i$ be a ν-dimensional domain, $\nu \in \mathbb{N}\backslash\{0\}$, and let $\Xi_i \subset \hat{\Omega}_i$, $i = 1, \dots, \nu$ be ν knot spans that generate $\prod_{i=1}^{\nu} n_i$ B-spline basis functions of order $\mathbf{p}$. Let $w_{\mathbf{i}} \in \mathbb{R}$, $\mathbf{i} \in I$, $I := \{\mathbf{j} = (j_1, \dots, j_\nu), 0 \leq j_i \leq n_i,\ i = 1, \dots, \nu\}$, be $\prod_{i=1}^{\nu} n_i$ given weights. From those weights and the B-spline basis functions, we obtain a NURBS basis $\{R_{\mathbf{i},\mathbf{p}}\}_{\mathbf{i}\in I}$. Then given a set of $\prod_{i=1}^{\nu} n_i$ points $\{\mathbf{P}_{\mathbf{i}}\}_{\mathbf{i}\in I} \subset \mathbb{R}^d$, $d \in \mathbb{N} \setminus \{0\}$, still called *control points*, a *NURBS geometry* of dimension ν embedded in $\mathbb{R}^d$ is parametrically described as

$$\mathbf{F} : \hat{\Omega} \to \Omega \subset \mathbb{R}^d, \ \ \mathbf{F}(\xi) = \sum_{\mathbf{i}\in I} R_{\mathbf{i},\mathbf{p}}(\xi)\mathbf{P}_{\mathbf{i}} \in \mathbb{R}^d \ \text{ for all } \xi \in \hat{\Omega}.$$

Therefore, as for B-spline geometries, a NURBS geometry is defined parametrically as a linear combination of the NURBS basis functions, taking the control points as coefficients.

2.2 Isogeometric analysis and collocation methods

B-splines or NURBS-based isogeometric analysis uses respectively the B-splines or the NURBS basis functions that have been used to build the geometrical domain in order to approximate the solution of a PDE. More precisely, if $\mathbf{F}$ is the parametrization map of the B-spline or NURBS region representing the geometry, as in Definitions 2.1.13, 2.1.14 and 2.1.16, then the finite approximated space in which lies the numerical solution is

$$\mathbb{B}_p = \text{span}\{R_p \circ \mathbf{F}^{-1} : R_p \text{ is a B-spline or NURBS basis function of order } p\}.$$

The image of the elements in the parametric space are elements in the physical space and constitute the physical mesh. In the following, h will represent the size of the elements in the physical space. For more details about the theory of isogeometric analysis, we refer the interested reader to [10].

To solve a PDE, the most widely used technique is the finite element method in which the differential problem is transformed into its weak form: the PDE in its strong form is multiplied by a test function and then integrated over the computational domain. From this weighted residual formulation, the Galerkin method is often used. It consists in seeking the solution in a suitable discrete space which is also the space in which belong the test functions [26]. In the case of isogeometric analyis, the solution is sought in the space $\mathbb{B}_p$ [10].

2.2.1 Refinement

It has been previously said that the isogeometric analysis had been invented in order to have a method that is always geometrically exact, and that makes the refinement of the mesh easier. First, IGA preserves the geometry exactly since the meshing is defined directly from the parametrization of the geometrical B-spline or NURBS geometry. In this section, we discuss about refinement.

In order to have more detailed B-spline or NURBS geometries, and then in order to have more precise IGA solutions to the differential problems on those geometries, three different procedures called *refinement* can be followed. Two of them are similar to what is used in the finite element method and are called p- and h-refinements; the third one is proper to IGA and is called k-refinement. Since one of the principal goals of IGA is to always work on exact geometries, p-, h- or k-refinements should never change the geometry considered.

Firstly, *p-refinement*, also called *order elevation*, consists in increasing the degree of the underlying polynomials, and thus also in increasing the degree of the B-spline basis functions. However, the regularity of the basis should not change in the whole domain. Furthermore, we recall that the continuity of the basis at each knot is equal to p meines the multiplicity of each knot. Consequently, to keep the same regularity in the basis when p is increased by 1, we also need to increase the multiplicity of each knot by 1, and no additional knot needs to be inserted into the underlying knot span(s). The original function space is embedded into the resulting function space that we get from the span of the basis functions obtained from the new knot span, that is, the basis is enriched. For more details about the algorithms used to do such refinement, we refer the interesting reader to [25].

Secondly, *h-refinement*, also called *knot insertion*, consists in adding some knots to the original knot vector. Still to keep the same geometry (parametrically and geometrically) and to have the same regularity of the B-spline basis functions in the whole domain, we need to change and choose appropriately the control points in the following way: let $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$ be the initial knot vector, and let $\tilde{\Xi} = \{\mu_1 = \xi_1, \ldots, \mu_{n+m+p+1} = \xi_{n+p+1}\}$ be the knot vector obtained after adding m new knots. Moreover, let $\mathcal{C}$ be the original set of control points of cardinality n. Then the new set of control points that one has to take in order to obtain the exact same geometry is defined by the transformation $T^p\mathcal{C}$, where $T^p \in [\xi_1, \xi_k]^{(n+m)\times n}$ is defined recursively as: for all $i = 1, \ldots, n+m$, for all $j = 1, \ldots, n$,

$$\begin{cases} T^0_{ij} = \begin{cases} 1 \text{ if } \xi_j \leq \mu_i < \xi_{j+1}, \\ 0 \text{ otherwise}, \end{cases} \\ T^{q+1}_{ij} = \frac{\mu_{i+q}-\xi_j}{\xi_{j+q}-\xi_j} T^q_{ij} + \frac{\xi_{j+q+1}-\mu_{i+q}}{\xi_{j+q+1}-\xi_{j+1}} T^q_{i(j+1)}, \quad \text{for } q = 0, \ldots, p-1. \end{cases}$$

More details can be found in [25]. If an already existing knot is added, its multiplicity in the knot vector is increased. Hence, the continuity of the basis functions is reduced, but thanks to this choice of control points, the continuity of the geometry is conserved. As for the p-refinement case, the original function space is embedded into the newly created function space. With this type of refinement, both the number of basis functions and the number of elements increase.

Finally, *k-refinement* is a combination of order elevation and knot insertion. More precisely, it consists first into elevating the order from some p to some $q > p$, and then into adding a knot $\tilde{\xi}$ into the knot vector so that the basis functions at $\tilde{\xi}$ are C^{q-1}-continuous. One has to be careful to the fact that this sequence of operations is not commutative. Indeed, if a knot $\tilde{\xi}$ is inserted before elevating the order from p to $q > p$, then the basis functions at $\tilde{\xi}$ would only be C^{p-1}-continuous. This type of refinement is unique to IGA and cannot be found in the finite element analysis since in the latter case, basis functions are only C^0-continuous between elements.

In the following, we will concentrate on B-spline geometries that always have the same fixed continuity between elements. More precisely, we will work with the spaces $\mathbb{B}_p^{p-1}$ whose B-splines have order p and are globally C^{p-1}-continuous. That is, they are C^{p-1}-continuous between elements since they are already always C^{∞}-continuous inside each element.

2.2.2 Isogeometric collocation methods

In contrast to the Galerkin isogeometric method, isogeometric collocation is based on the evaluation of the strong formulation of the PDE at some collocation nodes. This method requires basis functions that are sufficiently smooth to handle possibly high order differential operators. This is naturally the case with the basis functions used in isogeometric analysis presented previously. IGA collocation methods have been introduced by Auricchio and al. in [2]. The major advantage of isogeometric collocation with respect to the Galerkin isogeometric analysis is the low effort required for the construction of the problem and for the assembly. Indeed in Galerkin methods, the construction of the problem and the assembly are based on full Gauss quadrature to compute the integrals. This is very efficient for C^0-continuous basis functions such as in the finite element method, but it is inefficient for isogeometric analysis whose B-spline basis functions have a higher order continuity [10, 17]. Instead, collocation methods only need one point evaluation per basis function, reducing notably the computational costs.

However, in the actual state of the research, the rate of convergence of isogeometric collocation methods are in general lower than in isogeometric Galerkin methods [1, 23, 24]. In the following, we present more in details how to solve a PDE with an IGA collocation method, together with the most widely used collocation points found in literature.

Let us introduce briefly the IGA collocation method. Let us consider the following general differential problem:

$$\begin{cases} \mathcal{L}u = f \text{ in } \Omega, \\ \mathcal{B}u = g \text{ on } \partial\Omega, \end{cases}$$

where $\Omega \in \mathbb{R}^r$ is the geometrical domain with $r \in \mathbb{N}\setminus\{0\}$, $u : \Omega \to \mathbb{R}$ is the solution, $\mathcal{L}$ and $\mathcal{B}$ are linear differential operators representing respectively the problem and the boundary conditions, and f and g are given data functions. Any collocation method used to solve such problem is based on the choice of a finite set of collocation points; the way to choose such points will be discussed later on. In general, collocation points are defined in the parametric space $\hat{\Omega}$ from which the B-spline or NURBS geometry Ω has been built. So let $\hat{\mathcal{C}} := \{\hat{\tau}_i\}_{i\in I} \subset \hat{\Omega}$ and let $\tau_i := \mathbf{F}(\hat{\tau}_i)$, for all $i \in I$, where $\mathbf{F}$ is the parametrization map of Ω, as in section 2.2. We then define $\mathcal{C} := \{\tau_i\}_{i\in I} \subset \Omega$. Let us separate $\mathcal{C}$ in two distinct sets: $\mathcal{C}_{\mathcal{B}}$ corresponds to the set of collocation points belonging to the boundary $\partial\Omega$, and $\mathcal{C}_{\mathcal{L}}$ corresponds to the set of collocation points belonging to the interior of Ω. Note that this is equivalent to separating $\hat{\mathcal{C}}$ into the set of points belonging to the boundary $\partial\hat{\Omega}$ and the set of points belonging to the interior of $\hat{\Omega}$, and then mapping the two sets to Ω through $\mathbf{F}$. Then, the collocation problem reads

$$\begin{cases} \mathcal{L}u(\tau) = f(\tau), \ \forall\tau \in \mathcal{C}_{\mathcal{L}}, \\ \mathcal{B}u(\tau) = g(\tau), \ \forall\tau \in \mathcal{C}_{\mathcal{B}}. \end{cases}$$

Collocation points influence strongly the convergence and the stability of the method. Consequently, they must be chosen carefully. Moreover, if multivariate NURBS or B-splines are considered, collocation points are first defined on each direction of the parametric space and then obtained on the whole space thanks to a tensor-product rule. That is why in the following, we will mostly concentrate on one-dimensional B-spline spaces.

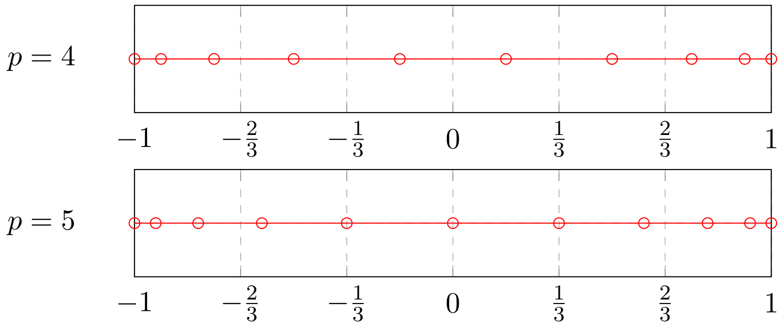


Figure 2.3: Distribution of Greville abscissae. On top, $p = 4$ and $n = 10$, and the knot vector used is $\Xi = \{-1, -1, -1, -1, -1, -\frac{2}{3}, -\frac{1}{3}, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1, 1\}$. At the bottom, $p = 5$, $n = 11$, and the knot vector used is $\Xi = \{-1, -1, -1, -1, -1, -1, -\frac{2}{3}, -\frac{1}{3}, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1, 1, 1\}$.

Let $\mathbb{B}_p$ be the space in which the solution of the differential equation is sought, where $p \in \mathbb{N} \setminus \{0\}$ is the order of the underlying B-splines, and let n be the dimension of $\mathbb{B}_p$. Moreover, let $N_{i,p}$, $i = 1, \ldots, n$ be the B-spline basis functions of $\mathbb{B}_p$. Then there exist $u_i \in \mathbb{R}$, $i = 1, \ldots, n$ such that u can be written as

$$u = \sum_{i=1}^{n} u_i N_{i,p}.$$

Consequently, the collocation problem is transformed into the following linear system whose unknowns are the coefficients u_i:

$$\begin{cases} \sum_{i=1}^{n} u_i \, \mathcal{L} N_{i,p}(\tau) = f(\tau), \ \forall \tau \in \mathcal{C}_{\mathcal{L}}, \\ \sum_{i=1}^{n} u_i \, \mathcal{B} N_{i,p}(\tau) = g(\tau), \ \forall \tau \in \mathcal{C}_{\mathcal{B}}. \end{cases}$$

This system can be rewritten under matrix form and if the choice of collocation points leads to a well-posed problem, it can be easily solved to find the solution $u \in \mathbb{B}_p$.

2.2.3 Greville and Demko abscissae

The most widely used IGA collocation points are the *Greville abscissae* [2]. Given a knot vector $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$, and if univariate NURBS or B-splines are considered, they are defined as the mean of p consecutive knots, that is:

$$\hat{\tau}_i := \frac{\xi_{i+1} + \xi_{i+2} + \ldots + \xi_{i+p}}{p}, \quad \tau_i = \mathbf{F}(\hat{\tau}_i),$$

for all $i = 1, \ldots, n$. If open knots are used, $\xi_1 = \xi_2 = \ldots = \xi_{p+1}$ and $\xi_{n+1} = \xi_{n+2} = \ldots = \xi_{n+p+1}$. Consequently, it is easy to see that in this case, $\hat{\tau}_1 = \xi_1$ and $\hat{\tau}_n = \xi_{p+n+1}$. In the case of bivariate NURBS or B-splines, Greville abscissae are defined by means of a tensor product rule: given the two knot vectors $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$ and $H = \{\eta_1, \ldots, \eta_{m+q+1}\}$, they are defined as: $\forall i = 1, \ldots, n$, $\forall j = 1, \ldots, m$,

$$\hat{\xi}_i := \frac{\xi_{i+1} + \xi_{i+2} + \ldots + \xi_{i+p}}{p}, \quad \hat{\eta}_j := \frac{\eta_{j+1} + \eta_{j+2} + \ldots + \eta_{j+q}}{q}, \quad \tau_{ij} = \mathbf{F}(\hat{\xi}_i, \hat{\eta}_j).$$

This is easily generalizable to get the Greville abscissae associated with any multivariate NURBS or B-spline. Some results show that the method derived from those collocation points is stable up to order 3, but it can be unstable on particular non-uniform meshes when the order is larger than 19, see [20, 21].

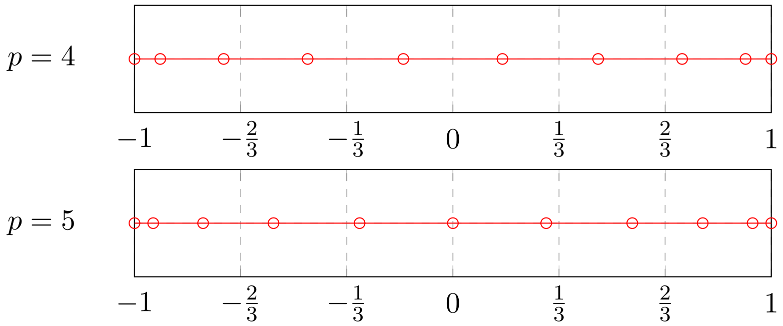


Figure 2.4: Distribution of Demko abscissae. On top, $p = 4$ and $n = 10$, and the knot vector used is $\Xi = \{-1, -1, -1, -1, -1, -\frac{2}{3}, -\frac{1}{3}, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1, 1\}$. At the bottom, $p = 5$, $n = 11$, and the knot vector used is $\Xi = \{-1, -1, -1, -1, -1, -1, -\frac{2}{3}, -\frac{1}{3}, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1, 1, 1\}$.

Figure 2.3 shows the distribution of Greville abscissae in the case in which $\hat{\Omega} = \Omega = (-1, 1)$ is divided into 6 elements and is represented by B-splines of order 4 or 5 that are globally C^3- and C^4-continuous, respectively.

Other very widely used IGA collocation points in the literature are the *Demko abscissae* proposed by Demko in [14]. Those points correspond to the extrema of the Chebyshev splines, that is the splines whose extrema take the values ±1 and that have the maximum number of oscillations. Demko abscissae can be obtained thanks to Remez iterative algorithm. For more information, the interesting reader can look at the Matlab documentation of the spline toolbox [12]. This is for now the only choice of collocation points which it is proved to lead to a stable method for any mesh size and any B-spline order [2].

Figure 2.4 shows the distribution of Demko abscissae in the same case as Figure 2.3 for Greville abscissae, that is in the case in which $\hat{\Omega} = \Omega = (-1, 1)$ is divided into 6 elements and is represented by B-splines of order 4 or 5 that are globally C^3- and C^4-continuous, respectively. It can be clearly seen that Demko and Greville abscissae are not the same but their number in each element is the same.

However, none of these choices of collocation points give an optimal error convergence. By optimal, we mean the same convergence of the error as what we would get if the approximate solution were obtained with the Galerkin isogeometric method [10, 26]. Indeed, it has been seen in [3, 9, 10] that under h-refinement, that is under refinement of the mesh only, without changing the order p of the spline basis functions nor their continuity, the error behaves as h^p in the H^1-norm, and as h^{p+1} in the L^2-norm, with h being the size of the elements of the mesh. However, it has been shown ([14, 15, 23]) that Greville and Demko abscissae lead to isogeometric collocation methods that converge:

- optimally in the H^1-norm for even values of the B-spline order p, i.e. as h^p;
- one order sub-optimally in the L^2-norm for p even, i.e. as h^p instead as h^{p+1};
- one order sub-optimally in the H^1-norm for odd values of p, i.e. as h^{p-1} instead of h^p;
- two orders sub-optimally in the L^2-norm when p odd is used, i.e. as h^{p-1} instead of h^{p+1}.

Furthermore, Auricchio and al. in [2] have shown under a theoretical framework that in the L^∞- and the $W^{1,\infty}$-norms, the optimal convergence of the error of an isogeometric collocation solution has a behavior in h^p; while in the $W^{2,\infty}$-norms, the optimal convergence of the error of an IGA collocation solution has a behavior in h^{p-1}. Let us recall that

$$\| \cdot \|_{W^{j,\infty}} = \sum_{i=0}^{j} \|D^i \cdot \|_{L^\infty},$$

where D^i is the operator representing the i^{th} derivative. Finally, still in [2], it is shown that when Greville or Demko abscissae are used, the error behaves optimally in the $W^{2,\infty}$-norm for any choice of p, and in the L^∞- and in the $W^{1,\infty}$-norms when p is even, but it is one order sub-optimal in the L^∞- and in the $W^{1,\infty}$-norms when p is odd.

Chapter 3

Collocation method via suitable quadrature formulas

It is well known that the Galerkin spectral element method with numerical integration (SEM G-NI) is equivalent to the spectral collocation method (collocated SM) on Gauss-Legendre-Lobatto points, as it is shown in [8], see also [26]. In this chapter, our aim is to look for similar results with IGA, that is to find a suitable quadrature formula so that the IGA Galerkin method can be equivalent to the IGA collocation method.

We will first review how this is done in the case of the Galerkin spectral element method with numerical integration, and then we will try to move this idea to the isogeometric analysis.

3.1 Equivalence between SEM G-NI and the spectral collocation method

In this section, we first present the Galerkin spectral element method with numerical integration, and then prove the equivalence of this method and the spectral collocation method.

3.1.1 The spectral element method with numerical integration

As its name indicates, the Galerkin spectral element method with numerical integration (SEM G-NI) is a Galerkin method on piecewise polynomial subspaces of high degree, which makes use of Gaussian numerical integration on each element. The space where the solution is sought is the same as the space of the test functions. As basis functions of this space, piecewise Lagrange interpolants on well-chosen nodes are used. That is, the computational domain Ω is decomposed into n_{el} elements (x_k, x_{k+1}), $k = 0, \dots, n_{\text{el}} - 1$, and basis functions are defined on each element. Looking first on a single reference element $(-1, 1)$, and given a set of n nodes $\{\tau_i\}_{i=1}^{n}$ in $(-1, 1)$, the basis functions $\hat{\phi}_i$ for $i \in \{1, \dots, n\}$ are:

$$\hat{\phi}_i(x) = \prod_{j=1, j\neq i}^{n} \frac{x - \tau_j}{\tau_i - \tau_j}.$$

Usually, one chooses the Gauss-Legendre-Lobatto (GLL) nodes. That is, if p is the desired underlying polynomial degree, then the nodes correspond to the roots of $(1 - x^2)L_p'(x)$, where L_p is the Legendre polynomial of degree p defined recursively in this way on the reference element

$(-1,1)$: $\forall x \in (-1,1)$,

$$\begin{cases} L_0(x) = 1, \\ L_1(x) = x, \\ L_{k+1}(x) = \frac{2k+1}{k+1} x L_k(x) - \frac{k}{k+1} L_{k-1}(x), \text{ for } k > 1. \end{cases}$$

There are $p+1$ such nodes. Therefore, if we let

$$\Xi_k(\xi) = (\xi+1)\frac{x_{k+1} - x_k}{2} + x_k, \ \ \xi \in [-1,1],$$

be the affine transformation that maps $(-1,1)$ into (x_k, x_{k+1}), $k = 0, \dots, n_{\text{el}} - 1$, then the basis functions used in SEM on each element are $\phi_i^{(k)} := \hat{\phi}_i \circ \Xi_k^{-1}$, $k = 0, \dots, n_{\text{el}} - 1$, $i = 0, \dots, p$. They correspond to Lagrange interpolants at the Gauss-Legendre-Lobatto nodes of the interval (x_k, x_{k+1}), that is at the nodes in $\{\Xi_k(\tau_i)\}_{i=0}^{p}$, for all $k = 0, \dots, n_{\text{el}} - 1$. To define a global basis from them, that is still interpolating at the Gauss-Legendre-Lobatto nodes on each element, we first define the basis functions associated to internal nodes of each element (x_k, x_{k+1}), for $k = 0, \dots, n_{\text{el}} - 1$, as follows:

$$\Phi_{pk+(i+1)}(x) = \begin{cases} \phi_i^{(k)}(x), & \text{if } x \in (x_k, x_{k+1}), \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

for all $i = 1, \dots, p-1$. Then, we define the basis functions associated to element's boundary nodes:

$$\Phi_{pk+1}(x) = \begin{cases} \phi_0^{(k)}(x), & \text{if } x \in (x_k, x_{k+1}), \\ \phi_p^{(k-1)}(x), & \text{if } x \in (x_{k-1}, x_k), \\ 0, & \text{otherwise,} \end{cases} \tag{3.2}$$

for all $k = 1, \dots, n_{\text{el}} - 1$. Finally, we define the basis functions associated to the boundaries of Ω, Φ_1 and $\Phi_{pn_{\text{el}}+1}$, as in equation (3.1) with $k = 0, i = 0$ and $k = n_{\text{el}}, i = 0$, respectively. As a consequence, the finite approximate function space in which the solution is sought is $\mathbb{Q}_p(\Omega_h)$, that is the set of continuous piecewise polynomials of degree at most p, where Ω_h is a uniform partition, approximation of Ω. Note that we will write $\mathbb{Q}_p^0(\Omega_h)$ the set of functions in $\mathbb{Q}_p(\Omega_h)$ that are zero at the boundary of Ω_h. If a single element is considered, we speak of spectral method instead of spectral element method.

Moreover, from the weak formulation of the differential equation, SEM with numerical integration achieves efficiency by using quadrature formulas to compute the integrals. When not made explicit, the quadrature rule used in SEM G-NI is the Gauss-Legendre-Lobatto quadrature rule, for which the quadrature points reside at the nodal points. This quadrature rule results in diagonal mass matrices since Lagrange basis functions are interpolant on the quadrature nodes. Moreover, the weights of this quadrature rule on the reference element $(-1,1)$ are computed as follows: for all $j \in \{0, \dots, p\}$,

$$w_j = \frac{2}{p(p+1)} \frac{1}{\left[L_p\left(\tau_j\right)\right]^2}.$$

The scaled weights on each element (x_k, x_{k+1}) are then $\Xi_k(w_j)$ for all $j = 1, \dots, p$. The degree of accuracy of this quadrature formula is $2p-1$. It will thus not compute exactly mass matrices. For more details about the spectral element methods, we refer the interested reader to [7, 8].

3.1.2 Equivalence between formulations

In the following, we will show that SEM G-NI is equivalent to the so-called spectral collocation method. The spectral collocation method is the collocation method where the basis functions are the same as the ones of SEM G-NI and where the collocation points are the Gauss-Legendre-Lobatto nodes on each element of the space decomposition. We will see that, however, the differential operator needs to be slightly modified.

Let us consider a general one-dimensional diffusion-transport-reaction problem with Dirichlet homogeneous boundary conditions, which consists in finding u such that:

$$\begin{cases} \mathcal{L}u := (-\mu u')' + (cu)' + \sigma u = f \text{ in } (a,b), \\ u(a) = u(b) = 0, \end{cases} \tag{3.3}$$

with $\mu \in L^\infty(a,b)$, $\mu(x) \geq \mu > 0$, $c \in \mathbb{R}$, $\sigma \in L^2(a,b)$, $\sigma \geq 0$ almost everywhere, $a, b \in \mathbb{R}$ and $f \in L^2(a,b)$.

The weak formulation of this problem reads: find $u \in V := H_0^1(a,b)$ such that for all $v \in V$,

$$\int_\Omega \mu u'v' \,\mathrm{d}\Omega - \int_\Omega cuv' \,\mathrm{d}\Omega + \int_\Omega \sigma uv \,\mathrm{d}\Omega = \int_\Omega fv \,\mathrm{d}\Omega.$$

It is well known that this problem is well posed under the condition $\frac{1}{2}c' + \sigma \geq 0$ almost everywhere. That is, in this case, there exists a unique solution u in $H_0^1(\Omega)$, thanks to Lax-Milgram theorem. For more details, refer to the book of Quarteroni [26]. Considering a space decomposition of (a,b), say $\Omega_h = \{(t_{i-1}, t_i)\}_{i=1}^{n_{\text{el}}}$, and fixing the order of the polynomials to p, the Galerkin spectral element method with numerical integration states as follows: find $u_p \in V_p := \mathbb{Q}_p^0(\Omega_h)$ such that $\forall v_p \in V_p$,

$$\begin{aligned} &\sum_{i=1}^{n_{\text{el}}} \Big[\big((\mu u_p')|_{(t_{i-1},t_i)}, v_p'|_{(t_{i-1},t_i)}\big)_p - \big((cu_p)|_{(t_{i-1},t_i)}, v_p'|_{(t_{i-1},t_i)}\big)_p + \big((\sigma u_p)|_{(t_{i-1},t_i)}, v_p|_{(t_{i-1},t_i)}\big)_p \Big] \\ &= \sum_{i=1}^{n_{\text{el}}} (f|_{(t_{i-1},t_i)}, v_p|_{(t_{i-1},t_i)})_p, \end{aligned} \tag{3.4}$$

where $(\cdot,\cdot)_p$ is the discrete Gauss-Legendre-Lobatto inner product defined as

$$(h,g)_p = \sum_{i=0}^{p} \alpha_i h(\tau_i) g(\tau_i) \quad \forall h, g \in C^0(I),$$

where I is any interval of $\mathbb{R}$, $\{\tau_i\}_{i=0}^p$ are the Gauss-Legendre-Lobatto nodes on I, and $\{\alpha_i\}_{i=0}^p$ are the corresponding Gauss-Legendre-Lobatto quadrature formula weights.

From equation (3.4), we want to retrieve the spectral collocation formulation of our problem. To do so, we would like to counter-integrate by parts (3.4). Let us recall that the basis functions used on each element are the Lagrange interpolants on the Gauss-Legendre-Lobatto nodes defined globally as it is done in section 3.1.1. Let $\Pi_p : C^0([a,b]) \to V_p$ be the interpolant operator such that $\Pi_p(g)|_{[t_i,t_{i+1}]} = \sum_{j=0}^p g(\tau_{i,j})\phi_{i,j}$ for all $g \in V_p$, where $\{\phi_{i,j}\}_{j=0}^p$ are the Legendre interpolants on the Gauss-Legendre-Lobatto nodes $\{\tau_{i,j}\}_{j=0}^p$ on the interval $[t_i, t_{i+1}]$, for all $i = 0, \ldots, n_{\text{el}} - 1$. Finally, let $\{\alpha_{i,j}\}_{j=0}^p$ be the corresponding Gauss-Legendre-Lobatto quadrature formula weights. Consequently, for all $g \in C^0([a,b])$ and for all $i = 0, \ldots, n_{\text{el}} - 1$, since $\Pi_p g$ is a polynomial of degree p,

$$\sum_{j=0}^{p} \alpha_{i,j} g(\tau_{i,j}) = \sum_{j=0}^{p} \alpha_{i,j} \Pi_p g(\tau_{i,j}) = \int_{t_i}^{t_{i+1}} \Pi_p g(x) \,\mathrm{d}x.$$

Therefore, since $\Pi_p(\mu u_p')v_p'$ and $[\Pi_p(\mu u_p')]'v_p$ belong to $\mathbb{Q}_{2p-1}(\Omega_h)$ and since Gauss-Legendre-Lobatto quadrature formula has a degree of exactness equal to $2p-1$, then

$$\begin{aligned}
\sum_{i=0}^{n_{\mathrm{el}}-1} \Big((\mu u_p')|_{(t_i,t_{i+1})}, v_p'|_{(t_i,t_{i+1})}\Big)_p &= \sum_{i=0}^{n_{\mathrm{el}}-1} \Big(\Pi_p(\mu u_p')|_{(t_i,t_{i+1})}, v_p'|_{(t_i,t_{i+1})}\Big)_p \\
&= \sum_{i=0}^{n_{\mathrm{el}}-1} \int_{t_i}^{t_{i+1}} \Pi_p(\mu u_p')|_{(t_i,t_{i+1})} v_p'|_{(t_i,t_{i+1})} \,\mathrm{d}x \\
&= \int_a^b \Pi_p(\mu u_p') v_p' \,\mathrm{d}x \\
&= -\int_a^b [\Pi_p(\mu u_p')]' v_p \,\mathrm{d}x \\
&= -\sum_{i=0}^{n_{\mathrm{el}}-1} \int_{t_i}^{t_{i+1}} [\Pi_p(\mu u_p')]'|_{(t_i,t_{i+1})} v_p|_{(t_i,t_{i+1})} \,\mathrm{d}x \\
&= -\sum_{i=0}^{n_{\mathrm{el}}-1} \Big([\Pi_p(\mu u_p')]'|_{(t_i,t_{i+1})}, v_p|_{(t_i,t_{i+1})}\Big)_p,
\end{aligned}$$

and similarly

$$\begin{aligned}
-\sum_{i=0}^{n_{\mathrm{el}}-1} \Big((c u_p)|_{(t_i,t_{i+1})}, v_p'|_{(t_i,t_{i+1})}\Big)_p &= -\sum_{i=0}^{n_{\mathrm{el}}-1} \Big(\Pi_p(c u_p)|_{(t_i,t_{i+1})}, v_p'|_{(t_i,t_{i+1})}\Big)_p \\
&= -\sum_{i=0}^{n_{\mathrm{el}}-1} \int_{t_i}^{t_{i+1}} \Pi_p(c u_p)|_{(t_i,t_{i+1})} v_p'|_{(t_i,t_{i+1})} \,\mathrm{d}x \\
&= -\int_a^b \Pi_p(c u_p) v_p' \,\mathrm{d}x \\
&= \int_a^b [\Pi_p(c u_p)]' v_p \,\mathrm{d}x \\
&= \sum_{i=0}^{n_{\mathrm{el}}-1} \int_{t_i}^{t_{i+1}} [\Pi_p(c u_p)]'|_{(t_i,t_{i+1})} v_p|_{(t_i,t_{i+1})} \,\mathrm{d}x \\
&= \sum_{i=0}^{n_{\mathrm{el}}-1} \Big([\Pi_p(c u_p)]'|_{(t_i,t_{i+1})}, v_p|_{(t_i,t_{i+1})}\Big)_p.
\end{aligned}$$

Therefore, equation (3.4) can be rewritten as

$$\begin{aligned}
\sum_{i=0}^{n_{\mathrm{el}}-1} \Big[-\Big([\Pi_p(\mu u_p')]'|_{(t_i,t_{i+1})}, v_p|_{(t_i,t_{i+1})}\Big)_p &+ \Big([\Pi_p(c u_p)]'|_{(t_i,t_{i+1})}, v_p|_{(t_i,t_{i+1})}\Big)_p \\
&+ \Big((\sigma u_p)|_{(t_i,t_{i+1})}, v_p|_{(t_i,t_{i+1})}\Big)_p \Big] \\
&= \sum_{i=0}^{n_{\mathrm{el}}-1} (f|_{(t_i,t_{i+1})}, v_p|_{(t_i,t_{i+1})})_p,
\end{aligned}$$

for all $v_p \in V_p$. Now, let us consider the global basis $\{\Phi_i\}_{i=1}^{pn_{\mathrm{el}}+1}$ of V_p as it is defined in equations (3.1) and (3.2). Then, we choose v_p to be any basis function of V_p, say Φ_{pi+j} for some $i = 0, \dots, n_{\mathrm{el}}-1$ and $j = 0, \dots, p-1$, $(i,j) \neq (0,0)$, since we know that $\Phi_{pi+j}(\tau_{k,l}) = \delta_{ij}\delta_{kl}$ by definition, for all $k = 0, \dots, n_{\mathrm{el}}-1$, $l = 0, \dots, p-1$, $(k,l) \neq (0,0)$, then

$$-[\Pi_p(\mu u_p')]'(\tau_{i,j}) + [\Pi_p(c u_p)]'(\tau_{i,j}) + (\sigma u_p)(\tau_{i,j}) = f(\tau_{i,j}),$$

according to the definition of the discrete Gauss-Legendre-Lobatto inner product.

Let $\mathcal{L}_p$ be the differential operator such that $\mathcal{L}_p g := -[\Pi_p(\mu g')]' + [\Pi_p(cg)]' + \sigma g$, for any suitable function g. Then the problem expressed with SEM G-NI in (3.4) is equivalent to the following collocation problem: find $u_p \in \mathbb{Q}_p(\Omega_h)$ such that

$$\begin{cases} \mathcal{L}_p u_p(\tau_{i,j}) = f(\tau_{i,j}), \text{ for } i = 0, \dots, n_{\text{el}} - 1,\ j = 0, \dots, p-1, (i,j) \neq (0,0), \\ u_p(a) = u_p(b) = 0. \end{cases}$$

Notice that the number of collocation points without repetitions is $n_{\text{el}}p + 1$, which is exactly the dimension of $\mathbb{Q}_p(\Omega_h)$.

In the case in which other boundary conditions are considered, they have to be carefully taken into account when integration and counter-integration by part is done. If another differential operator $\mathcal{L}$ is considered, the equivalent collocation problem is obtained by replacing $\mathcal{L}$ by the so-called pseudo-spectral operator $\mathcal{L}_p$, that is by substituting every derivative of $\mathcal{L}$ by the corresponding derivative of the Gauss-Legendre-Lobatto interpolation.

The method just presented is mainly based on the two following ingredients:

1. the presence of a high precision quadrature formula (Gauss-Legendre-Lobatto formula) to integrate the polynomials of $\mathbb{Q}_p(\Omega_h)$, where Ω_h is the partition of the geometrical domain and p is a given polynomial degree;

2. the existence of interpolant basis functions of $\mathbb{Q}_p(\Omega_h)$ (Legendre polynomials) that can be used to interpolate any function at the quadrature nodes of ingredient 1.

In the following, we try to mimic what has just been done with SEM G-NI and translate it to isogeometric analysis. Ingredient 2 is easier to obtain, so we first look for a high precision quadrature formula as in ingredient 1.

3.2 Minimizing the quadrature error with a minimum number of nodes

It has been seen in section 3.1 that for the diffusion-transport-reaction model problem (3.3), a high precision formula is needed in particular to compute the mass and the stiffness matrices. Consequently, let us concentrate on integrating as accurately as possible the stiffness and the mass matrices, with the least possible number of nodes. Since B-splines are piecewise polynomial functions, Gauss quadrature formulas are indeed valid but they require too many quadrature points. Indeed, to build a collocation method out of a quadrature formula as in section 3.1, the number of quadrature points has to be equal to the dimension of the space considered.

Let $\Omega = (a, b)$ still be the one dimensional domain considered. The following notation is used:

- p: degree of the underlying polynomials/B-splines;
- n_{el}: number of elements in the decomposition of the domain Ω;
- $\mathbb{B}_k^c(\Omega)$: set of B-splines of degree k that are globally C^c-continuous on a domain Ω, with $0 \leq c \leq k-1$.

Consider the space $\mathbb{B}_p^{p-1}(\Omega)$, let $n = p + n_{\text{el}}$ be its dimension, and let $\mathcal{B} = \{\phi_i\}_{i=1}^n$ be the basis given by Cox-de-Boor recursion formula, see equation (2.1) in section 2.1. We look for the best quadrature points so that $\int_a^b \phi_i \phi_j \,\mathrm{d}x$ and $\int_a^b \phi_i' \phi_j' \,\mathrm{d}x$ are as accurate as possible, for all

$i, j = 1, \dots, n$. Note that any product of two elements of $\mathcal{B}$ belongs to $\mathbb{B}_{2p}^{p-1}$, and any product of the derivative of two elements of $\mathcal{B}$ belongs to $\mathbb{B}_{2(p-1)}^{p-2}$. Consequently, if we call $\mathcal{B}_m$ a basis of $\mathbb{B}_{2p}^{p-1}$ and $\mathcal{B}_s$ a basis of $\mathbb{B}_{2(p-1)}^{p-2}$, it is enough to find the best quadrature formulas so that $\int_a^b \psi \,\mathrm{d}x$, $\forall \psi \in \mathcal{B}_m$ and $\int_a^b \varphi \,\mathrm{d}x$, $\forall \varphi \in \mathcal{B}_s$ are as accurate as possible. Note that the subscripts m and s stand respectively for mass and stiffness matrices; they are used to simplify the notation.

3.2.1 Mass matrix

Let us first concentrate on the mass matrix, that is let us minimize

$$\sqrt{\sum_{\psi \in \mathcal{B}_m} \left| \int_a^b \psi \,\mathrm{d}x - \sum_{i=1}^{n_q} \alpha_i \psi(\tau_i) \right|^2}, \tag{3.5}$$

whose unknowns are

- the number n_q of quadrature points;
- the quadrature points τ_i, $i = 1, \dots, n_q$;
- the quadrature weights α_i, $i = 1, \dots, n_q$.

Once the quadrature points and their number n_q are determined, the quadrature weights are

$$\alpha_i := \int_a^b \prod_{j=1, j \neq i}^{n_q} \frac{x - \tau_j}{\tau_i - \tau_j} \,\mathrm{d}x, \;\; \forall i = 1, \dots, n_q, \tag{3.6}$$

that is we are looking for a Lagrange quadrature formula. Let $N = p + n_{\text{el}}(p+1)$ be the dimension of $\mathbb{B}_{2p}^{p-1}$. The algorithm implemented to find the minimum of equation (3.5) is the following:

Input: $a, b, p, n_{\text{el}}, \text{tol}$.

1. Build $\mathcal{B}_m$;
2. For each $\psi \in \mathcal{B}_m$, compute the exact integral $\int_a^b \psi \,\mathrm{d}x$ thanks to the (non optimal but exact) Gauss-Legendre quadrature formula requiring $\lceil \frac{2p+1}{2} \rceil$ function evaluations on each element;
3. Loop over n_q ranging from 1 to N:
    a) Set conditions on the set of quadrature points τ to find: τ should be symmetric, containing values between a and b, sorted in increasing order;
    b) Loop over a certain number of trials n_t ranging from 1 to $n_t^{\max}$:
        (i) Let τ_0 be a random initial value of quadrature points τ;
        (ii) Find the minimum of the function `f2min` that, given as input a set of quadrature points τ, computes the corresponding weights α as in equation (3.6), and returns (3.5);
        (iii) Let τ_{n_t} be the argument of the minimum found;
    c) Define
    $$\tau_{n_q} := \arg \min_{\tau = \tau_1, \dots, \tau_{n_t^{\max}}} \texttt{f2min}(\tau_{n_t});$$

n_{el}	p	$2p$	$p-1$ (continuity)	n_q nodes	$n_q - 2$ interior nodes	n	N
1	1	2	0	3	1	2	3
	2	4	1	4	2	3	5
	3	6	2	5	3	4	7
	4	8	3	6	4	5	9
	5	10	4	7	5	6	11
2	1	2	0	4	2	3	5
	2	4	1	5	3	4	8
	3	6	2	7	5	5	11
	4	8	3	8	6	6	14
	5	10	4	10	8	7	17
3	1	2	0	6	4	4	7
	2	4	1	8	6	5	11
	3	6	2	10	8	6	15
	4	8	3	19(12)	17(10)	7	19
	5	10	4	17(13)	15(11)	8	23
4	1	2	0	8	6	5	9
	2	4	1	10	8	6	14
	3	6	2	19(12)	17(10)	7	19
	4	8	3	24(15)	22(13)	8	24
	5	10	4	29(16)	27(14)	9	29

Table 3.1: Minimal number of quadrature nodes needed to integrate exactly the mass matrix with tol $= 10^{-7}$, or between brackets tol $= 10^{-6}$.

```
    d) If f2min($\tau_{n_q}$) < tol, end the loop;
Output: $n_q, \tau_{n_q}$.
```

We would like to get $n_q \leq n$. Note that it is important to know how to deal with the boundary conditions. To stay in the same framework as section 3.1, consider a one-dimensional diffusion-transport-reaction problem with constant parameters and homogeneous Dirichlet boundary conditions that admits a unique solution, that is problem (3.3) with μ, c and σ constant. In this case, we have to consider all basis functions $\mathcal{B}_m$ and force a and b to be nodes of quadrature.

The algorithm just presented has been implemented in MATLAB thanks to the minimizer function `fmincon` [22]. Table 3.1 sums up the minimal number of quadrature nodes obtained when $n_{\text{el}} = 1, \dots, 4$ and $p = 1, \dots, 5$ are used, with a tolerance tol $= 10^{-7}$. Moreover, Figure 3.1 shows the distribution of the best quadrature nodes found, together with the basis functions $\mathcal{B}_m$ of $\mathbb{B}_{2p}^{p-1}$ for each p and each n_{el}. Since the minimization problem is not linear, the results for $n_{\text{el}} \geq 3$ and $p \geq 3$ may not be very precise and should be considered with care.

For $n_{\text{el}} = 1$, the quadrature nodes obtained are the Gauss-Legendre-Lobatto nodes. We could have expected this since on a single element, a B-spline is a polynomial and nothing more. In this case, $n_q = p + 2 = n + 1$, as it is already known for polynomials. Consequently, with one element, the mass matrix cannot be exactly integrated with $n_q = p + 1 = n$ quadrature points or fewer. When $n_{\text{eq}} > 1$, the situation is even worst: always more nodes are required and there is no possibility to find $n = n_q$ quadrature points to integrate exactly the mass matrix. The only case with $n_{\text{el}} > 1$ for which Gauss points can be recognized is when $n_{\text{el}} = 2$ and $p = 1$. Here, the points found are the Gauss-Radau points on each of both elements. Note that when $p = 1$, the space considered is $\mathbb{B}_2^0$ and the functions of this space are globally C^0-continuous, as the basis functions used in the spectral element method. However, Gauss quadrature nodes

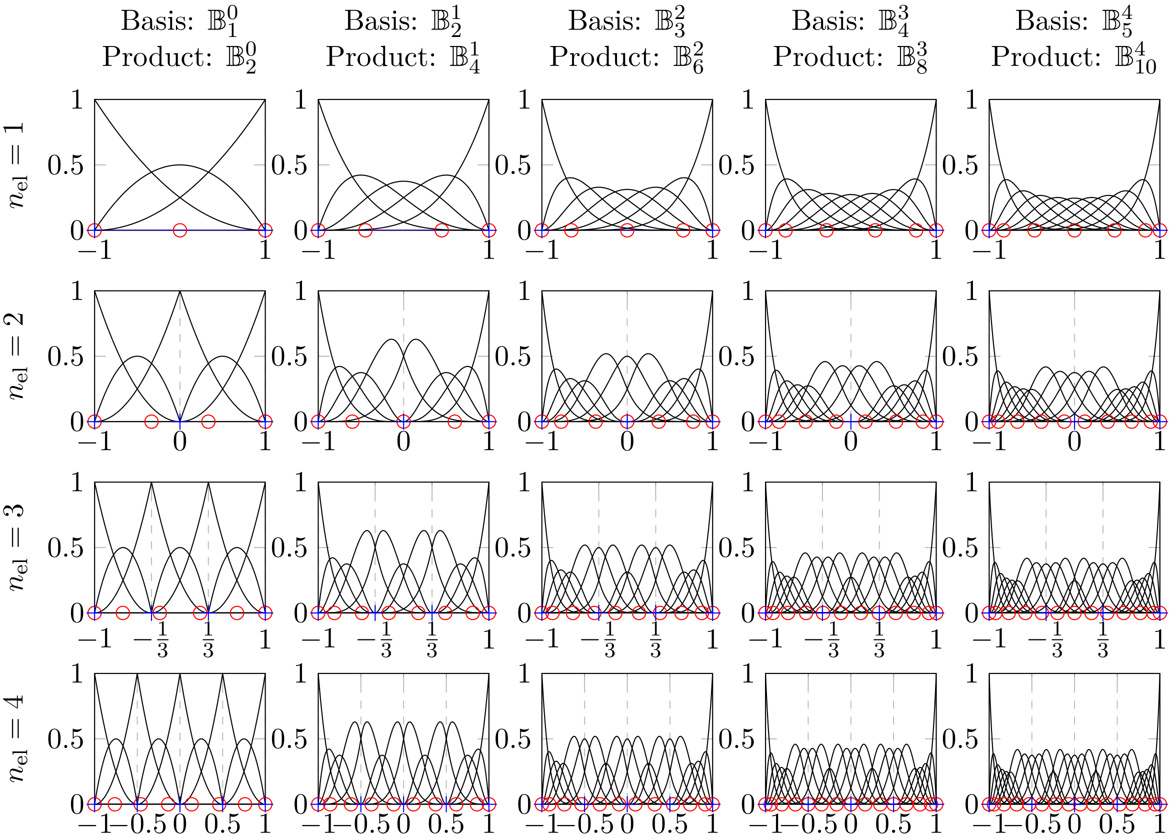


Figure 3.1: Distribution of the nodes minimizing the quadrature error for the mass matrix (red circles ○), with the respective basis $\mathcal{B}_m$ (black curves -). Blue crosses + separate elements.

should not necessarily be expected on each element since those quadrature nodes are optimal for discontinuous functions at the elements' boundaries, and not for C^0-continuous functions.

Different initial values τ_0 of point 3.b)(i) of the algorithm have been tried, such as equally spaced points in (a, b) or the n_q Gauss-Legendre-Lobatto points on the whole domain (a, b), but the results do not significantly change. Also, if the basis functions considered in the sum of equation (3.5) are only the basis functions that are zero on the boundaries a and b, the exact same results are obtained, that is the algorithm finds the same quadrature points.

3.2.2 Stiffness matrix

Now, let us see what happens when the stiffness matrix is considered instead of the mass matrix, that is let us minimize

$$\sqrt{\sum_{\varphi \in \mathcal{B}_s} \left| \int_a^b \varphi \, \mathrm{d}x - \sum_{i=1}^{n_q} \alpha_i \varphi(\tau_i) \right|^2}, \tag{3.7}$$

whose unknowns are still the number n_q of quadrature points, the quadrature points τ_i, for $i = 1, \ldots, n_q$, and the quadrature weights α_i, $i = 1, \ldots, n_q$. The whole procedure and the algorithm used are exactly the same as for the mass matrix, provided we replace $\mathbb{B}_{2p}^{p-1}$ by $\mathbb{B}_{2(p-1)}^{p-2}$ and $\mathcal{B}_m$ by $\mathcal{B}_s$. Moreover, in this case, the exact Gauss-Legendre quadrature formula used in step 2 requires $\left\lceil \frac{2p-1}{2} \right\rceil$ function evaluations on each element instead of $\left\lceil \frac{2p+1}{2} \right\rceil$.

n_{el}	p	n_q nodes	$n_q - 2$ interior nodes	n	N
1	2	3	1	3	3
	3	4	2	4	5
	4	5	3	5	7
	5	6	4	6	9
2	2	4	2	4	5
	3	5	3	5	8
	4	7	5	6	11
	5	8	6	7	14
3	2	6	4	5	7
	3	10(8)	8(6)	6	11
	4	15(10)	13(8)	7	15
	5	*	*	8	19
4	2	8	6	6	9
	3	14(10)	12(8)	7	14
	4	*	*	8	19
	5	*	*	9	24

Table 3.2: Minimal number of quadrature nodes needed to integrate exactly the stiffness matrix with tol $= 10^{-7}$, or between brackets tol $= 10^{-6}$. Asterisks * correspond to cases in which the algorithm did not converge, that is for which the quadrature error has stayed above 10^{-6} for all $n_q = 1, \ldots, N-1$, where N is the number of degrees of freedom of $\mathbb{B}^{p-2}_{2(p-1)}$.

Again, the algorithm has been implemented in Matlab thanks to the minimizer function `fmincon`. Table 3.2 sums up the minimal number of quadrature nodes obtained when we use $n_{\text{el}} = 1, \ldots, 4$ and $p = 1, \ldots, 5$, with a tolerance tol $= 10^{-7}$. Moreover, Figure 3.2 shows the distribution of the best quadrature nodes found, together with the basis functions $\mathcal{B}_s$ of $\mathbb{B}^{p-2}_{2(p-1)}$ for each p and each n_{el}. As previously, since the minimization problem is not linear, the results for $n_{\text{el}} \geq 3$ and $p \geq 3$ may not be very precise and should be considered with care.

Still, for $n_{\text{el}} = 1$, the quadrature nodes obtained are the Gauss-Legendre-Lobatto nodes since on a single element, a B-spline is a polynomial and nothing more. In this case, $n_q = p + 1 = n$, as it is already known for polynomials. Consequently, with one element, the stiffness matrix can be exactly integrated with exactly $n_q = p + 1 = n$ quadrature points. When $n_{\text{eq}} > 1$, the situation is getting worst: always more nodes are required and there is no possibility to find $n = n_q$ quadrature points to integrate exactly the mass matrix.

Let us recall that the support of any B-spline basis function of order k defined by Cox-de-Boor formula is equal to $k+1$ knot spans. Moreover, the k first and the k last such B-spline basis functions are influenced by the boundary since more than one boundary knot are used to define them. Consequently, both for the mass and the stiffness matrices, we would expect to see a certain repetition pattern appear far from the boundaries when the number of basis functions n is larger than $4p$ for the mass matrix, or when $n > 4(p-1)$ for the stiffness matrix. Let us remember that the number of basis functions of $\mathbb{B}^c_k$ defined on n_{el} elements is $(k+1) + (n_{\text{el}} - 1)(k - c)$. For the mass matrix, the space $\mathbb{B}^{p-1}_{2p}$ is considered, so in this case $n = (2p+1) + (n_{\text{el}} - 1)(p+1)$. Consequently, a repetition pattern could be seen for $n_{\text{el}} \geq 3$. For the stiffness matrix, the considered space is $\mathbb{B}^{p-2}_{2(p-1)}$, so in this case, $n = (2p-1) + (n_{\text{el}} - 1)p$. Therefore, a repetition pattern could be also seen for $n_{\text{el}} \geq 3$. However, already when $p = 1$ in the case in which the mass matrix is considered, we can easily see that it is not the case. Indeed, when $n_{\text{el}} = 3$, two nodes are present in the middle element and are symmetrically located with

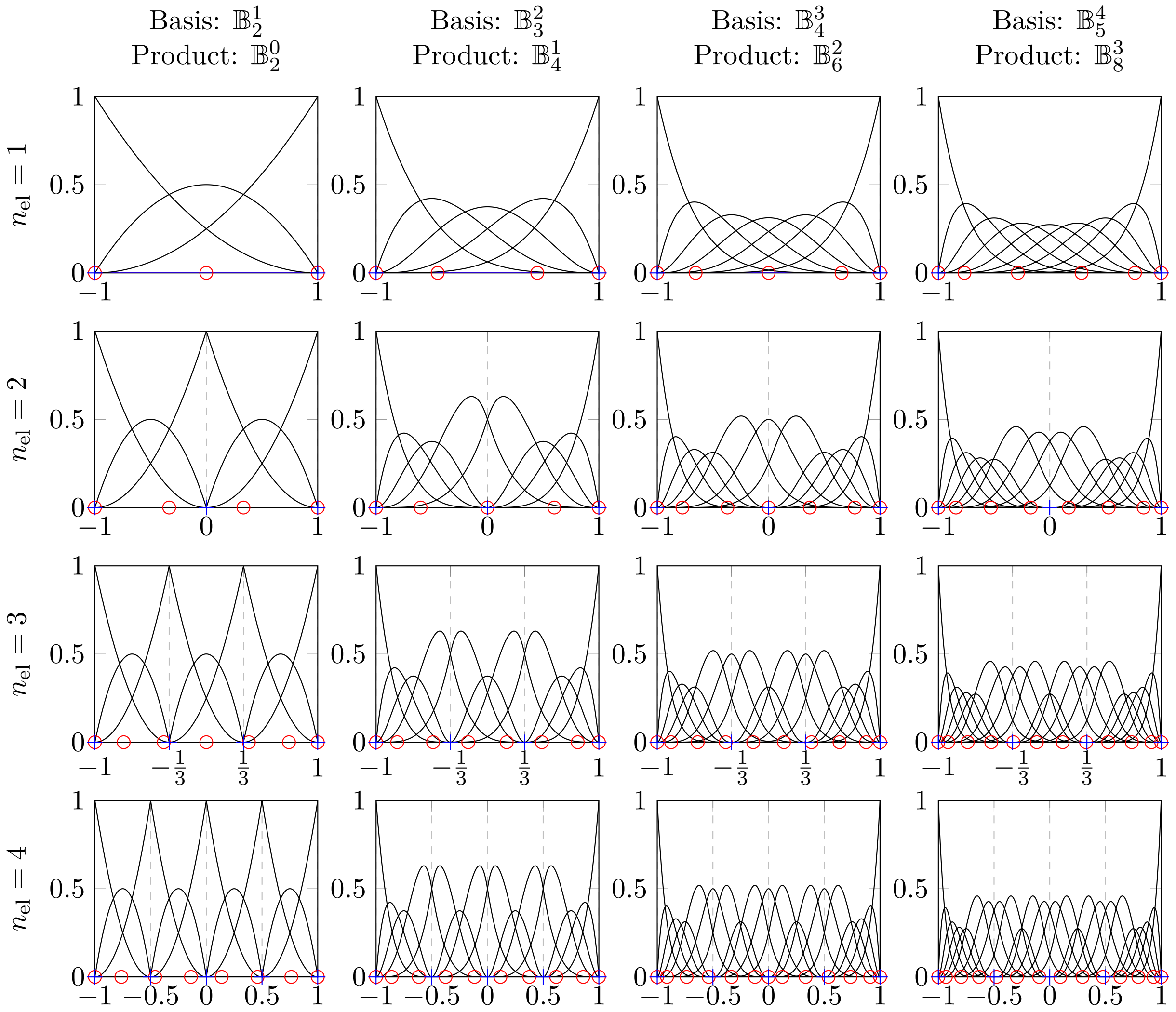


Figure 3.2: Distribution of the nodes minimizing the quadrature error for the stiffness matrix (red circles ○), with the respective basis $\mathcal{B}_s$ (black curves -). Elements are separated by the blue crosses +.

respect to the center of the element. When $n_{\mathrm{el}} = 4$, two nodes are still present in each of the two middle elements, but they are not symmetrically distributed inside their respective element. Instead, they are pushed towards the boundary. When the stiffness matrix is considered with $p = 2$, the result is even worst: only one quadrature node is present in the middle element when $n_{\mathrm{el}} = 3$, while two nodes are present in the two middle elements when $n_{\mathrm{el}} = 4$.

The results above can be compared with the results of Hughes, Reali and Sangalli in [19]. In this paper, the authors try to find efficient quadrature rules for isogeometric analysis in a similar way as it has been done previously in this section, that is by means of minimizing the quadrature error, but the aim of this paper is not to find an IGA collocation method. As a consequence, the boundary nodes a and b are not forced to be quadrature points. Moreover, their study is only done on a biunit interval, that is when $n_{\mathrm{el}} = 2$. If in our algorithm, we do not force a and b to be quadrature points anymore, then we find the same results as Hughes and al. when $n_{\mathrm{el}} = 2$, that is we get a number of collocation points $n_q = \lceil \frac{N}{2} \rceil$, where N is the number of degrees of freedom of the B-spline space considered. The coordinates of the optimal quadrature points in this specific case can be found in Tables 8 and 10 of [19], when $p = 1$ and $p = 2$, respectively.

Therefore, this confirms the results presented in this section. Unfortunately, it has just been shown that there does not exist any quadrature formula obtained by minimization of the quadrature error whose minimum number of quadrature nodes is equal to the number of basis functions of the considered space, and which computes exactly either the mass or the stiffness matrix of an isogeometric problem.

3.3 Convergence of the quadrature error

In the previous section 3.2, it has been shown that there does not exist any quadrature rule that computes exactly the mass and the stiffness matrices of an isogeometric problem with a number of nodes equal to the dimension of the B-spline space considered. However, this is also the case when basis functions of the spectral element method are used. Indeed, in SEM G-NI, the stiffness matrix is computed exactly, but the mass matrix is not, since Gauss-Legendre-Lobatto quadrature formula has degree of exactness equal to $2p-1$, where p is the degree of the underlying polynomials. Nonetheless, the error introduced by the numerical integration with Gauss-Legendre-Lobatto quadrature formula converges towards zero. An estimation of the rate of convergence is given in [27]: let $f \in H^s(a,b)$ with $s \geq 1$, and let $I_n^{\mathrm{GLL}} f$ be the approximation of the integral of f between a and b obtained with the Gauss-Legendre-Lobatto quadrature formula using $n+1$ nodes. Then

$$e_n := \left| \int_a^b f(x)\,\mathrm{d}x - I_n^{\mathrm{GLL}} f \right| \leq C \left(\frac{1}{n}\right)^s \|f\|_{H^s(a,b)}, \tag{3.8}$$

where C is a constant independent from n but which could depend on s. In particular, if f is the product of two polynomials of degree p, as it would be if the mass matrix were computed, then e_n would tend to 0 when n tends to infinity.

In this section, the idea is to consider the quadrature formula obtained thanks to the algorithm presented in section 3.2 when we fix the number of quadrature nodes n_q to be the dimension of the B-spline space n. Then, we want to see if it gives a quadrature error that converges to 0 when the B-spline order p or the number of elements n_{el} increases. In a few words, we try to see if a property similar to inequality (3.8) holds for the isogeometric analysis. Note that since in isogeometric analysis $n = p + n_{\mathrm{el}}$, increasing either p or n_{el} by one will only increase n and thus the number of quadrature points by 1.

The following Figure 3.3 shows the evolution of the minimum quadrature error when $n_q = n$ quadrature points are used, on the mass matrix, computed as in equation (3.5), with respect to the number of elements n_{el} for some p fixed, pr with respect to p for some n_{el} fixed. In Figure 3.4, this evolution of the error is represented when the quadrature error is computed on the stiffness matrix instead of the mass matrix, as in equation (3.7).

Results are very disappointing. Indeed, in both cases, no convergence is observed at all when the number of elements is increased. Instead, the quadrature error increases with n_{el} and seems to converge towards a value close to 10^{-1} for any value of p. Moreover, when p is increased, the quadrature error converges, but a lot too slowly, except when $n_{\mathrm{el}} = 1$. In this case indeed, quadrature points are the Gauss-Legendre-Lobatto points, so the quadrature formula found verifies equation (3.8). This convergence is observed very well in the case in which the mass matrix is considered, but not when the stiffness matrix is considered. This is due to round-off errors, machine precision has been attained.

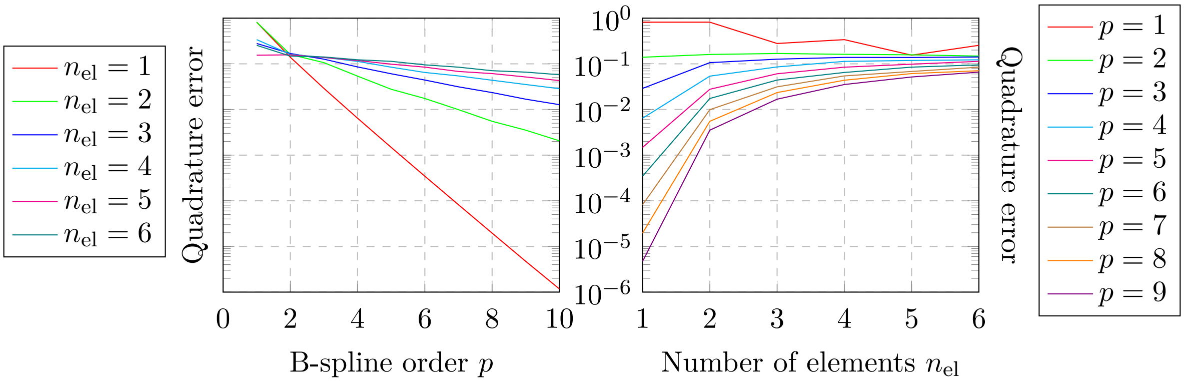


Figure 3.3: Evolution of the quadrature error of optimal quadrature formulas found to numerically integrate the mass matrix.

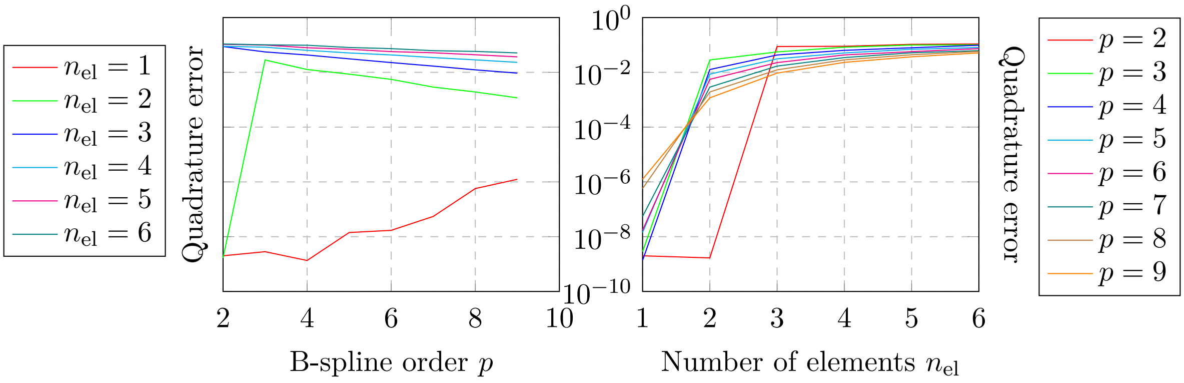


Figure 3.4: Evolution of the quadrature error of optimal quadrature formulas found to numerically integrate the stiffness matrix.

Consequently, fixing $n_q = n$ and launching the algorithm introduced in section 3.2 does not lead to a convergent and usable quadrature formula.

3.4 Least-squares collocation method

In section 3.2, an algorithm has been introduced to find the minimum number of quadrature nodes used to integrate exactly any space of B-splines. To seek the solution of a differential problem such as problem (3.3) in a certain space $\mathbb{B}_p^{p-1}(\Omega)$, $p \geq 1$, where Ω is the B-spline geometry considered, it is required to numerically integrate the mass and the stiffness matrices, that is functions belonging to $\mathbb{B}_{2p}^{p-1}(\Omega)$ and to $\mathbb{B}_{2(p-1)}^{p-2}(\Omega)$ respectively. It has been found that the number of quadrature nodes required to integrate such functions is greater than the dimension of the original space $\mathbb{B}_p^{p-1}(\Omega)$. Therefore, a collocation method derived from those quadrature formulas, as it has been done for the spectral element method (see section 3.4), can not exist.

However, having more nodes than required, a least-squares collocation method can be created, following roughly the idea of Anitescu and al. in [1] with different collocation points. Let us see if the quadrature points that have been found in section 3.2 can serve as collocation points in a least-squares sense. More precisely, let us consider a one dimensional simple Laplace problem (second order differential problem) with homogeneous Dirichlet boundary conditions as follows:

$$\begin{cases} -u''(x) = (m\pi)^2 \sin(m\pi x) \text{ in } (-1,1), \\ u(-1) = u(1) = 0, \end{cases} \tag{3.9}$$

with $m \in \mathbb{N}$. The exact solution of this differential problem is $u_{ex}(x) = \sin(m\pi x)$. Let n_{el} be the number of elements on which the B-spline geometry $\Omega := (-1,1)$ is built. Let $p \in \mathbb{N} \setminus \{0\}$ be the order of the B-spline basis in which the solution is sought, that is the space of solutions considered is $\mathbb{B}_p^{p-1}(-1,1)$. Let $\mathcal{B} = \{N_i\}_{i=1}^n$ be a basis of $\mathbb{B}_p^{p-1}(-1,1)$, where $n = n_{\text{el}} + p$ is the dimension of this space.

The solution can thus be written as $u = \sum_{i=1}^n u_i N_i$. Let us recall that the weak formulation of problem (3.9) writes: Find $u \in H_0^1(\Omega)$ such that for all $v \in H_0^1(\Omega)$,

$$\int_{-1}^1 u'v' \,\mathrm{d}x = \int_{-1}^1 fv \,\mathrm{d}x.$$

We thus have to deal with the stiffness matrix formed by the integrals

$$\int_{-1}^1 N_i' N_j' \,\mathrm{d}x, \quad \forall i, j = 1, \ldots, n.$$

Consequently, let $\{\tau_i\}_{i=1}^{n_q}$ be the quadrature points found in section 3.2 that are optimal to compute exactly the stiffness matrix in this case. By optimal, we mean the quadrature formula that uses the minimum number of quadrature points. We order the quadrature points so that the boundary points verify $\tau_1 = -1$ and $\tau_{n_q} = 1$. Consequently, the collocation problem of equation 3.9 writes

$$\begin{cases} -\sum_{i=1}^n u_i N_i''(\tau_j) = (m\pi)^2 \sin(m\pi\tau_j), \quad \forall i = 1, \ldots, n, \forall j = 2, \ldots, n_q - 1, \\ \sum_{i=1}^n u_i N_i(\tau_1) = \sum_{i=1}^n u_i N_i(\tau_{n_q}) = 0. \end{cases}$$

Written in matricial form, this is equivalent to $A\mathbf{u} = \mathbf{f}$, where $A \in \mathbb{R}^{n_q \times n}$, $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{f} \in \mathbb{R}^{n_q}$ and

$$A := \begin{pmatrix} N_1(\tau_1) & \ldots & N_n(\tau_1) \\ N_1''(\tau_2) & \ldots & N_n''(\tau_2) \\ \vdots & \ddots & \vdots \\ N_1''(\tau_{n_q-1}) & \ldots & N_n''(\tau_{n_q-1}) \\ N_1(\tau_{n_q}) & \ldots & N_n''(\tau_{n_q}) \end{pmatrix}, \quad \mathbf{u} := \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \text{ and } \mathbf{f} := \begin{pmatrix} 0 \\ (m\pi)^2 \sin(m\pi\tau_2) \\ \vdots \\ (m\pi)^2 \sin(m\pi\tau_{n_q-1}) \\ 0 \end{pmatrix}$$

Recall that $n_q \geq n$ as it has been seen in section 3.2. The principle of this least-squares collocation method is to solve this system in the least-squares sense. However, we have to impose strongly the boundary conditions, and not only in the least-squares sense, that is, we need to introduce a discrete form of lifting. But in equation (3.9), boundary conditions are homogeneous and consequently, we impose $u_1 = u_n = 0$ since N_1 and N_n are the only non-zero basis functions at the boundaries -1 and 1 (see the Cox-de-Boor formula (2.1)). Then we solve the reduced system $\tilde{A}\tilde{\mathbf{u}} = \tilde{\mathbf{b}}$, with $\tilde{A}$ being A without its first and last rows and columns, $\tilde{\mathbf{u}}$ being $\mathbf{u}$ without its first and last entries, and $\tilde{\mathbf{b}}$ also being $\mathbf{b}$ without its first and last entries.

Figure 3.5 shows the numerical solution in multiple cases, that is when $m = 1$ or when $m = 2$, for $p = 2$ and different values of n_{el}, and for $n_{\text{el}} = 2$ and different values of p. We can already see without computing the numerical error that when $m = 1$, we do not have a better approximation of the solution when $n_{\text{el}} = 3$ than when $n_{\text{el}} = 2$, and we do not have neither a better approximation of the solution when $p = 3$ than when $p = 2$. However, the solution seems

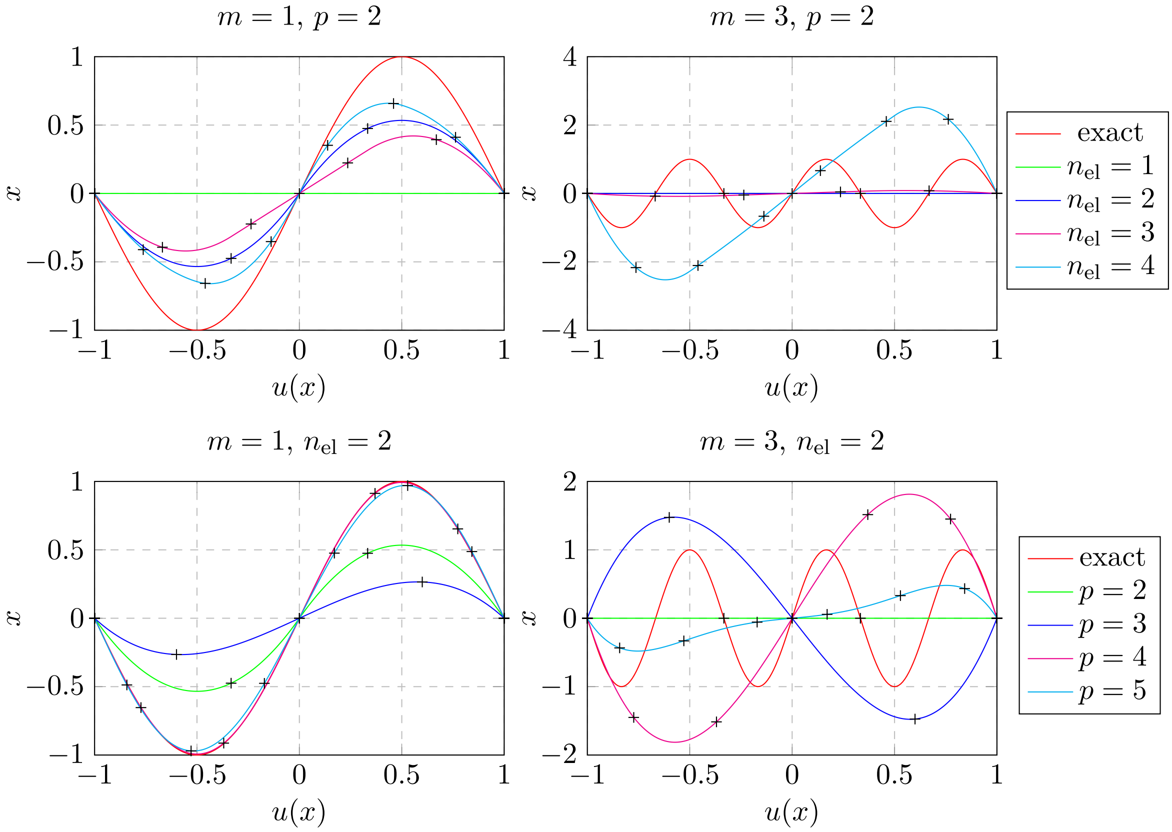


Figure 3.5: Comparison of the exact and the numerical solution found with the least-squares collocation method for $m = 1$ and $m = 3$, and for $p = 2$ and different values of n_{el}, or $n_{\text{el}} = 2$ and different values of p in the first and second rows respectively. The crosses $+$ correspond to the collocation points.

to converge. Instead, when $m = 3$, no convergence is observed at all and thus the least-squares approximation does not lead to satisfying results.

Consequently, as it has already been seen in the previous sections, we have not found a way to build any well convergent isogeometric collocation method from a Gauss quadrature formula as it has been proposed in section 3.2, not even through a least-squares problem.

3.5 Maximizing the degree of exactness given a number of integration nodes

In the previous sections of this chapter, we have first tried to find the minimum number of quadrature nodes required to minimize the quadrature error, and then, we have looked at the evolution of the quadrature error when we fix the number of quadrature nodes to be equal to the number of degrees of freedom of our B-spline space. In this section, we will approach the problem from another point of view: the idea is to maximize the degree of exactness of a quadrature formula, given a certain number of quadrature points. Let us use the same notation as in section 3.2.

The problem is formulated in the following way: let us fix the number of elements n_{el} in the decomposition of the one dimensional domain $\Omega = (a, b)$, and let n_q be a fixed number of quadrature points. Then, the goal is to find the largest positive integer p such that there exist n_q

quadrature points τ_i, $i = 1, \dots, n_q$, and n_q corresponding quadrature weights α_i, $i = 1, \dots, n_q$, such that the mass or the stiffness matrix is exactly computed thanks to the quadrature formula derived from them, where the basis functions belong to $\mathbb{B}_p^{p-1}$. Note that once the set $\{\tau_i\}_{i=1}^{n_q}$ is determined, the corresponding quadrature weights are given by equation (3.6). We now separate the cases in which the mass matrix and the stiffness matrix are considered. In a similar way as in section 3.2, let $\mathcal{B}_m^p$ be a basis of $\mathbb{B}_{2p}^{p-1}$, and $\mathcal{B}_s^p$ be a basis of $\mathbb{B}_{2(p-1)}^{p-2}$, for a given $p \in \mathbb{N} \setminus \{0\}$, where the subscripts m and s stand respectively for mass and stiffness matrices.

3.5.1 Mass matrix

Let us first concentrate on the mass matrix, that is we want to find the largest positive integer p such that there exist $\{\tau_i\}_{i=1}^{n_q}$ and their corresponding weights $\{\alpha_i\}_{i=1}^{n_q}$ that verify

$$\epsilon_p := \sqrt{\sum_{\psi \in \mathcal{B}_m^p} \left| \int_a^b \psi \, \mathrm{d}x - \sum_{i=1}^{n_q} \alpha_i \psi(\tau_i) \right|^2} = 0. \tag{3.10}$$

The algorithm implemented to solve this problem is the following:

```
Input: $a, b, n_{\text{el}}, n_q, \text{tol}$.

1. Initialization: $p = 0$, $\epsilon = 0$;
2. While $\epsilon < \text{tol}$, do:
    a) Set $p = p + 1$;
    b) Build $\mathcal{B}_m^p$;
    c) For each $\psi \in \mathcal{B}_m^p$, compute the exact integral $\int_a^b \psi \, \mathrm{d}x$ thanks to the (non optimal but
       exact) Gauss-Legendre quadrature formula requiring $\lceil \frac{2p+1}{2} \rceil$ function evaluations on
       each element;
    d) Set conditions on the set of quadrature points $\tau$ to find: $\tau$ should be symmetric,
       containing values between $a$ and $b$, sorted in increasing order;
    e) Loop over a certain number of trials $n_t$ ranging from 1 to $n_t^{\max}$:
        (i) Let $\tau_0$ be a random initial value of quadrature points $\tau$;
        (ii) Find the minimum of the function f2min that, given as input a set of quadrature
             points $\tau$, computes the corresponding weights $\alpha$ as in equation (3.6), and returns
             $\epsilon$ as in equation (3.10);
        (iii) Let $\epsilon$ be the the minimum found;
3. Decrement $p$ by 1 to get the last value of $p$ such that $\epsilon < \text{tol}$;

Output: $p$.
```

Our hope is to get $n_q \leq n_{\text{el}} + p$, since $n_{\text{el}} + p$ is the number of degrees of freedom of the B-spline space $\mathbb{B}_p^{p-1}$ on n_{el} elements. Note that it is important to know how to deal with the boundary conditions. As in section 3.2, and to stay in the same framework as section 3.1, let us consider a well defined one-dimensional diffusion-transport-reaction problem with constant parameters and homogeneous Dirichlet boundary conditions, that is problem 3.3 with μ, c and σ constant. In this case, we have to consider all basis functions of $\mathcal{B}_m^p$ in the sum of equation (3.10) and force a and b to be nodes of quadrature. Consequently, n_q has to be greater than 2.

		n_q							
		3	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n_{el}	**1**	1	2	3	4	5	6	7	8
	2	–	1	2	2	3	4	4	5
	3	–	–	–	1	1	2	1	3
	4	–	–	–	–	–	1	1	2

(a) Maximal B-spline order p.

		n_q							
		3	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n_{el}	**1**	2	3	4	5	6	7	8	9
	2	–	3	4	4	5	6	6	7
	3	–	–	–	4	4	5	4	6
	4	–	–	–	–	–	5	5	6

(b) Dimension of the largest B-spline space $\mathbb{B}_p^{p-1}$ on n_{el} elements.

Table 3.3: Maximal B-spline order p such that the mass matrix is exactly integrated by a quadrature formula of n_q nodes, and the dimension (number of degrees of freedom) of the corresponding B-spline space $\mathbb{B}_p^{p-1}$ on n_{el} elements.

The algorithm just presented has been implemented in Matlab thanks to the minimizer function `fmincon` [22]. Table 3.3a gives the largest p in each case, for n_{el} taking values from 1 to 4, and for n_q taking values from 3 to 10. Moreover, Table 3.3b gives the corresponding degrees of freedom of $\mathbb{B}_p^{p-1}$ on n_{el} elements, where p is the value given in Table 3.3a.

Note that for $n_{\text{el}} = 1$, p is equal to $n_q - 2$ and we could have expected this result. Indeed, in this case, B-splines are simple polynomials, and it is well known that the quadrature formula that contains the domain boundaries as quadrature nodes, and that gives the largest degree of exactness, is the Gauss-Legendre-Lobatto quadrature formula [26]. Moreover, we know that such quadrature formula on n_q nodes has degree of exactness equal to $2n_q - 3$. Consequently, since we want to compute the mass matrix, then we need to integrate every product of two basis functions of $\mathbb{B}_p^{p-1}$. That is, on a single interval, we need to integrate a polynomial of degree $2p$. Therefore, p verifies $2p \leq 2n_q - 3$, so $p \leq n_q - \frac{3}{2}$. Thus the maximal value of such p is $n_q - 2$, since p is a positive integer. And the value of the quadrature nodes found when $n_{\text{el}} = 1$ are indeed the n_q Gauss-Legendre-Lobatto quadrature nodes.

Moreover, the value of p when $n_q = 9$ and $n_{\text{el}} = 3$ does not seem to be coherent and should be taken into account with care. This error can be caused by the minimization function that has not converged. Furthermore, it is interesting to notice that the values found are very coherent with the ones of Table 3.1. Indeed, when we look at the same value of n_{el}, the value of n_q in Table 3.1 always corresponds to the same value of p as in Table 3.3a, when this n_q is present. Moreover, the quadrature nodes found are also the same as the ones found in section 3.2.1. But again, this shows that the results are not the ones we were hoping for. Indeed, when we look at Table 3.3b, we notice that the dimension of the space $\mathbb{B}_p^{p-1}$ on n_{el} elements, where p is the value returned by the algorithm, is always strictly smaller than the chosen values of n_q.

3.5.2 Stiffness matrix

Now, let us see what happens when the stiffness matrix is considered instead of the mass matrix. In this case, we want to find the largest positive integer p such that there exist $\{\tau_i\}_{i=1}^{n_q}$ and their corresponding weights $\{\alpha_i\}_{i=1}^{n_q}$ that verify

$$\epsilon_p := \sqrt{\sum_{\psi \in \mathcal{B}_s^p} \left| \int_a^b \psi \, \mathrm{d}x - \sum_{i=1}^{n_q} \alpha_i \psi(\tau_i) \right|^2} = 0.$$

The whole procedure and the algorithm used are exactly the same as for the mass matrix, provided we replace $\mathbb{B}_{2p}^{p-1}$ by $\mathbb{B}_{2(p-1)}^{p-2}$ and $\mathcal{B}_m^p$ by $\mathcal{B}_s^p$. Moreover, in this case, the exact

		n_q							
		3	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n_{el}	**1**	2	3	4	5	6	7	8	9
	2	–	2	3	3	4	5	5	6
	3	–	–	1	2	3	3	2	5
	4	–	–	–	1	1	2	3	3

(a) Maximal B-spline order p.

		n_q							
		3	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n_{el}	**1**	3	4	5	6	7	8	9	10
	2	–	4	5	5	6	7	7	8
	3	–	–	4	5	6	6	5	8
	4	–	–	–	5	5	6	7	7

(b) Dimension of the largest B-spline space $\mathbb{B}_p^{p-1}$ on n_{el} elements.

Table 3.4: Maximal B-spline order p such that the stiffness matrix is exactly integrated by a quadrature formula of n_q nodes, and the dimension (number of degrees of freedom) of the corresponding B-spline space $\mathbb{B}_p^{p-1}$ on n_{el} elements.

Gauss-Legendre quadrature formula used in step 2c requires $\left\lceil \frac{2p-1}{2} \right\rceil$ function evaluations on each element instead of $\left\lceil \frac{2p+1}{2} \right\rceil$.

Again, the algorithm has been implemented in Matlab thanks to the minimizer function `fmincon`. As in the case of the mass matrix, Table 3.4a gives the largest p in each case, for n_{el} taking values from 1 to 4, and for n_q taking values from 3 to 10. Moreover, Table 3.4b gives the corresponding degrees of freedom of $\mathbb{B}_p^{p-1}$ on n_{el} elements, where p is the value given in Table 3.4a.

Note that for $n_{\mathrm{el}} = 1$, p is equal to $n_q - 1$ and we could have expected this result. Indeed, in this case and as it has already been underlined when analysing the mass matrix, B-splines are simple polynomials, and it is well known that the quadrature formula that contains the domain boundaries as quadrature nodes, and that gives the largest degree of exactness, is the Gauss-Legendre-Lobatto quadrature formula [26]. Moreover, we know that such quadrature formula on n_q nodes has degree of exactness equal to $2n_q - 3$. Consequently, since we want to compute the stiffness matrix, then we need to integrate every product of the first derivative of two basis functions of $\mathbb{B}_p^{p-1}$. That is, on a single interval, we need to integrate a polynomial of degree $2(p-1)$. Therefore, p verifies $2(p-1) \leq 2n_q - 3$, so $p \leq n_q - \frac{1}{2}$. Thus the maximal value of such p is $n_q - 1$, since p is a positive integer. And the value of the quadrature nodes found when $n_{\mathrm{el}} = 1$ are indeed the n_q Gauss-Legendre-Lobatto quadrature nodes.

Moreover, the value of p when $n_q = 9$ and $n_{\mathrm{el}} = 3$ does not seem to be coherent and should be taken into account with care. This error can be caused by the minimization function that has not converged. Furthermore, it is again interesting to notice that the values found are very coherent with the ones of Table 3.2. Indeed, when we look at the same value of n_{el}, the value of n_q in Table 3.2 almost always corresponds to the same value of p as in Table 3.3a, when this n_q is present. Moreover, the quadrature nodes found are also the same as the ones found in section 3.2.2. But again, this shows that the results are not the ones we were hoping for. Indeed, when we look at Table 3.4b, we notice that the dimension of the space $\mathbb{B}_p^{p-1}$ on n_{el} elements, where p is the value returned by the algorithm, is always strictly smaller than the chosen values of n_q, except when $n_{\mathrm{el}} = 1$, or when $n_{\mathrm{el}} = 2$ and n_q is very small.

3.6 Partial conclusions

Therefore, we have found quadrature formulas thanks to the minimization of the quadrature error, but we have not been able to retrieve a collocation method from it since the number of quadrature points needed is larger than the dimension of the problem. Moreover, forcing the number of quadrature points to be equal to the number of degrees of freedom does not lead to a convergent quadrature error. Consequently, we cannot build a convergent collocation method from it. And finally, solving the collocation problem in a least-squares sense does not lead to a convergent method neither. The only case for which it works is when $n_{\mathrm{el}} = 1$ since in this case, isogeometric analysis and the spectral element method are equivalent [26]. We should thus change point of view on the problem.

Chapter 4

Gauss-Lobatto Lagrange collocation method

It is well known that the Galerkin spectral element method with numerical integration (also known as SEM G-NI) is equivalent to the spectral collocation method on the Gauss-Legendre-Lobatto nodes on each element [7, 8], as it has already been recalled in Chapter 3. Moreover, Nguyen and Schillinger in [24] have used an extraction operator that links the Gauss-Lobatto Lagrange functions, used as basis functions in SEM G-NI, with B-splines. This extraction operator has been firstly introduced by Schillinger and al. in [29]. Thanks to this operator, they are able to develop a new isogeometric collocation method that links the geometric flexibility and the improved approximation properties of IGA, formation efficiency of the collocation methods and the accuracy and robustness of the Galerkin methods. However, the full advantage of IGA is not used, and more specifically, the higher continuity of the B-spline basis functions is not fully exploited. In this chapter, we will combine the idea of Nguyen and Schillinger in [24] together with the high order continuity of the B-splines.

4.1 Gauss-Lobatto Lagrange extraction of B-splines

In this section, the idea of Nguyen and Schillinger in [24] is briefly presented. The aim is to find a link between the smooth B-splines and Gauss-Lobatto Lagrange polynomials. Since collocation points as well as Gauss-Legendre-Lobatto points are constructed by means of tensor-products of the one dimensional rule, we essentially concentrate on the one dimensional case, as it has been done in the previous chapters.

Let Ω be a one dimensional B-spline domain, that is Ω is a B-spline curve. Then, let $\hat{\Omega} = (a, b)$, $a, b \in \mathbb{R}$, be the parametrization space of Ω. Ω, and thus also $\hat{\Omega}$, are discretized into n_{el} elements E_i with $i = 0, \ldots, n_{\text{el}-1}$, where $E_i := (x_i, x_{i+1}) \subset \hat{\Omega}$, $x_0 = a$ and $x_{n_{\text{el}}} = b$. Let $p \in \mathbb{N} \setminus \{0\}$ be the order of the B-spline domain Ω. In each element E_i, consider the $p+1$ Gauss-Legendre-Lobatto nodes τ_{ij}, $j = 1, \ldots, p+1$ and the corresponding Gauss-Lobatto Lagrange basis functions $L_{i(p+1)+j}$, that is $L_{i(p+1)+j}$ is a polynomial of degree p on E_i, it is identically 0 on $\Omega \setminus E_i$, and it verifies

$$L_{i(p+1)+j}(\tau_{ik}) = \delta_{jk}, \tag{4.1}$$

for all $k = 1, \ldots, p+1$ (for a more detailed presentation of such functions, see section 3.1). A visual representation of this notation is given in Figure 4.1 in the case $p = 3$, $n_{\text{el}} = 2$. Write $\mathbf{L}$ the function vector containing all those Lagrange basis functions in order.

Let $\mathbb{B}_p(\Omega)$ be the B-spline space of interest. No specific continuity is specified between elements, it can be of any type. Then, let $\mathbf{N} = \{N_i\}_{i=1}^{n}$ be the set of B-spline basis functions

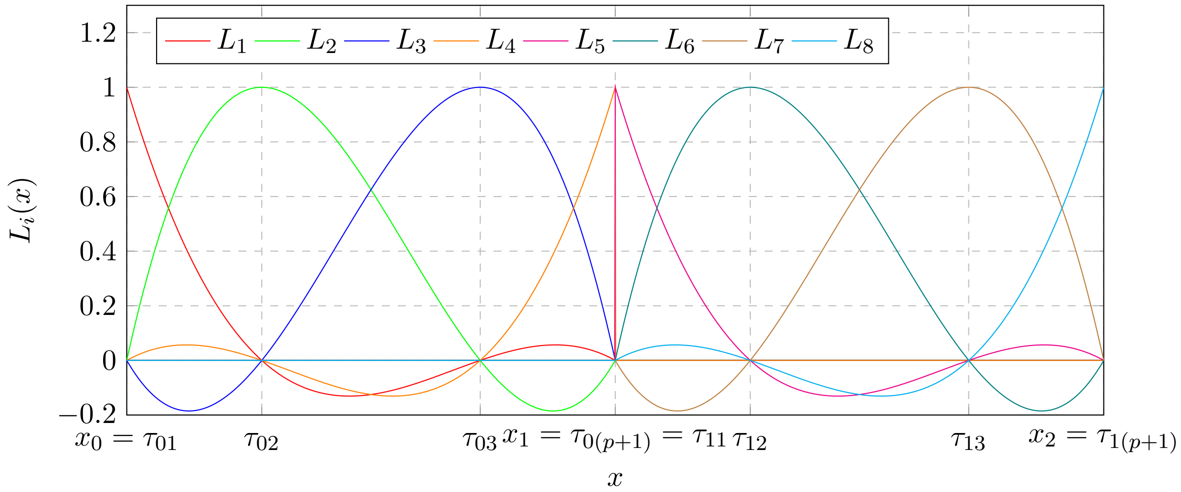


Figure 4.1: Explanation of the notation used in section 4.2 in the case $n_{\text{el}} = 2$, $p = 3$.

defined by Cox-de-Boor formula (2.1), where $n = n_{\text{el}} + p$ is the dimension of the space. Note that the basis functions take values in the parametric space $\hat{\Omega}$, and not in Ω itself. Finally, let $\mathbf{P} = \{P_i\}_{i=1}^n$ be the corresponding set of control points belonging to $\mathbb{R}^d$. Consider $\mathbf{P}$ as a matrix in $\mathbb{R}^{n\times d}$, whose i^{th} line contains the coordinates of the i^{th} control point P_i, and consider $\mathbf{N}$ as a function vector. If we write $S : \hat{\Omega} \to \Omega$ the parametrization of Ω, then from (2.2), for all $\xi \in \hat{\Omega}$,

$$S(\xi) = \sum_{i=1}^{n} N_i(\xi) P_i = \mathbf{P}^T \mathbf{N}(\xi).$$

Then, on each element E_i, $i = 0, \dots, n_{el} - 1$, each N_j is a polynomial, $j = 1, \dots, n$. And since $\{L_{i(p+1)+k}\}_{k=1}^{p+1}$ forms a basis of the polynomials on E_i, then

$$N_j|_{E_i} = \sum_{k=1}^{p+1} \beta_{jk}^{(i)} L_{i(p+1)+k} \Rightarrow N_j = \sum_{i=0}^{n_{\text{el}}-1} \sum_{k=1}^{p+1} \beta_{jk}^{(i)} L_{i(p+1)+k}, \tag{4.2}$$

where $\beta_{jk}^{(i)} \in \mathbb{R}$ for all $i = 0, \dots, n_{\text{el}} - 1$, $j = 1, \dots, n$, $k = 1, \dots, p + 1$. More precisely, since the Lagrange polynomials $\mathbf{L}$ are interpolatory, the coefficients are uniquely defined as

$$\beta_{jk}^{(i)} := N_j(\tau_{ik}).$$

Equation 4.2 can then be written in matrix form as

$$\mathbf{N} = \mathbf{D}^T \mathbf{L}, \text{ with } \mathbf{D}^T = \begin{pmatrix} \beta_{11}^{(0)} & \beta_{12}^{(0)} & \dots & \beta_{1(p+1)}^{(0)} & \beta_{11}^{(1)} & \dots & \beta_{1(p+1)}^{(n_{\text{el}}-1)} \\ \beta_{21}^{(0)} & \beta_{22}^{(0)} & \dots & \beta_{2(p+1)}^{(0)} & \beta_{21}^{(1)} & \dots & \beta_{2(p+1)}^{(n_{\text{el}}-1)} \\ \vdots & & \ddots & & \ddots & & \vdots \\ \beta_{n1}^{(0)} & \beta_{n2}^{(0)} & \dots & \beta_{n(p+1)}^{(0)} & \beta_{n1}^{(1)} & \dots & \beta_{n(p+1)}^{(n_{\text{el}}-1)} \end{pmatrix}.$$

Consequently, for all $\xi \in \hat{\Omega}$,

$$S(\xi) = \mathbf{P}^T \mathbf{N}(\xi) = \mathbf{P}^T \mathbf{D}^T \mathbf{L}(\xi) = (\mathbf{D}\mathbf{P})^T \mathbf{L}(\xi). \tag{4.3}$$

Therefore, the B-spline can be expressed with respect to the Gauss-Lobatto Legendre basis functions instead of the classical Cox-de-Boor basis functions, and the control points $\mathbf{P}$ have been mapped to **DP**. It is important to note that since Lagrange polynomials are interpolatory (this is not the case of the Cox-de-Boor basis B-splines), then the curve S passes through the mapped control points.

Globally, the B-spline curve S expressed with n degrees of freedom is now expressed with $n_{\text{el}}(p+1)$ degrees of freedom, so that not all properties of the B-splines are taken into account. In particular, their higher continuity at the elements' boundary is not considered. However, locally, the B-spline function is still expressed with only $p+1$ degrees of freedom on each element. Indeed, Cox-de-Boor formula gives the expression of the n B-spline basis functions built in such a way that each element is included in the support of only $p+1$ basis functions. Consequently, S has only $p+1$ degrees of freedom on each element. Nevertheless, note that the B-spline curve is still the same, that is the higher continuity property is conserved.

In their work, Nguyen and Schillinger have developped an isogeometric collocation method using the new expression of S given in equation (4.3). They show that this collocation method together with reduced Gauss-Lobatto quadrature gives the same accuracy as the Galerkin isogeometric method with full Gauss quadrature. The use of reduced Gauss-Lobatto quadrature is the only place where the properties of higher continuity of the B-splines are used [28]. In the following, the aim is to use the Gauss-Lobatto Lagrange extraction of B-splines made explicit in equation (4.3) and impose strongly the higher continuity condition on each elements' boundary. All of this in order to express back S with only n degrees of freedom on the whole domain.

4.2 Taking advantage of the higher continuity of B-splines

The idea is now to take the Gauss-Lobatto Lagrange basis functions and to impose the higher-continuity conditions to obtain new basis functions made of B-splines. All of this is motivated by the hope that a well-chosen subset of the Gauss-Legendre-Lobatto nodes could be a suitable set of collocation points to obtain an optimally convergent isogeometric collocation method.

Let u_{ex} be the solution of a second order general differential problem $\mathcal{L}u = f$ on $\Omega = (a, b)$, where $\mathcal{L}$ is any differential operator, together with Dirichlet boundary conditions $u(a) = d_1 \in \mathbb{R}$, $u(b) = d_2 \in \mathbb{R}$. We use the same notation and the same hypothesis as in the previous section 4.1. Let us assume that the exact and the discrete Galerkin problem on $\mathbb{B}_p^{p-1}$ admit unique solutions, respectively. Let u be the discrete solution given by the Galerkin formulation. Then on Ω,

$$u = \sum_{i=1}^{n_{\text{el}}(p+1)} u_i L_i,$$

or equivalently on each element, that is for all $i = 0, \ldots, n_{\text{el}} - 1$, for all $x \in E_i$,

$$u(x) = \sum_{j=1}^{p+1} u_{i(p+1)+j} L_{i(p+1)+j}(x). \tag{4.4}$$

Notice that without taking into account any boundary condition, this problem has $n_{\text{el}}(p+1)$ degrees of freedom, but the function described in this way is not even necessarily C^0-continuous. Let us now impose the conditions to have a C^{p-1} global continuity solution u, as an isogeometric solution in $\mathbb{B}_p^{p-1}$ has. The conditions are the following: for each interior node x_i, $i = 1, \ldots, n_{\text{el}}-1$,

$$u^{(d)}(x_i^-) = u^{(d)}(x_i^+), \qquad \text{for all } d = 0, \ldots, p-1,$$

where $u^{(d)}$ is the d^{th} derivative of u, $u^{(d)}(x_i^-) := \lim_{\substack{x \to x_i \\ <}} u^{(d)}(x)$ and $u^{(d)}(x_i^+) := \lim_{\substack{x \to x_i \\ >}} u^{(d)}(x)$.

Thanks to equation (4.4), this translates as: for all $i = 1, \dots, n_{\mathrm{el}} - 1$,

$$\sum_{j=1}^{p+1} u_{(i-1)(p+1)+j} L^{(d)}_{(i-1)(p+1)+j}(x_i^-) = \sum_{j=1}^{p+1} u_{i(p+1)+j} L^{(d)}_{i(p+1)+j}(x_i^+). \tag{4.5}$$

This formula contains $p(n_{\mathrm{el}} - 1)$ constraints, so that equation (4.4) together with equation (4.5) reduces the number of degrees of freedom of u to $n_{\mathrm{el}}(p+1) - p(n_{\mathrm{el}} - 1) = n_{\mathrm{el}} + p$.

At this point, we need to choose $n_{el}+p$ degrees of freedom among the u_i, $i = 1, \dots, n_{\mathrm{el}}(p+1)$, and re-express u using only them.

4.2.1 General case

Let I be the set of indices corresponding to the chosen degrees of freedom; its cardinality is $n_{\mathrm{el}} + p$. Let $\bar{I} = \{1, \dots, n_{\mathrm{el}}(p+1)\} \setminus I$ be the set of indices corresponding to the remaining degrees of freedom, and let $I_i := I \cap E_i$ and $\bar{I}_i := \bar{I}_i \cap E_i$, for all $i = 0, \dots, n_{\mathrm{el}} - 1$.

We note that for $p \geq 1$, B-splines need to be at least C^0-continuous at the internal nodes, that is equation (4.5) needs to be satisfied with at least $d = 0$. This directly implies $u_{(p+1)i} = u_{(p+1)i+1}$ for all $i = 1, \dots, n_{\mathrm{el}} - 1$, thanks to equation (4.1). Consequently, it is clear that keeping both degrees of freedom $u_{(p+1)i}$ and $u_{(p+1)i+1}$ does not make sense: we choose only one of them. The remaining p constraints are given by equation (4.5) for $d = 1, \dots, p$.

Now, let us put the $n_{\mathrm{el}} + p$ degrees of freedom we have chosen on one side of the equality, and the remaining ones on the other side. Then for all $i = 1, \dots, n_{\mathrm{el}} - 1$, for all $d = 0, \dots, p+1$,

$$\sum_{j \in I_{i-1}} u_j L_j^{(d)}(x_i^-) - \sum_{j \in I_i} u_j L_j^{(d)}(x_i^+) = \sum_{j \in \bar{I}_{i-1}} u_j L_j^{(d)}(x_i^+) - \sum_{j \in \bar{I}_i} u_j L_j^{(d)}(x_i^-). \tag{4.6}$$

Let $k_i := \sum_{j=0}^{i} \# I_j$ and $\bar{k}_i := \sum_{j=0}^{i} \# \bar{I}_j$, for all $i = 0, \dots, n_{\mathrm{el}} - 1$. Then equation (4.6) can be written under matricial form as

$$B u^{\mathrm{dof}} = C \bar{u}, \tag{4.7}$$

where

$$\begin{aligned}
& u^{\mathrm{dof}} \in \mathbb{R}^{n_{\mathrm{el}}+p} \text{ such that } u_i^{\mathrm{dof}} = u_{I(i)}, \forall i = 1, \dots, n_{\mathrm{el}} + p, \\
& \bar{u} \in \mathbb{R}^{p(n_{\mathrm{el}}-1)} \text{ such that } \bar{u}_i = u_{\bar{I}(i)}, \forall i = 1, \dots, p(n_{\mathrm{el}} - 1), \\
& B \in \mathbb{R}^{p(n_{\mathrm{el}}-1) \times (p+n_{\mathrm{el}})} \text{ such that } \forall d = 0, \dots, p-1, \forall i = 0, \dots, n_{\mathrm{el}} - 2, \\
& \qquad B_{ip+d+1,j} = \begin{cases} L^{(d)}_{I(j)}(x_{i+1}^-), & \text{if } j = k_{l-1}+1, \dots, k_l, \\ -L^{(d)}_{I(j)}(x_{i+1}^+), & \text{if } j = k_l + 1, \dots, k_{l+1}, \\ 0, & \text{otherwise,} \end{cases} \\
& C \in \mathbb{R}^{p(n_{\mathrm{el}}-1) \times p(n_{\mathrm{el}}-1)} \text{ such that } \forall d = 0, \dots, p-1, \forall i = 0, \dots, n_{\mathrm{el}} - 2, \\
& \qquad C_{ip+d+1,j} = \begin{cases} -L^{(d)}_{\bar{I}(j)}(x_{i+1}^-), & \text{if } j = \bar{k}_{l-1}+1, \dots, \bar{k}_l, \\ C_{ip+d+1,j} = L^{(d)}_{\bar{I}(j)}(x_{i+1}^+), & \text{if } j = \bar{k}_l + 1, \dots, \bar{k}_{l+1}, \\ 0, & \text{otherwise,} \end{cases}
\end{aligned}$$

with $I(i)$ and $\bar{I}(i)$ being respectively the i^{th} entry of I and of $\bar{I}$.

Let $\mathbf{u}$ be the vector of unknown coefficients of u in the decomposition (4.4), that we order in the following way:

$$\mathbf{u} := \begin{pmatrix} u^{\mathrm{dof}} \\ \bar{u} \end{pmatrix} \in \mathbb{R}^{n_{\mathrm{el}}(p+1)}.$$

Moreover, let $\mathbf{L}(x) \in \mathbb{R}^{n_{\text{el}}(p+1)}$ be the vector of Gauss-Legendre-Lobatto Lagrange functions, as in section 4.1, but ordered in the same way as $\mathbf{u}$. That is,

$$\begin{cases} \mathbf{L}_i = L_{I^{-1}(i)} \text{ if } i = 1, \dots, n_{\text{el}} + p, \\ \mathbf{L}_i = L_{\bar{I}^{-1}(i-n_{\text{el}}-p)} \text{ if } i = n_{\text{el}} + p + 1, \dots, n_{\text{el}}(p+1). \end{cases} \tag{4.8}$$

Consequently, thanks to this notation and thanks to equation (4.7), equation (4.4) can be written as: $\forall x \in \Omega$,

$$u(x) = \mathbf{L}^T(x)\mathbf{u} = \mathbf{L}^T(x) \begin{pmatrix} u^{\text{dof}} \\ \bar{u} \end{pmatrix} = \mathbf{L}^T(x) \begin{pmatrix} u^{\text{dof}} \\ C^{-1}Bu^{\text{dof}} \end{pmatrix} = \mathbf{L}^T(x) \begin{pmatrix} I_{nel+p} \\ C^{-1}B \end{pmatrix} u^{\text{dof}}, \tag{4.9}$$

if C is invertible. But note that C is always invertible since it is made of evaluations at different nodes of the linearly independent piecewise polynomials $L_i \in \mathbb{Q}_p(\Omega_h)$, $i \in \bar{I}$, of degree exactly equal to p, and of their $p-1$ first derivatives. Let $\mathbf{F} := \begin{pmatrix} I_{nel+p} & (C^{-1}B)^T \end{pmatrix} \mathbf{L}(x)$, then $\mathbf{F}$ is a new basis of piecewise polynomials that are globally C^{p-1}-continuous, that is a basis of B-splines of $\mathbb{B}_p^{p-1}$. Moreover, the new basis functions are linear combinations of the Gauss-Lobatto Lagrange functions.

With this new basis, a natural choice of collocation points are the Gauss-Legendre-Lobatto nodes corresponding to the chosen degrees of freedom, that is

$$\{\tau_0, \tau_{n_{\text{el}}(p+1)}\} \cup \tau, \text{ where } \tau = \{\tau_{ij} : i(p+1)+j \in I, 1 \le j \le p+1, 0 \le i \le n_{\text{el}}-1\} \setminus \{\tau_0, \tau_{n_{\text{el}}(p+1)}\},$$

since the new basis $\mathbf{F}$ has been built from those points and is interpolatory at those points. Indeed, from equation (4.1), and given the ordering of $\mathbf{L}$ in equation (4.8), then

$$\mathbf{F}(\tau_{ij}) = \begin{pmatrix} I_{nel+p} & (C^{-1}B)^T \end{pmatrix} \mathbf{L}(\tau_{ij}) = \begin{pmatrix} I_{nel+p} & (C^{-1}B)^T \end{pmatrix} \mathbf{e}_{I^{-1}(i(p+1)+j)}$$

and $I^{-1}(i(p+1)+j) \le n_{\text{el}} + p$ since $i(p+1)+j \in I$. So $\mathbf{F}(\tau_{ij}) = \mathbf{e}_{I^{-1}(i(p+1)+j)}$.

Consequently, taking into account the Dirichlet homogeneous boundary conditions and remembering our problem $\mathcal{L}u = f$ on $\Omega = (a,b)$, the linear system to solve is simply:

$$A\mathbf{u} = \mathbf{b}, \text{ with } A = \begin{pmatrix} \mathbf{F}(a)^T \\ (\mathcal{L}\mathbf{F}(\tau))^T \\ \mathbf{F}(b)^T \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} d_1 \\ f(\tau) \\ d_2 \end{pmatrix}.$$

But note that $\mathbf{F}(a)^T = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$ and $\mathbf{F}(b)^T = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix}$, since the basis is interpolatory, so $\mathbf{u}_1 = d_1$ and $\mathbf{u}_{n_{\text{el}}+p} = d_2$. Consequently, it would be enough to solve

$$A_{\text{red}}\mathbf{u}_{\text{red}} = \mathbf{b}_{\text{red}},$$

and to impose $\mathbf{u}_1 = d_1$ and $\mathbf{u}_{n_{\text{el}}+p} = d_2$, with $A_{\text{red}} \in \mathbb{R}^{(n_{\text{el}}+p-2)\times(n_{\text{el}}+p-2)}$ being $\mathcal{L}\mathbf{F}(\tau)$ without its first and last column, $\mathbf{u}_{\text{red}}$ being $\mathbf{u}$ without its first and last entries, and $\mathbf{b}_{\text{red}}$ being $\mathbf{b}$ also without its first and last entries. Both problems returns the same solution.

Furthermore, note that for all $x \in E_{i-1}$ and for all $i = 1, \dots, n_{\text{el}} - 1$, $j = 1, \dots, p+1$,

$$L_{(i-1)(p+1)+j}(x) = L_{i(p+1)+j}(x_i + x)$$

by definition of the Lagrange functions on each element. Consequently, we also have

$$L^{(d)}_{(i-1)(p+1)+j}(x) = L^{(d)}_{i(p+1)+j}(x_i + x),$$

for all $x \in E_{i-1}$ and for all $d = 0, \dots, p$. Therefore, to compute matrices B and C, we only need to compute $2(p+1)p$ values instead of $n_{\text{el}}(p+1)p$, and those values are $L_i^{(d)}(x_1^-)$ and $L^{(d)}_{(p+1)+i}(x_1^+)$, $i = 1, \dots, p+1$, $d = 0, \dots, p$.

Finally, if $\mathcal{L}$ were not a second order differential operator, or if boundary conditions were different, the procedure would be similar but one has to be careful to handle correctly boundary conditions.

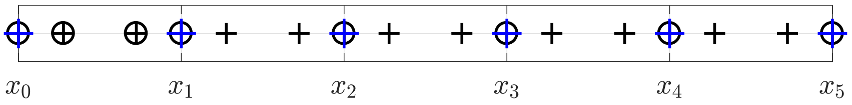


Figure 4.2: Initial choice of degrees of freedom, example with $n_{\text{el}} = 5$, $p = 3$. Blue crosses + separate elements, black crosses + correspond to the initial degrees of freedom, and bullets ○ correspond to the chosen ones. Note that 2 degrees of freedom are associated to each internal node before C^0-continuity is imposed, one corresponding to the interval for which this node is the right bound, the other one corresponding to the interval for which it is the left bound. We only keep the first one, i.e. $u_{i(p+1)}$ and not $u_{i(p+1)+1}$, for each $i = 1, \dots, n_{\text{el}} - 1$.

4.2.2 Initial choice of degrees of freedom: fill in one element

Our initial choice of degrees of freedom among the Gauss-Legendre-Lobatto nodes on each element is the following: we choose u_i for all $i \in I := \{1, 2, \dots, p+1, 2(p+1), \dots, n_{\text{el}}(p+1)\}$. We easily check that $\#I = n_{\text{el}} + p$. This is represented graphically in Figure 4.2 with $n_{\text{el}} = 5$ and $p = 3$.

As an example, we give the details of the construction of the new basis $\mathbf{F}$ as in section 4.2.1 with this choice of degrees of freedom and $p = 2$. With $p = 2$, C^1-continuity is required on the internal nodes. First, C^0-continuity (that is equation (4.5) with $d = 0$) directly implies $u_{3i} = u_{3i+1}$, for all $i = 1, \dots, n_{\text{el}} - 1$, thanks to relation (4.1). Now, there is one remaining condition on each internal node: for each $i = 1, \dots, n_{\text{el}} - 1$,

$$\sum_{j=1}^{3} u_{3(i-1)+j} L'_{3(i-1)+j}(x_i^-) = \sum_{j=1}^{3} u_{3i+j} L'_{3i+j}(x_i^+).$$

It is required to eliminate the remaining variables u_{3i+2}, $i = 1, \dots, n_{\text{el}} - 1$ that are not degrees of freedom anymore, keeping in mind that $u_{3i} = u_{3i+1}$. So:

$$u_{3i+2} = \frac{1}{L'_{3i+2}(x_i^+)} \left(\sum_{j=1}^{3} u_{3(i-1)+j} L'_{3(i-1)+j}(x_i^-) - u_{3i} L'_{3i+1}(x_i^+) - u_{3(i+1)} L'_{3(i+1)}(x_i^+) \right). \quad (4.10)$$

But this expression depends on the previous $u_{3(i-1)+2}$ which is not a degree of freedom, except if $i = 1$. Equation (4.10) is thus a recurrence relation with initial value u_2 known. Note moreover that $L_{i(p+1)+j}(x) = L_{(i+1)(p+1)+j}(x_{i+1} + x)$, for all $j = 1, \dots, p+1$ and for all $i = 0, \dots, n_{\text{el}} - 2$. Consequently, we can transform equation (4.10) into

$$u_{3i+2} = \frac{1}{L'_5(x_1^+)} \left(\sum_{j=1}^{3} u_{3(i-1)+j} L'_j(x_1^-) - u_{3i} L'_4(x_1^+) - u_{3(i+1)} L'_6(x_1^+) \right), \quad \forall i = 1, \dots, n_{\text{el}} - 1.$$

Re-writing u in function of the chosen degrees of freedom, we thus obtain

$$\begin{aligned} u = {} & u_1 \left[L_1(x) + \frac{L'_1(x_1^-)}{L'_5(x_1^+)} \chi_1 \right] + u_2 \left[L_2(x) + \frac{L'_2(x_1^-)}{L'_5(x_1^+)} \chi_2 \right] \\ & + \sum_{i=1}^{nel-2} u_{3i} \left[L_{3i}(x) + L_{3i+1}(x) - \frac{L'_6(x_1^+)}{L'_5(x_1^+)} \chi_{i-1} + \frac{L'_3\left(x_1^-\right) - L'_4\left(x_1^+\right)}{L'_5(x_1^+)} \chi_i + \frac{L'_1(x_1^-)}{L'_5(x_1^+)} \chi_{i+1} \right] \\ & + u_{3(n_{\text{el}}-1)} \left[L_{3(n_{\text{el}}-1)}(x) + L_{3(n_{\text{el}}-1)+1}(x) - \frac{L'_6(x_1^+)}{L'_5(x_1^+)} \chi_{n_{\text{el}}-2} + \frac{L'_3\left(x_1^-\right) - L'_4\left(x_1^+\right)}{L'_5(x_1^+)} \chi_{n_{\text{el}}-1} \right] \\ & + u_{3n_{\text{el}}} \left[L_{3n_{\text{el}}}(x) - \chi_{n_{\text{el}}-1} \right], \end{aligned}$$

given

$$\begin{cases} \chi_{n_{\mathrm{el}}-1} := L_{3n_{\mathrm{el}}-1}, \\ \chi_m := L_{3(m+1)-1} + \chi_{m+1} \dfrac{L'_{3(m+1)-1}(x^-_{m+1})}{L'_{3(m+2)-1}(x^+_{m+1})}, & \text{for } m = 1, \dots, n_{\mathrm{el}} - 2, \\ \chi_0 := 0. \end{cases}$$

This gives us the new basis of piecewise polynomials that are globally C^1-continuous, that is the basis of B-splines $\mathbf{F}$ we were looking for. To simplify the notation and keep the one of section 4.2.1, we write F_1 and F_2 the terms between squared brackets that multiply respectively u_1 and u_2, and we call F_{2+i}, $i = 1, \dots, n_{\mathrm{el}}$, the terms between squared brackets that multiply respectively u_{3i}. Consequently,

$$u = u_1 F_1 + u_2 F_2 + \sum_{i=1}^{n_{\mathrm{el}}} u_{3i} F_{2+i},$$

that is u is expressed with only $n_{\mathrm{el}} + p = n_{\mathrm{el}} + 2$ degrees of freedom, and the new basis functions are linear combinations of the Gauss-Lobatto Lagrange functions.

Remark that this approach gives explicitly and in a recursive way the expression of the new basis B-spline functions $\mathbf{F}$. But with p increasing, it becomes very complex to obtain such expression. The matricial way to find $\mathbf{F}$, presented in section 4.2.1, also gives an explicit expression of F and is easier to handle.

With this new basis, the natural choice of collocation points are the Gauss-Legendre Lobatto nodes corresponding to the chosen degrees of freedom, that is τ_{0j} for $j = 1, 2$ and τ_{3i} for $i = 1, \dots, n_{\mathrm{el}}$. Consequently, the linear system to solve is: $A\mathbf{u} = \mathbf{f}$, with

$$\begin{cases} \mathbf{u}_1 = u_1, \mathbf{u}_2 = u_2, \mathbf{u}_i = u_{3i} \text{ for } i = 1, \dots, n_{\mathrm{el}}, \\ \mathbf{f}_2 = f(\tau_{02}), \mathbf{f}_i = f\left(\tau_{3,(i-2)}\right) \text{ if } i = 3, \dots, n_{\mathrm{el}} + 1, \\ A_{2j} = \mathcal{L}F_j(\tau_{02}), A_{ij} = \mathcal{L}F_j\left(\tau_{3(i-2)}\right) \text{ if } i = 3, \dots, n_{\mathrm{el}} + 1, \end{cases} \tag{4.11}$$

for all $j = 1, \dots, n_{\mathrm{el}} + 2$, and $\mathbf{f}_1, \mathbf{f}_{n_{\mathrm{el}}+1}, A_{1j}$ and $A_{n_{\mathrm{el}}+2,j}$ depend on the boundary conditions, since only F_1 is non-zero on a and only $F_{n_{\mathrm{el}}+2}$ is non-zero on b.

The main problem with this approach is that the extremities of each element are collocation points, but it is exactly at those points that the continuity is limited to a $C^{p-1} = C^1$-continuity. Consequently, the differential operator L cannot be of order larger than 1. But this is not a problem only in the case $p = 2$, this issue exists with any $p \in \backslash\mathbb{N}\{0\}$: in general, the differential operator L cannot be of order larger than $p - 1$. To overcome this difficulty, different degrees of freedom should be chosen.

Example 1.a: Consider $\mathcal{L}u = u'$ and the problem $\mathcal{L}u(x) = (5\pi)\cos(5\pi x)$ in $\Omega = (-1, 1)$ with Dirichlet homogeneous boundary condition $u(-1) = 0$. The exact solution of this problem is $u_{\mathrm{ex}}(x) = \sin(5\pi x)$, for all $x \in \Omega$. We consider $p = 2$ as before, and n_{el} takes powers of 2 between 2^1 and 2^8. Let us call h the mesh size, that is $h := \frac{1}{n_{\mathrm{el}}}$. The boundary condition $u(-1) = 0$ is imposed by setting $\mathbf{f}_1 = 0$, $\mathbf{f}_{n_{\mathrm{el}}+2} = f(\tau_{3,n_{\mathrm{el}}})$, $A_{1j} = F_j(-1)$ and $A_{n_{\mathrm{el}}+2,j} = \mathcal{L}F_j(\tau_{3,n_{\mathrm{el}}})$ to complete the linear system (4.11). Otherwise, since we consider an homogeneous Dirichlet boundary condition, we can solve the following reduced problem: $A_{\mathrm{red}}\mathbf{u}_{\mathrm{red}} = \mathbf{b}_{\mathrm{red}}$, where $A_{\mathrm{red}} \in \mathbb{R}^{(n_{\mathrm{el}}+p-1)\times(n_{\mathrm{el}}+p-1)}$ is $\mathcal{L}\mathbf{F}(\tau)$ without its first column, $\mathbf{u}_{\mathrm{red}}$ is $\mathbf{u}$ without its first entry, and $\mathbf{b}_{\mathrm{red}}$ is $\mathbf{b}$ also without its first entry. And in this case, we impose $\mathbf{u}_1 = 0$. Figure 4.3 shows the convergence of the error in the L^2-norm and the evolution of the condition number of A and of A_{red} under h-refinement.

We can see that the error in the L^2-norm behaves like h^p, that is the rate of convergence is one order sub-optimal. More precisely, algebraically, it has been calculated that the error in

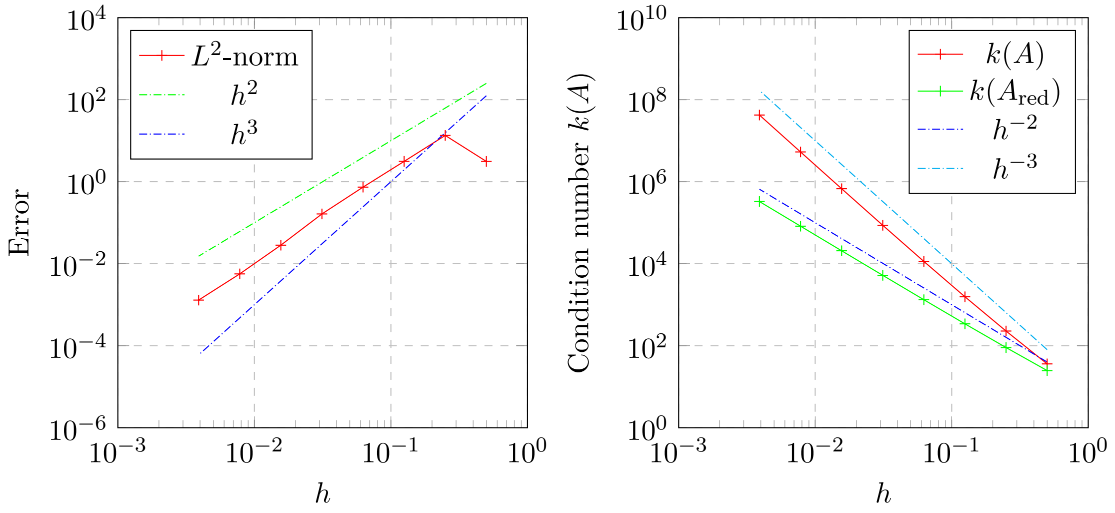


Figure 4.3: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 2$, and when a first order differential operator is used.

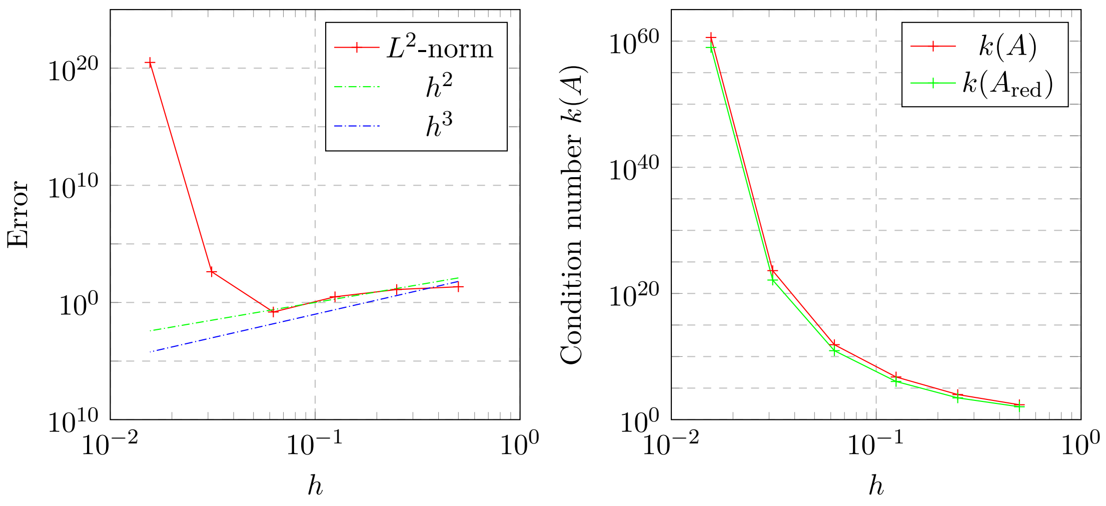


Figure 4.4: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 3$, and when a first order differential operator is used.

the L^2-norm behaves like $h^{2.12}$. This has been computed thanks to the values of the error for the smallest h we have, that is with h between $\frac{1}{32}$ and $\frac{1}{256}$. Moreover, the condition number of A behaves as h^{-3}, while the one of A_{red} behaves as h^{-2}. Consequently, it is better to solve the reduced problem with A_{red} (refer to the end of section 4.2.1), imposing strongly the boundary condition, instead of solving the full problem, especially when a large number of elements is considered.

Example 1.b: Consider the first order problem of Example 1.a, but with $p = 3$, and with h between $\frac{1}{2^1}$ and $\frac{1}{2^6}$. Figure 4.4 shows the convergence of the error in the L^2-norm and the evolution of the condition number of A and of A_{red} under h-refinement.

It is difficult to deduce the convergence rate of the error from those results since the error suddenly increases when h gets small. This comes from the fact that the condition number of both matrices A and A_{red} increase exponentially when h decreases, as it can also be seen in this

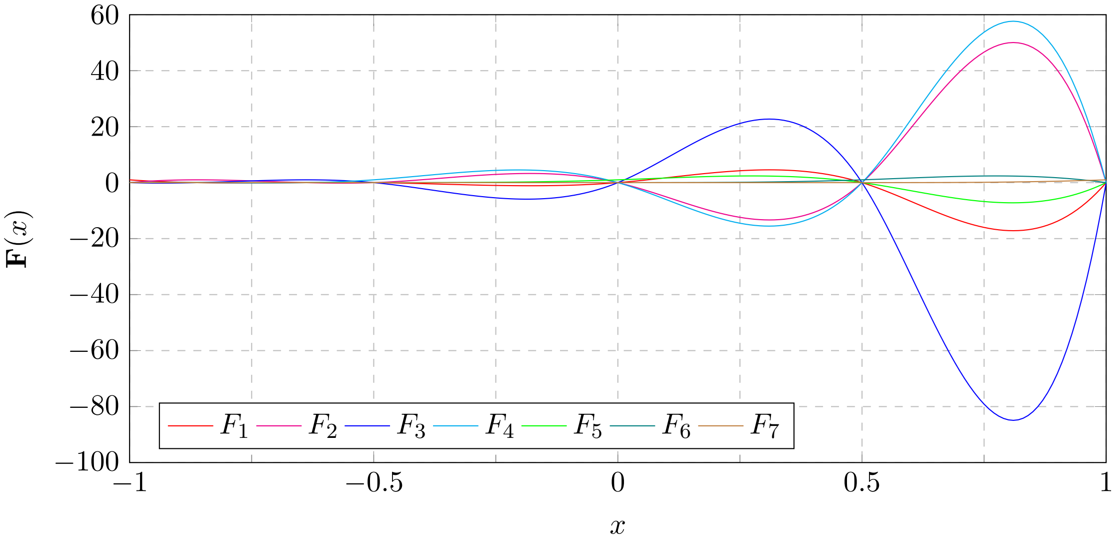


Figure 4.5: Basis functions $\mathbf{F}$ on the reference interval $(-1, 1)$ with $p = 3$ and $n_{\text{el}} = 4$.

Figure. This is due to the fact that the basis functions $\mathbf{F}$ oscillate around zero with an amplitude that is increasing with respect to the distance separating the first element to the point on which those functions are evaluated. Equivalently, this means that the Lebesgue constant of the basis functions interpolant grows with a high rate with respect to n_{el}. Therefore, the more elements, the more oscillatory is the behavior of the basis functions. This is shown in Figure 4.5.

Example 2.a: If we now consider a second order equation, for example Laplace equation $-u'' = f$, that is $\mathcal{L}u = -u''$, on $\Omega = (a, b)$ with homogeneous Dirichlet boundary conditions, then we have already seen that it is required to have $p \geq 3$ since we require at least a C^2-continuity at the boundaries of each internal element of the discretization of Ω. Since we consider homogeneous Dirichlet boundary conditions, we impose $\mathbf{f}_1 = 0$, $\mathbf{f}_{n_{\text{el}}+p} = 0$, $A_{1j} = F_j(-1)$ and $A_{n_{\text{el}}+p,j} = F_j(1)$, for all $j = 1, \dots, n_{\text{el}}+p$, to complete the linear system (4.11). Otherwise, since we consider homogeneous Dirichlet boundary conditions, we can solve the following reduced problem: $A_{\text{red}}\mathbf{u}_{\text{red}} = \mathbf{b}_{\text{red}}$, where $A_{\text{red}} \in \mathbb{R}^{(n_{\text{el}}+p-2)\times(n_{\text{el}}+p-2)}$ is $\mathcal{L}\mathbf{F}(\tau)$ without its first and last columns, $\mathbf{u}_{\text{red}}$ is $\mathbf{u}$ without its first and last entries, and $\mathbf{b}_{\text{red}}$ is $\mathbf{b}$ also without its first and last entries. And in this case, we impose $\mathbf{u}_1 = \mathbf{u}_{n_{\text{el}}+p} = 0$.

Figure 4.6 shows the convergence of the error in the L^2-norm, the H^1-norm and the H^1_0-norm under h-refinement, and the evolution of the condition number of A and of A_{red} also under h-refinement. The following cases are considered: $p = 3$, $n_{\text{el}} = 2, 2^2, \dots, 2^6$, $\Omega = (a, b) = (-1, 1)$ and $f(x) = (5\pi)^2 \sin(5\pi x)$ for all $x \in \Omega$. Let still $h = \frac{1}{n_{\text{el}}}$. Laplace equation with this forcing term and with Dirichlet homogeneous boundary conditions admits a unique exact solution $u_{\text{ex}}(x) = \sin(5\pi x)$, for all $x \in \Omega$. Again, it is difficult to deduce the convergence rate of the error from those results since the error suddenly increases when h gets small because of the exponential increase of the condition number of either A or A_{red}. Note moreover that the error is very large, and it is especially a lot larger than the error obtained when a similar first order problem is considered.

We have thus seen that another choice of degrees of freedom is needed, for many reasons: firstly because we want to use piecewise polynomials that are not necessarily of order strictly larger than the order of the differential operator taken into account, and secondly because we would like to reduce the amplitude of the oscillations of the basis functions.

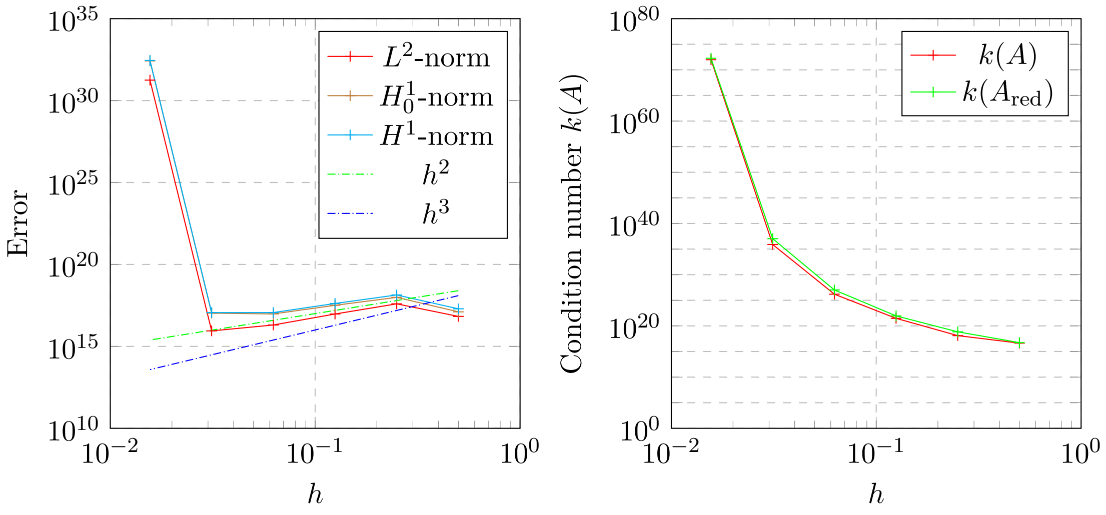


Figure 4.6: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 3$, and when a second order differential operator is used.

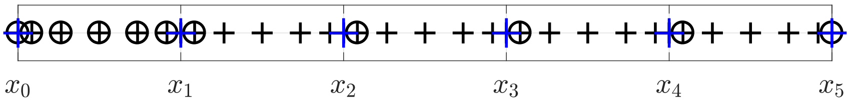


Figure 4.7: Choice of degrees of freedom such as boundary nodes are avoided, example with $n_{\text{el}} = 5$, $p = 3$. Crosses are the initial degrees of freedom, bullets correspond to the chosen ones. Note that 2 degrees of freedom are associated to each internal node, one corresponding to the interval for which this node is the right bound, the other one corresponding to the interval for which it is the left bound. None is kept.

4.2.3 Avoiding boundary nodes as degrees of freedom

Learning from the previous section, we decide to do the following choice of degrees of freedom: we choose u_i for all indices $i \in I$ where

$$I := \{1, \dots, p, p+3, 2(p+1)+2, \dots, (n_{\text{el}}-1)(p+1)+2, n_{\text{el}}(p+1)\}.$$

This is represented in Figure 4.7 with $n_{\text{el}} = 5$ and $p = 3$.

The whole procedure used to build the new B-spline basis functions $\mathbf{F}$ is exactly the same as the one previously introduced in section 4.2.1 with the right set I. We still chose as collocation points the Gauss-Legendre-Lobatto nodes corresponding to the chosen degrees of freedom, that is

$$\tau = \{\tau_{0j} : j = 1, \dots, p\} \cup \{\tau_{(p+1)i+2} : i = 1, \dots, n_{\text{el}} - 1\} \cup \{\tau_{n_{\text{el}}(p+1)}\}.$$

Example 1.c: Consider the first order problem of Examples 1.a and 1.b. Figure 4.8 shows the convergence of the error in the L^2-norm and the evolution of the condition number of A and of A_{red} under h-refinement, when $p = 2$; h takes values between $\frac{1}{2^1}$ and $\frac{1}{2^8}$. Figure 4.9 shows them when $p = 3$ instead, and h takes values between $\frac{1}{2^1}$ and $\frac{1}{2^6}$.

In both cases, we observe a convergence of the error under h-refinement that behaves like h^p in the L^2-norm as in the previous choice of degrees of freedom. However, the condition number of matrix A still increases exponentially when h is reduced in the case in which $p = 3$. Because of this high condition number, we cannot be sure neither that the behavior is asymptotically the one observed. In particular, for $h = \frac{1}{2^6}$, the error suddenly increases because the condition

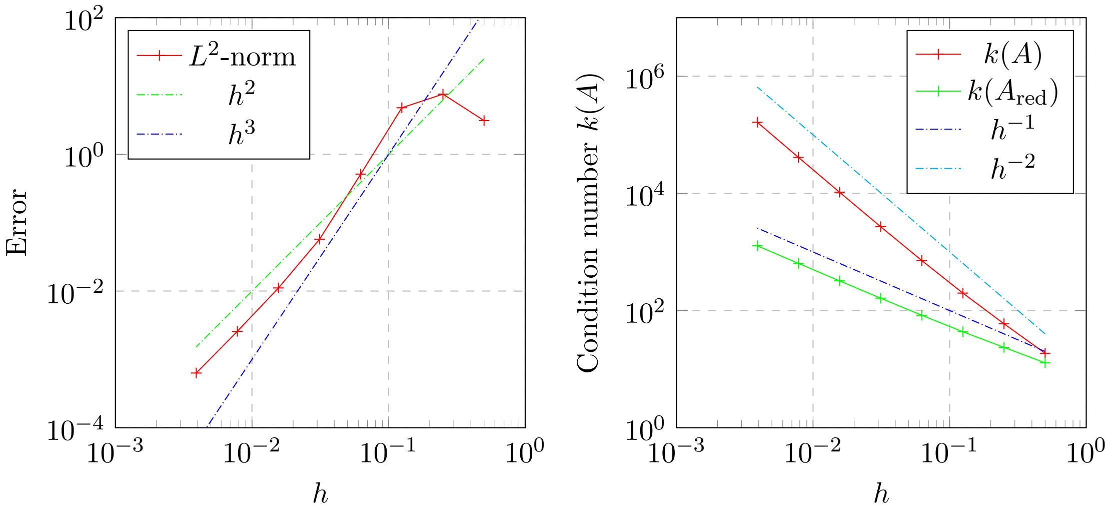


Figure 4.8: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 2$, and when a first order differential operator is used.

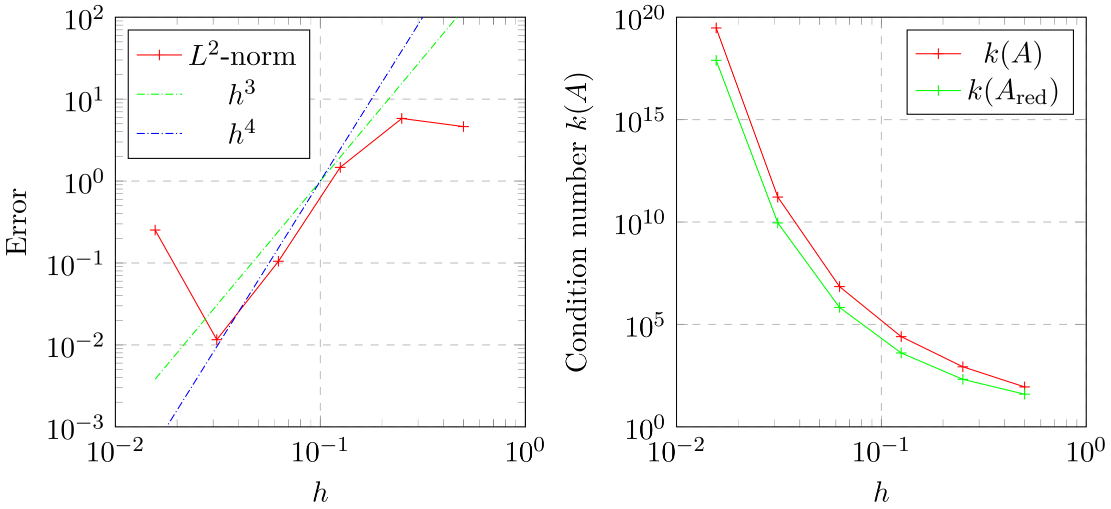


Figure 4.9: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 3$, and when a first order differential operator is used.

number is so high (around 10^{18}) that the machine epsilon is attained. Note finally that for $p = 2$, the chosen degrees of freedom are the only possible choice of degrees of freedom among the Gauss-Legendre-Lobatto nodes that excludes elements' boundaries except the boundaries of the whole domain. We did not attain the optimal convergence of the L^2-error, but almost (one order of magnitude different), and the condition number of A behaves as h^{-1}, and not like h^{-2} as it was the case with the initial choice of degrees of freedom. Algebraically, it has been found that the slope of $k(A)$ with respect to h is equal to -0.948. Furthermore, the condition number of A_{red} behaves as h^{-2} instead of h^{-3} as it was the case with the initial choice of degrees of freedom. Algebraically, it has been found that the slope of $k(A_{\text{red}})$ with respect to h is equal to -1.872. This choice of degrees of freedom is consequently better than the previous one for $p = 2$, since in the former the condition number increases with one less order of magnitude that in the latter, with respect to n_{el}. However, for $p \geq 3$, we need to choose other degrees of freedom.

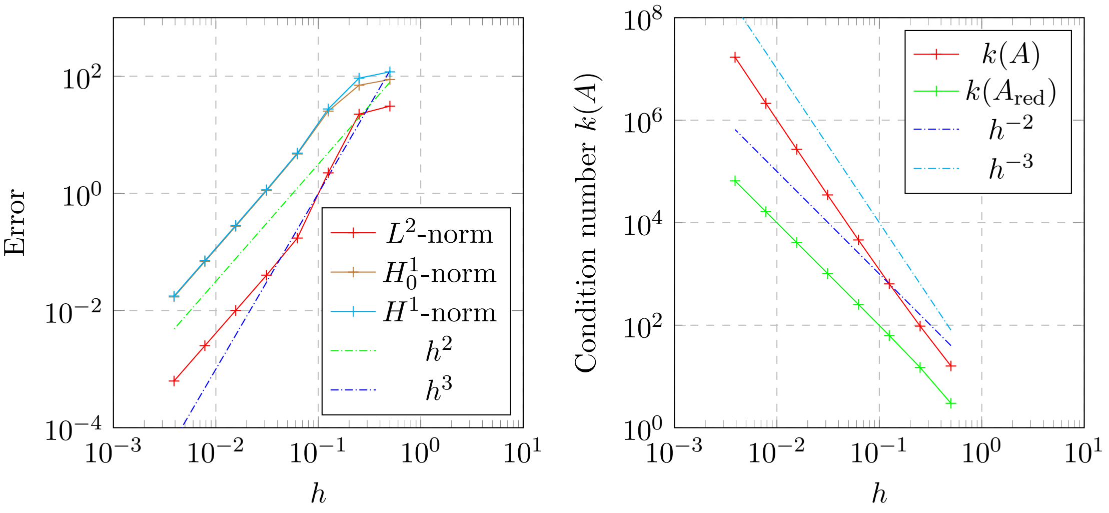


Figure 4.10: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 2$, and when a second order differential operator is used.

Example 2.b: Consider the second order problem of the previous Example 2.a, that is Laplace equation with homogeneous Dirichlet boundary conditions. Let h take powers of 2 between 2^{-1} and 2^{-8}, and let $p = 2$. The convergence of the error in both the H^1- and the L^2-norms behave like h^p. This is shown in Figure 4.10 where the convergence of the error in the L^2-norm, the H^1-norm and the H_0^1-norm under h-refinement is represented, together with the evolution of the condition number of A and of A_{red} also with respect to h. However, in some specific cases, it has been noticed that a convergence of the L^2-error as h^{p+1} can also be found (for example when the exact solution is a simple cubic polynomial).

Moreover, the condition number of matrix A behaves as h^{-3}, while the one of A_{red} behaves as h^{-2}. Consequently, again, it is better to work on the reduced problem. If we consider $p = 3$ instead of $p = 2$, the results are again difficult to interpret since the error gets suddenly very big because of the bad conditioning of matrix A: when $p = 3$, it increases exponentially when h is decreased.

Consequently, for both the first and the second order differential problems, we have found a method that is optimal in the H^1-norm and one order sub-optimal in the L^2-norm, when $p = 2$. Also in this case, the condition number of matrix A_{red} of the reduced problem grows as the number of elements if a first order problem is considered, or as the square of the number of elements if a second order problem is considered. However, for $p > 2$, the growth of the condition number of A is exponential and the convergence of the error is both harder to analyze and less accurate. This high $k(A)$ (or $k(A_{\text{red}})$, similarly) is still due to the exponential increase in amplitude of the oscillations present in the basis functions, because most of the quadrature nodes are present in the first element. Another choice of degrees of freedom should then be considered.

4.2.4 Well spread symmetric distribution of the degrees of freedom

Learning from sections 4.2.2 and 4.2.3, we need to choose degrees of freedom that are well distributed among the elements, that is, not all of them should be concentrated on one or just a few intervals, and they should be more numerous close to both boundaries of the domain. An idea is to take them symmetrically distributed with respect to the whole interval.

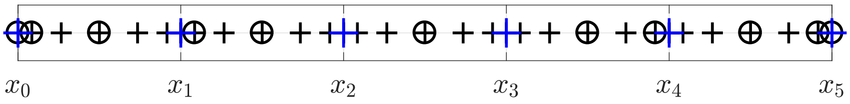


Figure 4.11: Well spread and symmetrically distributed degrees of freedom, example with $n_{\text{el}} = 5$, $p = 6$. In this case, $r = 2$ and $k = 0$. Blue crosses $+$ separate elements, black crosses $+$ correspond to the initial degrees of freedom, and bullets $\bigcirc$ correspond to the chosen ones. 2 degrees of freedom are associated to each internal node, one corresponding to the interval for which this node is the right bound, the other one corresponding to the interval for which it is the left bound. None is kept.

Assumption: p is even

We first make the assumption that p is even, so that in each element there are $p+1$ Gauss-Legendre-Lobatto nodes, that is an odd number of GLL nodes. In this way, there always exists a GLL node that is exactly in the middle of each element. We have thus chosen n_{el} nodes, being those middle GLL nodes, plus the two boundary ones. Therefore, it remains to choose $p-2$ degrees of freedom. We choose them as follows: let k and q be respectively the quotient and the remainder of the euclidean division of $p-2$ by $2n_{\text{el}}$, that is $p-2 = (2n_{\text{el}})k + q$. Since p is even, then $p-2$ is even too. Moreover, since $2n_{\text{el}}$ is even, then q should be even too. Consequently, let r be such that $2r = q$. Then the chosen degrees of freedom are the u_i with $i \in I$ and

$$\begin{aligned} I := \{1,& \\ &(i-1)(p+1)+2, \dots, (i-1)(p+1)+k+2 : i = 1, \dots, r, \\ &(i-1)(p+1)+2, \dots, (i-1)(p+1)+k+1 : i = r+1, \dots, n_{\text{el}}-r, \\ &(i-1)(p+1)+\frac{p}{2}+1 : i = 1, \dots, n_{\text{el}}, \\ &i(p+1)-k, \dots, i(p+1)-1 : i = r+1, \dots, n_{\text{el}}-r, \\ &i(p+1)-(k+1), \dots, i(p+1)-1 : i = n_{\text{el}}-r+1, \dots, n_{\text{el}}, \\ &n_{\text{el}}(p+1)\}, \end{aligned} \tag{4.12}$$

that is we add the first $k+1$ internal GLL nodes of the first r elements, the last $k+1$ internal GLL nodes of the last r elements, and the first and last k internal GLL nodes of all the remaining elements. This choice of degrees of freedom is represented in Figure 4.11 with $n_{\text{el}} = 5$ and $p = 6$.

The whole procedure used to build the new B-spline basis functions $\mathbf{F}$ is exactly the same as the one previously introduced in section 4.2.1 with the right set I. We still choose as collocation points the Gauss-Legendre-Lobatto nodes corresponding to the chosen degrees of freedom.

Example 1.d: Consider the first order problem of the previous Examples 1.a, 1.b and 1.c. Let h take powers of 2 between 2^{-1} and 2^{-8}, and let $p = 2$ and $p = 4$. The convergence of the error in the L^2-norm behaves asymptotically like h^p. This is shown in Figures 4.12 and 4.13, where the convergence of the error in the L^2-norm under h-refinement is represented, together with the evolution of the condition number of A and of A_{red} also with respect to h. Moreover, the condition number of matrix A behaves as h^{-2} when h is reduced while the one of matrix A_{red} behaves as h^{-1} in both cases $p = 2$ and $p = 4$.

Remark that the results when $p = 2$ are the same as the results presented in Example 1.*c* also with $p = 2$. We could have expected this result since in the case $p = 2$, both choices of degrees of freedom lead to the same set I. Moreover, this choice of degrees of freedom lead to a convergent method not only when $p = 2$. Indeed, we have managed to find a choice of GLL degrees of freedom such that the condition number of both matrices A and A_{red} do not explode

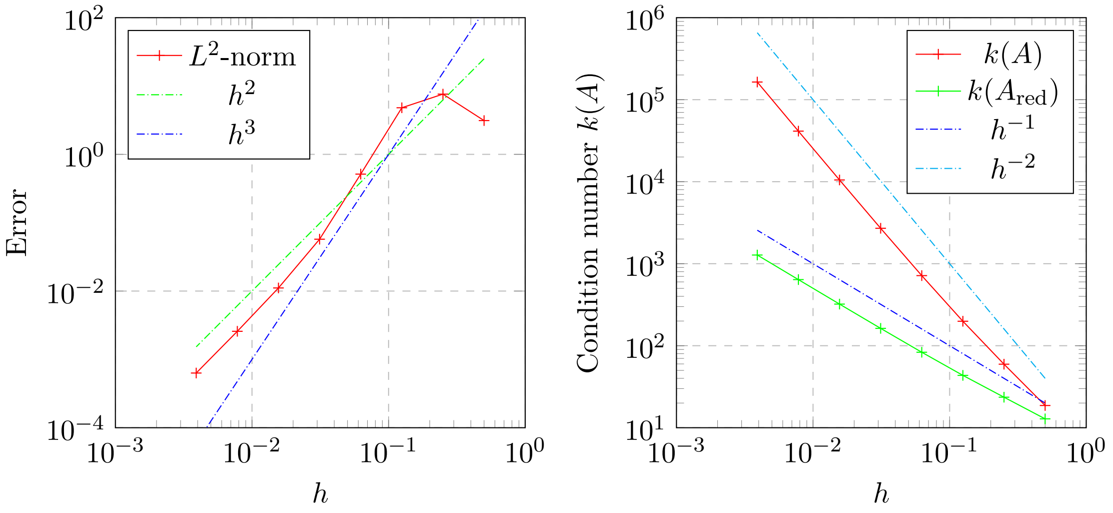


Figure 4.12: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 2$, and when a first order differential operator is used.

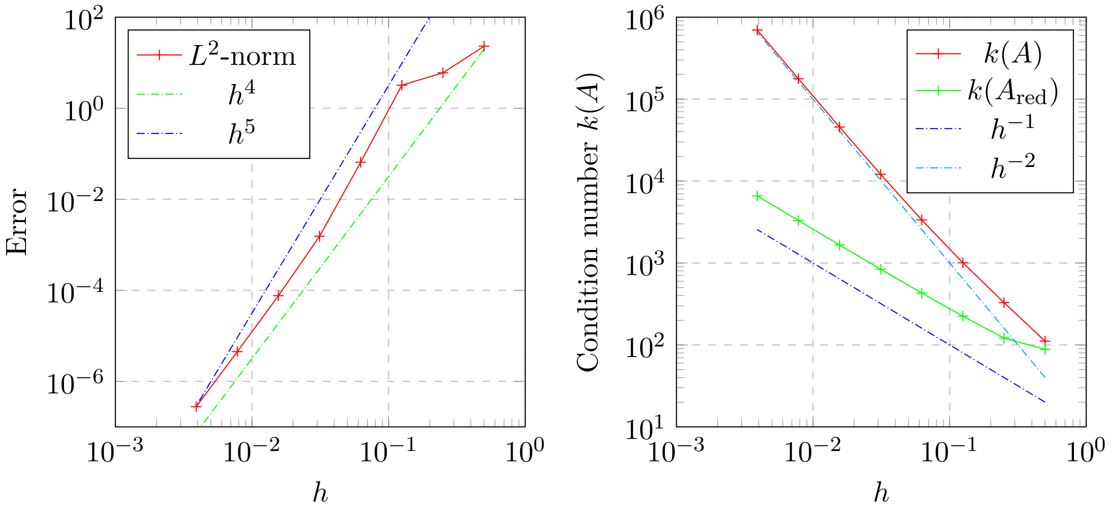


Figure 4.13: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 4$, and when a first order differential operator is used.

exponentially, and lead to a converging error, for some $p > 2$.

Example 2.c: Consider the second order problem of the previous Examples 2.a and 2.b, that is Laplace equation with homogeneous Dirichlet boundary conditions. Let h take powers of 2 between 2^{-1} and 2^{-8}, and let $p = 2$ and $p = 4$. The convergence of the error in both the H^1- and the L^2-norms behave asymptotically like h^p. This is shown in Figures 4.14 and 4.15, where the convergence of the error in the L^2-norm, the H^1-norm and the H_0^1-norm under h-refinement is represented, together with the evolution of the condition number of A and of A_{red} also with respect to h. Moreover, the condition number of A behaves as h^{-3} when $p = 2$, and slightly more slowly when $p = 4$, while the one of A_{red} behaves as h^{-2} in both cases $p = 2$ and $p = 4$.

As in the case of a first order differential problem, remark that the results when $p = 2$ are the same as the results presented in Example 2.b. It was again expectable since in the case $p = 2$, both choices of degrees of freedom lead to the same set I. Moreover, this choice of degrees of

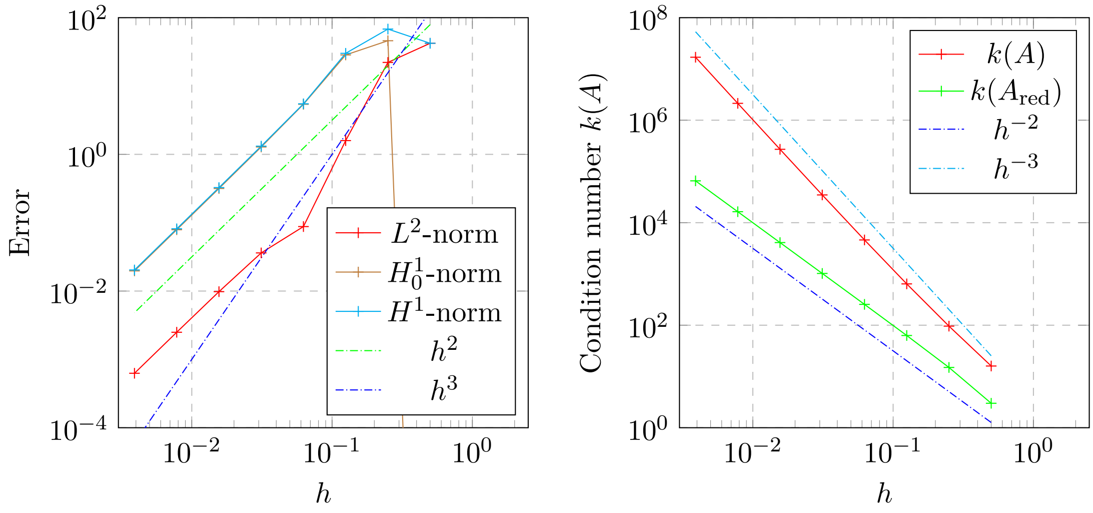


Figure 4.14: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 2$, and when a second order differential operator is used.

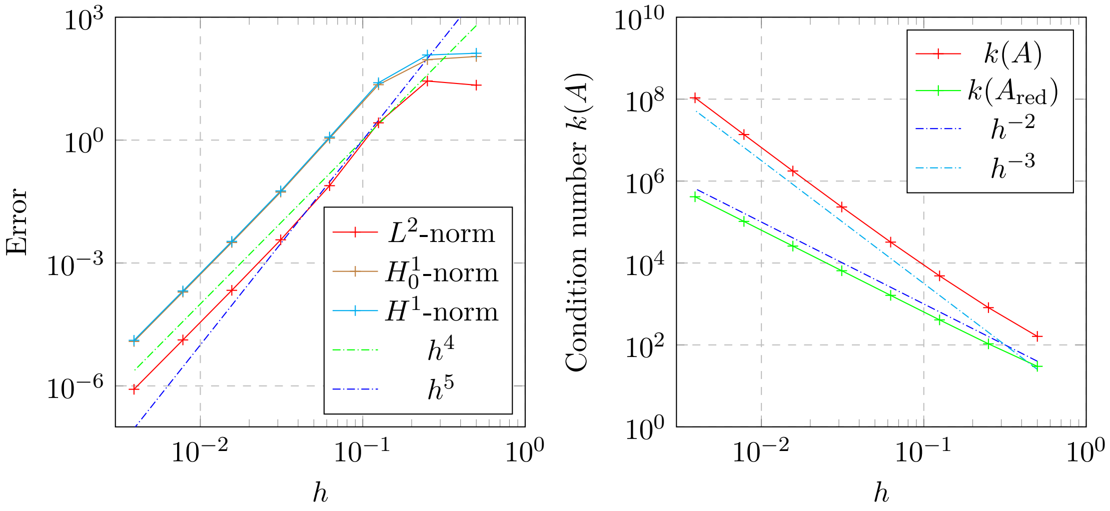


Figure 4.15: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 4$, and when a second order differential operator is used.

freedom confronted to a second order differential operator lead to a convergent method not only when $p = 2$. Indeed, we have managed to find a choice of GLL degrees of freedom such that the condition number of both matrices A and A_{red} do not explode exponentially, and lead to a converging error, for some $p > 2$.

Consequently, this choice of degrees of freedom leads to a good method in which both the L^2- and the H^1-error behave as h^p under h refinement, when p is even. However, even if this behavior is optimal in the H^1-norm, it is one order sub-optimal in the L^2-norm. Moreover, we need to find a similar collocation point distribution in the case in which p is odd.

Assumption: p is odd

When p is odd, we can not always obtain a symmetric pattern. Indeed, to choose $p + n_{\text{el}} - 2$ GLL nodes among the $n_{\text{el}}p - 1$ GLL nodes that are not on the boundary of any element, with p odd and in order to have them symmetrically distributed with respect to the middle of the domain, n_{el} needs to be odd. Consequently, we choose the degrees of freedom such that when n_{el} is odd, the nodes are indeed symmetrically distributed. First, let us see which choices can be done for $p = 3$, and then let us compare them. In a second time, we will do the same analysis for $p = 5$ and then try to generalize it for $p > 5$, p odd.

So let $p = 3$ and n_{el} be any positive integer. Since we want a symmetric pattern, we only need to determine which GLL nodes to chose on the first $\lceil \frac{n_{\text{el}}}{2} \rceil$ elements. Moreover, to have a well spread distribution of nodes in the domain, we ask to have at least one node per element. Since $p = 3$, we need to choose $n_{\text{el}} + 1$ GLL nodes among the $3n_{\text{el}} - 1$ GLL nodes that are not on the boundary of any element. Consequently, we will choose the two internal GLL nodes of only one element. To keep the symmetrical distribution of the nodes when n_{el} is odd, we need it to be either the $\lceil \frac{n_{\text{el}}}{2} \rceil$-th element, or the $\lfloor \frac{n_{\text{el}}}{2} + 1 \rfloor$-th one. Note that when n_{el} is odd, $\lceil \frac{n_{\text{el}}}{2} \rceil = \lfloor \frac{n_{\text{el}}}{2} + 1 \rfloor$, and it corresponds exactly to the middle element. Without loss of generality, since there is no preferential direction to our problem and since it would be enough to invert the parametric space, we choose the $\lceil \frac{n_{\text{el}}}{2} \rceil$-th element as the one in which we choose both internal GLL nodes. From these considerations, the four following almost exhaustive choices of degree of freedom can be made: we choose $\{u_i\}_{i \in I}$ such that:

1.

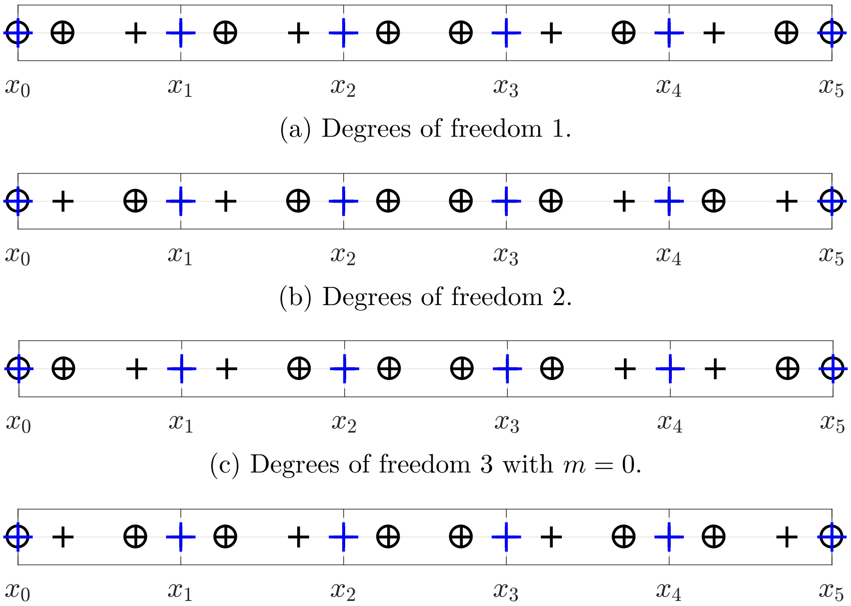
$$\begin{aligned} I = \{1,\, & i(p+1)+2 : i = 0, \dots, \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \\ & i(p+1)+3 : i = \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \dots, n_{\text{el}} - 1, \\ & n_{\text{el}}(p+1)\}, \end{aligned} \tag{4.13}$$

2. or

$$\begin{aligned} I = \{1,\, & i(p+1)+3 : i = 0, \dots, \left\lceil \frac{n_{\text{el}}}{2} \right\rceil - 1, \\ & \left\lceil \frac{n_{\text{el}}}{2} \right\rceil (p+1) + 2, \left\lceil \frac{n_{\text{el}}}{2} \right\rceil (p+1) + 3, \\ & i(p+1)+2 : i = \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \dots, n_{\text{el}} - 1, \\ & n_{\text{el}}(p+1)\}, \end{aligned}$$

3. or for any $m = 0, \dots, \lceil \frac{n_{\text{el}}}{2} \rceil - 1$,

$$\begin{aligned} I = \{1,\, & i(p+1)+2 : i = 0, \dots, m, \\ & i(p+1)+3 : i = m+1, \dots, \left\lceil \frac{n_{\text{el}}}{2} \right\rceil - 1, \\ & \left\lceil \frac{n_{\text{el}}}{2} \right\rceil (p+1) + 2, \left\lceil \frac{n_{\text{el}}}{2} \right\rceil (p+1) + 3, \\ & i(p+1)+2 : i = \left\lceil \frac{n_{\text{el}}}{2} \right\rceil + 1, \dots, n_{\text{el}} - 2 - m, \\ & i(p+1)+3 : i = n_{\text{el}} - 1 - m, \dots, n_{\text{el}} - 1, \\ & n_{\text{el}}(p+1)\}, \end{aligned}$$



(a) Degrees of freedom 1.

(b) Degrees of freedom 2.

(c) Degrees of freedom 3 with $m = 0$.

(d) Degrees of freedom 4 with $m = 0$.

Figure 4.16: Choices of well spread and symmetrically distributed degrees of freedom for $p = 3$, example with $n_{\text{el}} = 5$. Blue crosses + separate elements, black crosses + correspond to the initial degrees of freedom, and bullets ◯ correspond to the chosen ones. 2 degrees of freedom are associated to each internal node, one corresponding to the interval for which this node is the right bound, the other one corresponding to the interval for which it is the left bound. None of them is ever taken into account.

4. or for any $m = 0, \dots, \lceil \frac{n_{\text{el}}}{2} \rceil - 1$,

$$\begin{aligned} I = \{1,\, & i(p+1)+3 : i = 0, \dots, m, \\ & i(p+1)+2 : i = m+1, \dots, \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \\ & i(p+1)+3 : i = \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \dots, n_{\text{el}} - 2 - m, \\ & i(p+1)+2 : i = n_{\text{el}} - 1 - m, \dots, n_{\text{el}} - 1, \\ & n_{\text{el}}(p+1)\}. \end{aligned}$$

Those different choices of degrees of freedom are represented in Figure 4.16 with $n_{\text{el}} = 5$, and with $m = 0$ in the cases 3 and 4. The whole procedure used to build the new B-spline basis functions $\mathbf{F}$ is, in each case, exactly the same as the one previously introduced in section 4.2.1 with the right set I. We still choose as collocation points the Gauss-Legendre-Lobatto nodes corresponding to the chosen degrees of freedom.

Example 2.d: Consider the second order problem of the previous Examples 2. Let n_{el} take powers of 2 between 2^1 and 2^8 and let $p = 3$. In Table 4.1, the asymptotic behaviors of the error in both the L^2-norm and the H^1-norm, and of the condition number of matrices A and A_{red} are reported in the different cases, under h-refinement.

We observe that in many cases, no convergence is obtained and the condition number of A and A_{red} grows exponentially with the number of elements. Moreover, the optimal convergence rate of the error is never attained, neither in the L^2-norm nor in the H^1-norm. In the different cases in which convergence is observed, the L^2- and H^1-errors converge as h^2, that is as h^{p-1}.

Degrees of freedom	L^2-error	H^1-error	$k(A)$	$k\left(A_{\text{red}}\right)$
1.	$O(h^2) = O(h^{p-1})$	$O(h^2) = O(h^{p-1})$	$O(n_{\text{el}}^3)$	$O(n_{\text{el}}^2)$
2.	diverges	diverges	$O\big(\exp(n_{\text{el}})\big)$	$O\big(\exp(n_{\text{el}})\big)$
3. with $m = 0$	diverges	diverges	$O\big(\exp(n_{\text{el}})\big)$	$O\big(\exp(n_{\text{el}})\big)$
3. with $m = 1$	diverges	diverges	$O\big(\exp(n_{\text{el}})\big)$	$O\big(\exp(n_{\text{el}})\big)$
3. with $m = 2$	diverges	diverges	$O\big(\exp(n_{\text{el}})\big)$	$O\big(\exp(n_{\text{el}})\big)$
3. with $m = \lceil \frac{n_{\text{el}}}{4} \rceil$	diverges	diverges	$O\big(\exp(n_{\text{el}})\big)$	$O\big(\exp(n_{\text{el}})\big)$
3. with $m = \lceil \frac{n_{\text{el}}}{2} \rceil - 1$	$O(h^2) = O(h^{p-1})$	$O(h^2) = O(h^{p-1})$	$O(n_{\text{el}}^3)$	$O(n_{\text{el}}^2)$
4. with $m = 0$	$O(h^2) = O(h^{p-1})$	$O(h^2) = O(h^{p-1})$	$O(n_{\text{el}}^3)$	$O(n_{\text{el}}^2)$
4. with $m = 1$	$O(h^2) = O(h^{p-1})$	$O(h^2) = O(h^{p-1})$	$O(n_{\text{el}}^3)$	$O(n_{\text{el}}^2)$
4. with $m = 2$	$O(h^2) = O(h^{p-1})$	$O(h^2) = O(h^{p-1})$	$O(n_{\text{el}}^3)$	$O(n_{\text{el}}^2)$
4. with $m = \lceil \frac{n_{\text{el}}}{4} \rceil$	diverges	diverges	$O\big(\exp(n_{\text{el}})\big)$	$O\big(\exp(n_{\text{el}})\big)$
4. with $m = \lceil \frac{n_{\text{el}}}{2} \rceil - 1$	diverges	diverges	$O\big(\exp(n_{\text{el}})\big)$	$O\big(\exp(n_{\text{el}})\big)$

Table 4.1: Comparison of the asymptotic behavior of the error in the L^2- and H^1-norms, and of the condition numbers of A and of A_{red}, when $p = 3$ and when different well spread and symmetrically distributed collocation points are chosen.

This is one order sub-optimal in the H^1-norm, and two orders sub-optimal in the L^2-norm. In those same cases in which convergence is obtained, the condition number of A grows as n_{el}^3 while the condition number of A_{red} grows as in n_{el}^2, as it was already the case when $p = 2$. Again, it is thus better to solve the reduced problem.

Note that in the different cases analyzed, it was expectable to get similar results when different degrees of freedom were chosen. Indeed, for example, choice 1 and choice 4 with $m = 0$ only differ by the choice of degrees of freedom on the two boundary elements. Consequently, the behavior of the error and of the condition numbers are asymptotically the same. This is also the case between choice 2 and choice 3 with $m = 0$ for example. Furthermore, choice 1 and choice 3 with $m = \lceil \frac{n_{\text{el}}}{2} \rceil - 1$ only differ by the choice of degrees of freedom on two middle elements. Again, the behavior of the error and of the condition numbers are thus asymptotically the same, and this is also the case between choice 2 and choice 4 with $m = \lceil \frac{n_{\text{el}}}{2} \rceil - 1$ for example. Moreover, the difference between $k(A)$ or $k\left(A_{\text{red}}\right)$ corresponding to the two collocation methods obtained with such similar choices is always smaller than an order of magnitude. Consequently, the only interesting cases between the ones of Table 4.1 are the ones in which the degrees of freedom correspond to choices 1, 2, 3 with $m = \lceil \frac{n_{\text{el}}}{4} \rceil$, and 4 with $m = \lceil \frac{n_{\text{el}}}{4} \rceil$ are used. Among those four possibilities, only choice 1 leads to a convergent method.

The convergence of the error in the H^1- and L^2-norms, and the condition numbers of matrices A and A_{red} are shown in Figure 4.17 in this case, with n_{el} taking values between 2^1 and 2^{10}. Convergence as reported in Table 4.1 can be observed. Note that by taking values of n_{el} between 2^1 and 2^{10}, we only consider even numbers of elements, that is we only consider the exact cases in which symmetry cannot be observed. To make sure that this fact does not influence the convergence of the error and the conditionning of matrices A and A_{red}, Figure 4.18 reports the same information as Figure 4.17, but when n_{el} takes values between 3^1 and 3^6, that is odd values: the exact same results are observed. So with p odd, the error converges more slowly than when p is even. This fact has already been observed with different collocation methods in [1, 23, 24].

Example 1.e: Consider now the first order problem of the previous Examples 1. Let h take powers of 2 between 2^{-1} and 2^{-6}, let $p = 3$ and consider the degrees of freedom determined by choice 1. The convergence of the error in the L^2-norm, together with the condition numbers of

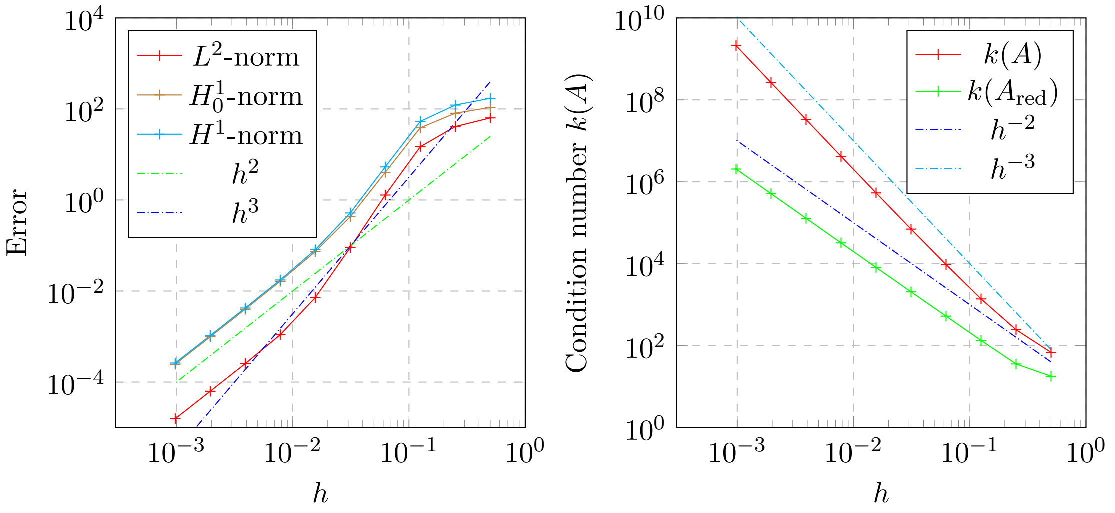


Figure 4.17: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 3$, when a second order differential operator and even numbers of elements are used. Degrees of freedom of choice 1 have been used.

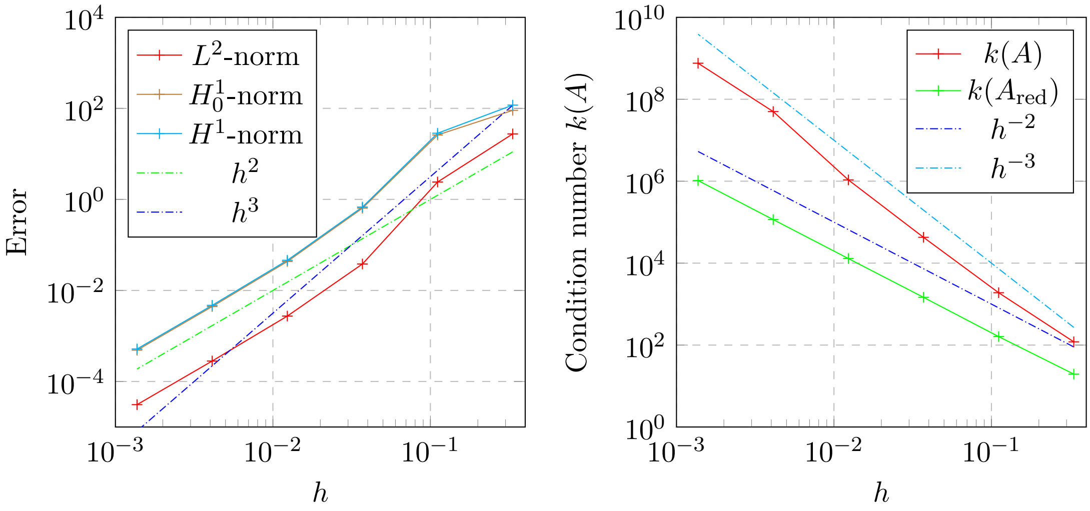


Figure 4.18: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 3$, when a second order differential operator and odd numbers of elements are used. Degrees of freedom of choice 1 have been used.

matrices A and A_{red} are reported in Figure 4.19, under h-refinement. No convergence of the error is obtained, and the condition numbers of A and of A_{red} grow exponentially when h is decreased. The other choices of degrees of freedom have also been tried on this problem, but convergence have never been obtained, and the condition numbers of A and of A_{red} always grow exponentially when h is decreased. Consequently, for first order problems, no good collocation method has been found.

So now, let us concentrate on second order problems, and let us find a way to choose the degrees of freedom that lead to collocation methods with the best possible convergence rate, for a general value of p odd. To do so, let us first choose the right degrees of freedom for $p = 5$, by trying to extrapolate logically the best choices found for $p = 2, 3, 4$.

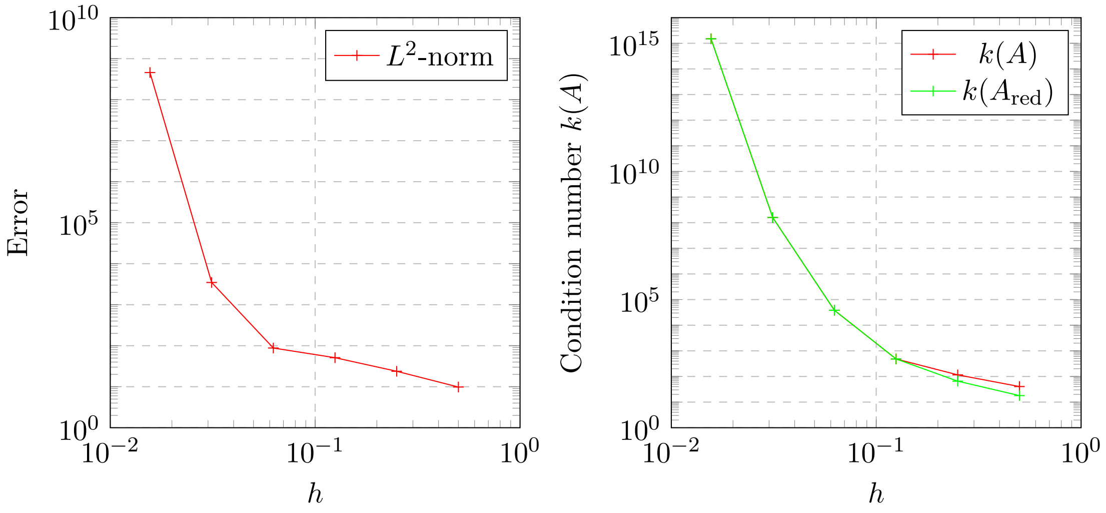


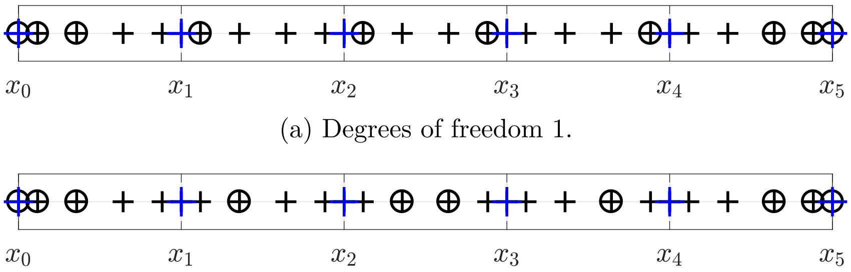
Figure 4.19: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 3$, and when a first order differential operator is used.

We have seen that for $p = 3$, the only element that contains two collocation points is the $\lceil \frac{n_{\text{el}}}{2} \rceil$-th element, that is the middle element if n_{el} is odd, or one of the two middle element if n_{el} is even. Instead, when $p = 4$, two elements need to contain two collocation points, and they are the boundary elements. Consequently, for $p = 5$, if we require to have at least one collocation point on each element, and if we require the boundary nodes of the domain to also be collocation points, then we still have to choose 3 collocation points. To follow what has been found as best for $p = 3$ and $p = 4$, we choose the two boundary elements and the $\lceil \frac{n_{\text{el}}}{2} \rceil$-th element to contain two collocation points. In this way, we choose to analyze two possible sets of degrees of freedom: we choose $\{u_i\}_{i\in I}$ such that:

1.
$$\begin{aligned} I = \{&1, 2, 3, \\ &i(p+1) + 2 : i = 1, \dots, \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \\ &i(p+1) + p : i = \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \dots, n_{\text{el}} - 2, \\ &(n_{\text{el}} - 1)(p+1) + j : j = p-1, p, p+1\}, \end{aligned}$$

2. or
$$\begin{aligned} I = \{&1, 2, \\ &i(p+1) + 3 : i = 0, \dots, \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \\ &i(p+1) + p - 1 : i = \left\lceil \frac{n_{\text{el}}}{2} \right\rceil, \dots, n_{\text{el}} - 1, \\ &(n_{\text{el}} - 1)(p+1) + j : j = p, p+1\}. \end{aligned} \tag{4.14}$$

Those different choices of degrees of freedom are represented in Figure 4.20 with $n_{\text{el}} = 5$. Other sets of degrees of freedom have been tested, but since their distributions do not follow a logical sequence with the best nodes found for $p = 2, 3, 4$, and since it has never improved the solution of the Laplace problem of Example 2, we only present in this report the results using



(a) Degrees of freedom 1.

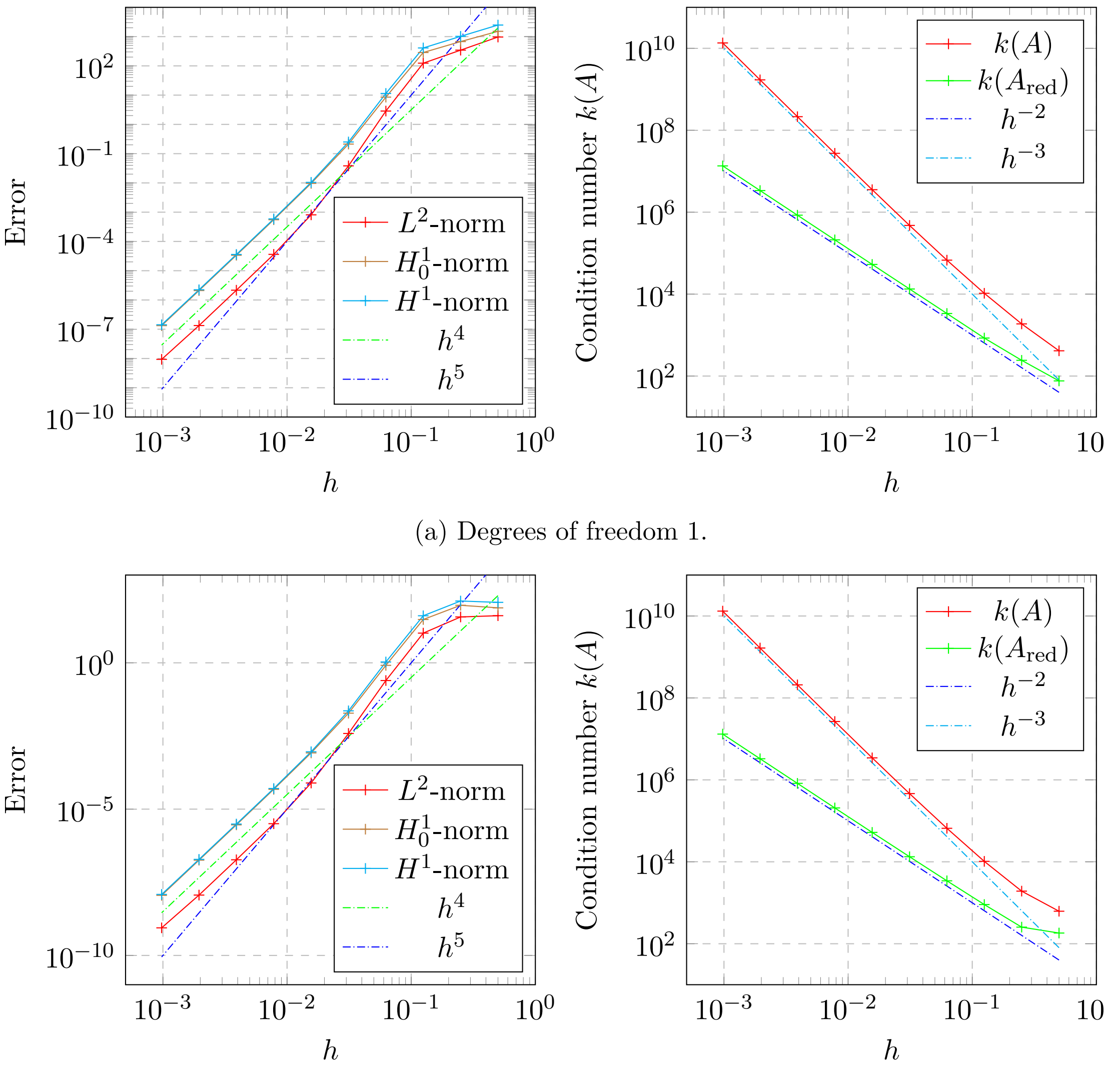
(b) Degrees of freedom 2.

Figure 4.20: Choices of well spread and symmetrically distributed degrees of freedom for $p = 5$, example with $n_{\text{el}} = 5$. Blue crosses $+$ separate elements, black crosses $+$ correspond to the initial degrees of freedom, and bullets ◯ correspond to the chosen ones. 2 degrees of freedom are associated to each internal node, one corresponding to the interval for which this node is the right bound, the other one corresponding to the interval for which it is the left bound. None of them is ever taken into account.

those two sets. Again, the procedure used to build the new B-spline basis functions $\mathbf{F}$ is in each case exactly the same as the one previously introduced in section 4.2.1 with the right set I. We still choose as collocation points the Gauss-Legendre-Lobatto nodes corresponding to the chosen degrees of freedom.

Example 2.e: Consider the second order problem of the previous Examples 2. Let h take powers of 2 between 2^{-1} and 2^{-10} and let $p = 5$. The convergence of the error in the H^1- and L^2-norms, and the condition numbers of matrices A and A_{red} are shown in Figure 4.21 under h-refinement for both choices of degrees of freedom. In both cases, the asymptotic behavior is exactly the same: both L^2- and H^1-errors behave asymptotically as $h^4 = h^{p-1}$, while $k(A)$ behaves asymptotically as h^{-3} and $k\left(A_{\text{red}}\right)$ as h^{-2}. We have thus chosen the right degrees of freedom with $p = 5$ to observe the same behavior of the condition number of A and of A_{red} as in the cases in which $p = 2, 3$ or 4. However, the convergence of the error is the same as in the case $p = 3$, that is one order sub-optimal in the H^1-norm, and two orders sub-optimal in the L^2-norm. Consequently, the method developed in this chapter seems to have a different behavior whether p is odd or even, and it behaves better when p is even.

Putting together the definitions of the best degrees of freedom for $p = 3$ and for $p = 5$, that is equations (4.13) and (4.14), we can generalize these choices to any odd integer p. Let k and q be respectively the quotient and the remainder of the euclidean division of $p - 2$ by $2n_{\text{el}}$, that is $p - 2 = (2n_{\text{el}})k + q$. Since p is odd, then $p - 2$ is odd too. Moreover, since $2n_{\text{el}}$ is even, then q should be odd too. Consequently, let r be such that $2r = q - 1$. Then in this case, the chosen



(a) Degrees of freedom 1.

(b) Degrees of freedom 2.

Figure 4.21: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 5$, and when a second order differential operator is used.

degrees of freedom are the u_i with $i \in I$ and

$$\begin{aligned} I := \{&1,\\ &(i-1)(p+1)+2,\dots,(i-1)(p+1)+k+2 : i = 1,\dots,r,\\ &(i-1)(p+1)+2,\dots,(i-1)(p+1)+k+1 : i = r+1,\dots,n_{\text{el}}-r,\\ &(i-1)(p+1)+\frac{p+1}{2} : i = 1,\dots,\left\lceil\frac{n_{\text{el}}}{2}\right\rceil,\\ &(i-1)(p+1)+\frac{p+3}{2} : i = \left\lceil\frac{n_{\text{el}}}{2}\right\rceil,\dots,n_{\text{el}}\\ &i(p+1)-k,\dots,i(p+1)-1 : i = r+1,\dots,n_{\text{el}}-r,\\ &i(p+1)-(k+1),\dots,i(p+1)-1 : i = n_{\text{el}}-r+1,\dots,n_{\text{el}},\\ &n_{\text{el}}(p+1)\}, \end{aligned} \tag{4.15}$$

that is we add the first $k+1$ GLL nodes of the first r elements, the last $k+1$ GLL nodes of the last r elements, and the first and last k GLL nodes of all the remaining elements. When we say

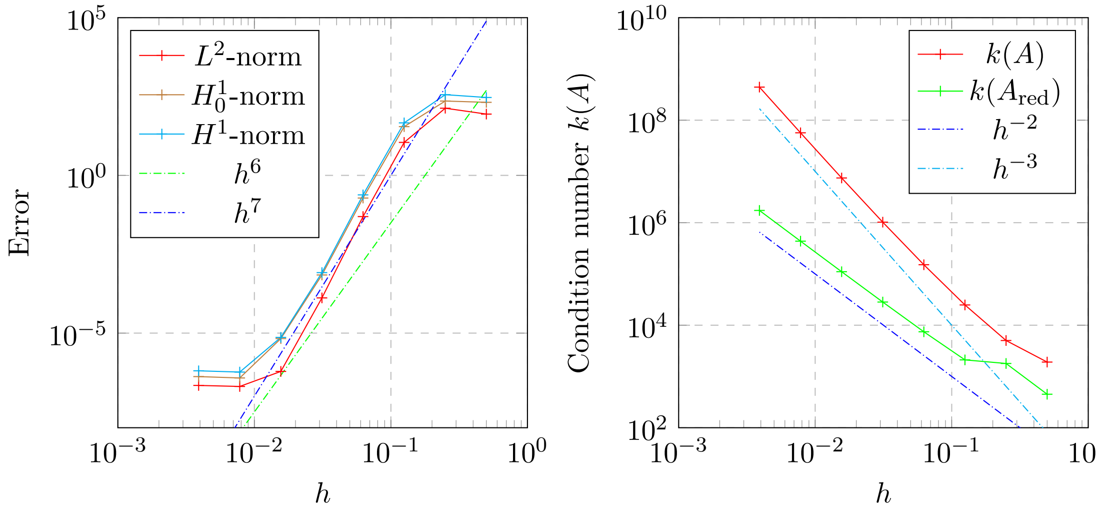


Figure 4.22: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, for $p = 7$, and when a second order differential operator is used.

"first" or "last" GLL nodes, we do not consider elements' boundaries GLL nodes. Moreover, the $\left\lceil \frac{n_{\text{el}}}{2} \right\rceil$-th element also contains an extra degree of freedom, coming from the fact that $2r = q - 1$ and not $2r = q$. It is easy to check that if $p = 3$, (4.15) is equivalent to (4.13), and if $p = 5$, (4.15) is equivalent to (4.14). We have checked this formula again for $p = 7$, in the following Example 2.f.

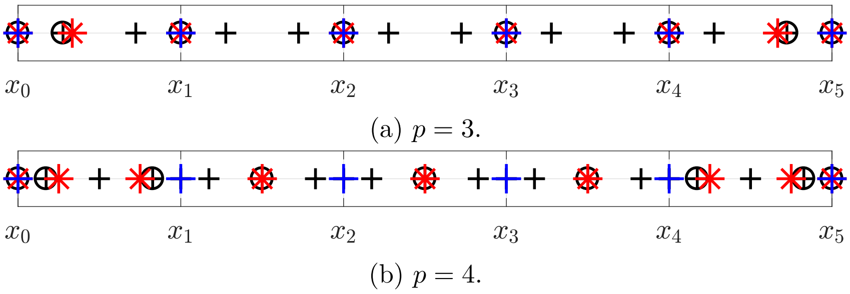
Example 2.f: Consider the second order problem of the previous Examples 2. Let h take powers of 2 between 2^{-1} and 2^{-8} and let $p = 7$. The convergence of the error in the H^1- and L^2-norms, and the condition numbers of matrices A and A_{red} are shown in Figure 4.22 under h-refinement.

We can see that the condition number of A still behaves as h^{-3} while the condition number of A_{red} still behaves as h^{-2}. The linear system solved is thus the reduced one since it gives more accurate results, especially when h is small. However and surprisingly, the error, both in the L^2-norm and in the H^1-norm, seems to behave more as $h^7 = h^p$ (or even faster) than as $h^6 = h^{p-1}$. But this result is not reliable since the asymptotic behavior of the error cannot be observed. Indeed, for $h < 2^{-6}$, the condition number of A_{red} becomes too high and the machine epsilon is attained. Consequently, the error does not decrease anymore. Indeed, in every previous example, we have observed that the error under h-refinement decreases faster when small numbers of elements are considered, and then a bit slower when the asymptotic regime is found, that is when a larger number of elements is considered. Therefore, it is justified to suppose that the asymptotic regime is not visible in Figure 4.22.

To sum up, through expressions (4.12) for p even and (4.15) for p odd, we have found a choice of degrees of freedom among the Gauss-Legendre-Lobatto nodes on each element that leads to a convergent isogeometric collocation method with the rate of convergence as high as possible. However, this IGA collocation method does not work with first order differential problems. Moreover, with this new method, we have not improved the rate of convergence of the error already found in the litterature with other IGA collocation methods [1, 23, 24]. When p is even, our method gives an error that is one order of convergence sub-optimal in the L^2-norm, with respect to the IGA Galerkin method, even if it is optimal in the H^1-norm. For p odd, the

Norm	Galerkin	CG		CGLL	
		p odd	p even	p odd	p even
L^2	$p+1$	$p-1$	p	$p-1$	p
H^1	p	$p-1$	p	$p-1$	p

Table 4.2: Comparison of IGA Galerkin method, IGA collocation method at Greville points and IGA collocation method at specific Gauss-Legendre-Lobatto points, by means of orders of convergence.



(a) $p = 3$.

(b) $p = 4$.

Figure 4.23: Choices of degrees of freedom for $p = 3$ and $p = 4$, example with $n_{\text{el}} = 5$. CBlue crosses + separate elements, black crosses + correspond to the initial degrees of freedom, bullets ◯ correspond to the chosen ones, and red stars ∗ correspond to Greville abscissae. Note that 2 degrees of freedom are associated to each internal node, one corresponding to the interval for which this node is the right bound, the other one corresponding to the interval for which it is the left bound. Only one of them is taken into account when necessary.

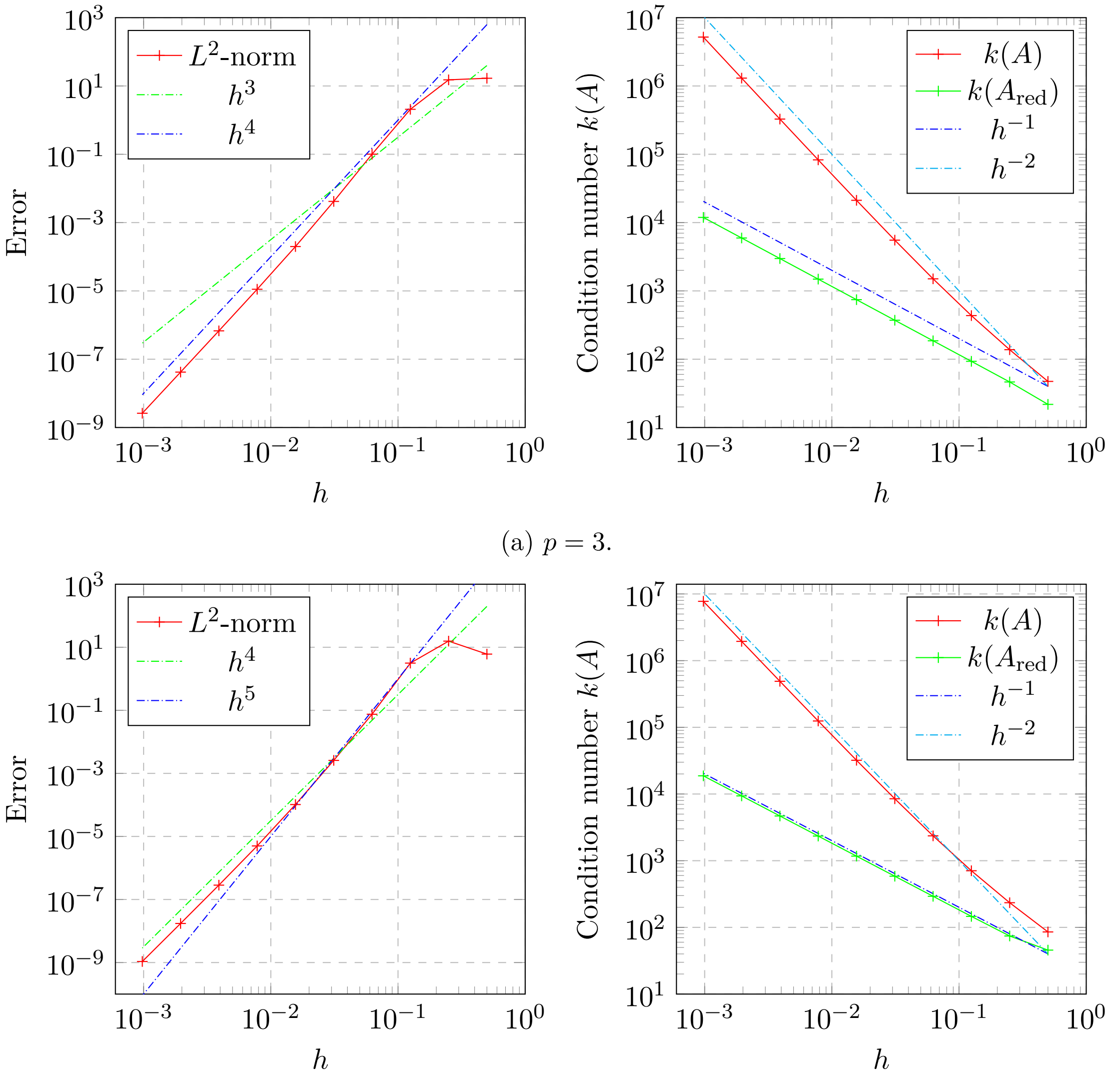
situation is worse, it is two orders of convergence sub-optimal in the L^2-norm and one order of convergence sub-optimal in the H^1-norm. Therefore, there is still some room for improvement.

4.2.5 Greville abscissae distribution among elements

It is well known that Greville abscissae lead to convergent IGA collocation methods. However, the rate of convergence of the error is not optimal for every value of B-spline order p, as it has been recalled in section 2.2.3. The following Table 4.2 sums up the different rates of convergence of the error in the L^2- and H^1-norm, when using Greville abscissae (CG, as Collocation at Greville points) or the method proposed and used in the previous section (CGLL, as Collocation at GLL points), on a second order differential problem. It is interesting to notice that CG and CGLL have the same rates of convergence in all cases.

Greville abscissae give an idea on how collocation nodes should be distributed among elements, that is how many collocation nodes there should be on each element of the space decomposition. The idea of this section is to exploit Greville abscissae in this way, in order to choose wisely the right Gauss-Legendre-Lobatto nodes on which the problem is collocated, as it has been explained in section 4.2.1. To do so, given any polynomial/B-spline order p and any number of elements n_{el}, Greville abscissae are first built. Then, for each Greville node, we look for the closest corresponding GLL node. Note that by doing so, we necessarily have the two domain boundary nodes. If two GLL nodes are exactly at the same distance of a Greville node, we choose the smallest GLL node by default, but this case never happened is our tests. In Figure 4.23, the chosen GLL points can be seen in the case $n_{\text{el}} = 5$, and $p = 3$ or $p = 4$.

Example 1.f: Consider now the first order problem of the previous Examples 1. Let h take



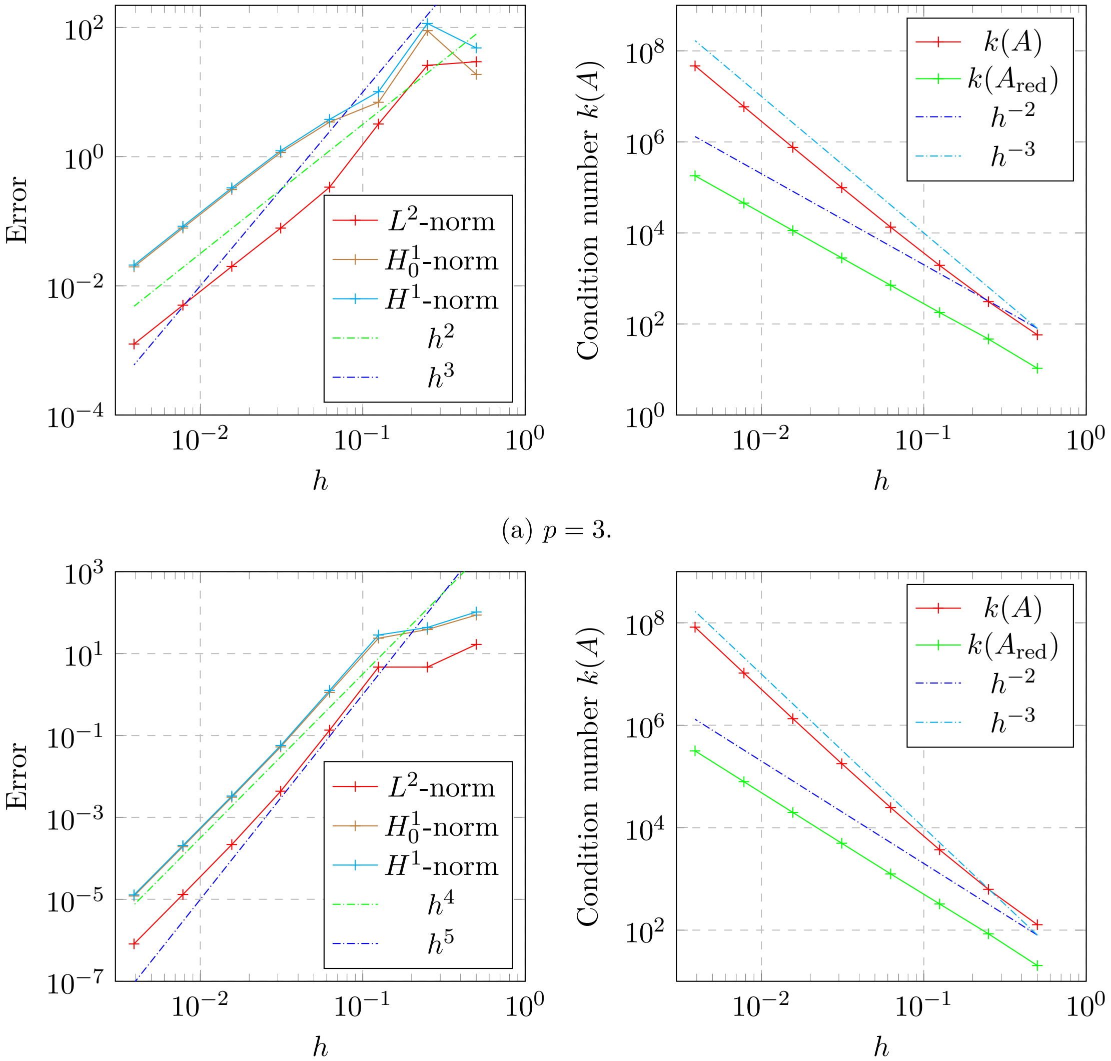
(a) $p = 3$.

(b) $p = 4$.

Figure 4.24: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, when a first order differential operator is used.

powers of 2 between 2^{-1} and 2^{-10}, and let $p = 3$ or $p = 4$. The convergence of the error in the L^2-norm and the condition numbers of matrices A and A_{red} are shown in Figure 4.24 under h-refinement.

Not only do we observe that the collocation method constructed in this way converges when a first order problem is considered, but it also converges optimally in the L^2-norm when p is odd; this means that in this case, the error converges as h^{p+1}. It has also been verified with $p > 3$, p odd. Nonetheless, observe in Figure 4.23 that when $p = 3$, the elements' boundaries are among the Greville abscissae, and thus they are taken into account as collocation points for our method since they are also GLL points. Consequently, this optimal convergence also comes with the drawback we previously wanted to avoid: at the elements' boundaries, the basis functions are limited to a C^{p-1}-continuity. So collocating the problem at those points means that the differential operator has to be of order at most $p - 1$. However, when p is even, the error still converges sub-optimally by one order of convergence: it behaves asymptotically as h^p instead of h^{p+1}. Finally, as with the previous methods, the condition number of A behaves as h^{-1} while



(a) $p = 3$.

(b) $p = 4$.

Figure 4.25: Error convergence and evolution of the condition number of A and of A_{red} under h-refinement, when a second order differential operator is used.

the one of A_{red} behaves as h^{-2}. Therefore, we solve the reduced problem instead of the full one.

Example 2.g: Now, let us consider the second order problem of the previous Examples 2. Let h take powers of 2 between 2^{-1} and 2^{-8} and let p take values 3 and 4. The convergence of the error in the H^1- and L^2-norms, and the condition numbers of matrices A and A_{red} are shown in Figure 4.25 under h-refinement.

When p is odd, the optimal convergence of the L^2-error observed in Example 1.f in the case of a first order differential operator is not present when a second order problem is considered. Instead, the error behaves as in section 4.2.4: that is for p odd, both the L^2- and the H^1-errors behave asymptotically as h^{p-1}, while for p even, both errors behave as h^p. Consequently, for second order problems, no improvement has been found with respect to the method found in section 4.2.4.

Finally, the same procedure can be done with the Demko abscissae instead of the Greville ones. Graphical results are not presented here because the same convergence results have been obtained in this case, that is, in the case in which for each Demko abscissae, we look for the closest corresponding GLL node.

4.3 Partial conclusions

To sum up, we have found in section 4.2.4 a subset of GLL nodes that lead to an IGA collocation method that has the same rate of convergence as the IGA collocation method at Greville abscissae on second order differential problems, thanks to the extraction operator introduced in section 4.1. Moreover, the subset of Gauss-Legendre-Lobatto collocation points obtained from Greville abscissae in section 4.2.5 does not lead to a better collocation method than in section 4.2.4 when a second order problem is considered. Instead, when a first order differential problem is considered, convergence is obtained with a good order of convergence in the L^2-norm: it is optimal when p is odd, and just one order sub-optimal when p is even. However, when p is odd, boundary points of the elements are considered, and this limits the type of problem that can be taken into account when using this method.

Chapter 5

Conclusions

In this project, we have explored and tested several ways to collocate first and second order differential problems in the isogeometric framework. The problem of finding a good set of collocation points in order to have an optimal convergence of the error is not easy and stays open. This exploration of isogeometric methods has been developed around two main ideas inspired from literature, whose results are summed up in this conclusion.

On one hand, we have numerically found optimal quadrature rules in order to integrate exactly the mass and the stiffness matrices arising from the isogeometric analysis when a Galerkin formulation is considered. If such quadrature formulas exist and require the evaluation of the basis functions at exactly n nodes, where n is the number of degrees of freedom of the problem, then we could have built an isogeometric collocation method from it. This is inspired by the equivalence between the Galerkin spectral element method with numerical integration using Gauss-Legendre-Lobatto quadrature formulas, and the spectral element collocation method [8]. However, we have seen numerically that any quadrature formula that integrates exactly functions of $\mathbb{B}_{2p}^{p-1}$ (needed to exactly compute the mass matrix) or $\mathbb{B}_{2(p-1)}^{p-2}$ (needed to exactly compute the stiffness matrix) require more quadrature points than degrees of freedom when $n_{\text{el}} > 1$. Indeed, the case $n_{\text{el}} = 1$ is different since in this case, IGA is equivalent to the spectral element method. This case has been verified through two different methods, leading to the same results: we have first found the minimum number of quadrature nodes required to have exact quadrature formulas, and then we have maximized the degree of exactness of a quadrature formula given a fixed number of quadrature nodes.

However, in the case of SEM G-NI, the mass matrix is also not integrated exactly by the GLL quadrature formula, but the quadrature error converges as h^p when the mesh is refined, where h is the mesh size and p is the degree of the underlying polynomials. Unfortunately, this is not the case with the quadrature formulas found in the isogeometric context. Furthermore, since we would like quadrature points to be collocation points, we have used the too large number of quadrature points needed to integrate exactly the mass and stiffness matrices in order to solve collocation problems in the least squares sense. No convergence of the error is observed in this case, this method is thus not usable.

On the other hand, we have developed a new isogeometric collocation method based on the Gauss-Lobatto Lagrange extraction of B-splines introduced by Nguyen and Schillinger in [24]. The extractor operator defined in this paper has allowed us to define a new basis of B-splines that is interpolatory at the Gauss-Legendre-Lobatto nodes and that is still strongly linked with the B-spline geometry on which the differential problem is defined, as the isogeometric paradigm of IGA requires. By then making the right choice of degrees of freedom as a subset of the Gauss-Legendre-Lobatto nodes, we have obtained a well convergent collocation method on second order differential problems. The results are summarized as follows:

- When the B-spline order p is even, the error in both the H^1- and the L^2-norms converges as h^p, i.e. the convergence order is the same as the Galerkin isogeometric method in the H^1-norm, and it is one order sub-optimal in the L^2-norm. Therefore, when p is even, this method converges with the same rate of convergence in both norms as the isogeometric collocation method when Greville or Demko abscissae are used. This convergence is obtained when either a first or a second order differential operator is used.

- When p is odd, the error in both the H^1- and the L^2-norms converges as h^{p-1}, i.e. the convergence order is one order sub-optimal in the H^1-norm and two orders sub-optimal in the L^2-norm with respect to the convergence order of the Galerkin isogeometric method. In this case, our method converges as well as the IGA collocation method given by Greville or Demko abscissae, but the method we have developed only works for second order differential problems. For first order differential problems and still p odd, optimal convergence has been obtained in the L^2-norm when the chosen GLL collocation points are the ones closest to Greville abscissae. However in this case, some collocation points are elements boundaries, and thus the order of the differential operators that can be considered are limited by the order continuity of the B-spline basis.

- When a first order differential problem is considered, the condition number of the collocation matrix $k(A)$ grows as n_{el}^2 where n_{el} is the number of elements. This happens when all basis functions are considered. But if we solve the reduced collocation problem by considering only the basis functions that fulfill the boundary conditions, the condition number of the reduced matrix, $k(A_{\mathrm{red}})$ grows more slowly, as n_{el}. In the case of a second order differential problem, $k(A)$ grows as n_{el}^3 and $k(A_{\mathrm{red}})$ as n_{el}^2. Therefore, the reduced problem should always be preferred.

Other ideas have been treated during this work, for example the idea to use Hermite quadrature formulas in Chapter 3 instead of the usual Gauss quadrature formulas, or the idea of considering the Gauss-Legendre points instead of the Gauss-Legendre-Lobatto points in Chapter 4, but none of them have led to convincing results.

Moreover, only a heuristic and numerical search of collocation points has been performed, without formal and analytic proof. Furthermore, once optimal collocation points have been found for one dimensional problems using B-splines basis functions, it would be interesting to study higher dimensional problems and NURBS-based isogeometric analysis. In a future work, tests should also be performed on other differential equations, such as the bi-harmonic equation for example, making the most of IGA as a high order numerical method.

Bibliography

[1] C. Anitescu, Y. Jia, Y.J. Zhang, and T. Rabczuk. An isogeometric collocation method using superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 284:1073–1097, 2015. http://doi.org/10.1016/j.cma.2014.11.038.

[2] F. Auricchio, L. Beirão da Veiga, Hughes T.J.R., Reali A., and Sangalli G. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, 20(11):2075–2107, 2010. http://dx.doi.org/10.1142/S0218202510004878.

[3] Y. Bazilevs, L. Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Methods and Models in Applied Sciences*, (16):1031–1090, 2006.

[4] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007. https://doi.org/10.1016/j.cma.2007.07.016.

[5] A. Buffa, C. De Falco, and G. Sangalli. IsoGeometric Analysis: Stable elements for the 2D Stokes equation. *International Journal for Numerical Methods in Fluids*, 65:1407–1422, 2011. http://dx.doi.org/10.1002/fld.2337.

[6] A. Buffa, A. Sangalli, and R. Vàzquez. Isogeometric analysis in electromagnetics: B-splines approximation. *Computer Methods in Applied Mechanics and Engineering*, 199:1143–1152, 2010. https://doi.org/10.1016/j.cma.2009.12.002.

[7] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral methods. Fundamentals in single domains.* Springer Verlag, Berlin Heidelberg New York, 2006.

[8] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral methods. Evolution to complex geometries and applications to fluid dynamics.* Springer Verlag, Berlin Heidelberg New York, 2007.

[9] O. Chanon. A comparison of the spectral element method and isogeometric analysis. Semester project, EPFL - Swiss Federal Institute of Technology of Lausanne, jan 2017.

[10] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric analysis: towards integration of CAD and FEA.* Wiley, 2009.

[11] C. De Boor. On calculating with B-splines. *Journal of approximation theory*, 6:50–62, 1972.

[12] C. De Boor. *Spline toolbox for use with MATLAB: user's guide, version 3.* MathWorks, 2005.

[13] L. Dedè, C. Jäggli, and A. Quarteroni. Isogeometric numerical dispersion analysis for two-dimensional elastic wave propagation. *Computer Methods in Applied Mechanics and Engineering*, 284:320–348, 2015.

[14] S. Demko. On the existence of interpolation projectors onto spline spaces. *Journal of Approximatio Theory*, 43:151–156, 1985.

[15] H. Gomez and L. De Lorenzis. The variational collocation method. *Computer Methods in Applied Mechanics and Engineering*, 309:152–181, 2016. http://doi.org/10.1016/j.cma.2016.06.003.

[16] E.W. Hobson. The theory of functions of a real variable and the theory of Fourier's series. 1907.

[17] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[18] T.J.R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS. *Computer Methods in Applied Mechanics and Engineering*, 197:4104–4124, 2008.

[19] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313, 2010. http://doi.org/10.1016/j.cma.2008.12.004.

[20] R.Q. Jia. Spline interpolation at knot averages. *Constructive Approximation*, 4:1–7, 1988.

[21] R.W. Johnson. A B-spline collocation method for solving the incompressible Navier-Stokes equations using an *ad hoc* method: the boundary residual method. *Computers and Fluids*, 34:121–149, 2005.

[22] The MathWorks Inc. *Optimization toolbox for use with MATLAB: user's guide, version 7.6.* MathWorks, 2017.

[23] M. Montardini, G. Sangalli, and L. Tamellini. Optimal-order isogeometric collocation at Galerkin superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 316:741–757, 2017. http://doi.org/10.1016/j.cma.2016.09.043.

[24] L.H. Nguyen and D. Schillinger. A collocated isogeometric finite element method based on Gauss-Lobatto Lagrange extraction of splines. *Computer Methods in Applied Mechanics and Engineering*, 319:720–740, 2017. http://doi.org/10.1016/j.cma.2016.09.036.

[25] L. Piegl and W. Tiller. *The NURBS book.* Springer-Verlag, 1997.

[26] A. Quarteroni. *Numerical models for differential problems*, volume 8 of *MS&A, Modeling, Simulation & Applications.* Springer Milan, 2014.

[27] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics (Texts in applied mathematics).* Springer Berlin Heidelberg, 2007.

[28] D. Schillinger, S. Hossain, and T. Hughes. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45, 2014. https://doi.org/10.1016/j.cma.2014.04.008.

[29] D. Schillinger, P. Ruthala, and L. Nguyen. Lagrange extraction and projection for NURBS basis functions: A direct link between isogeometric and standard nodal finite element formulations. *International Journal for Numerical Methods in Engineering*, 108:515–534, 2016. http://dx.doi.org/10.1002/nme.5216.