

UNIVERZA NA PRIMORSKEM  
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN  
INFORMACIJSKE TEHNOLOGIJE

Master's thesis  
(Magistrsko delo)

**Isogeometric Collocation**  
(Izogeometrična kolokacija)

Ime in priimek: *Tamara Orlich*

Študijski program: *Matematične znanosti, 2. stopnja*

Mentor: *prof. dr. Vito Vitrih*

Koper, januar 2022

## Ključna dokumentacijska informacija

Ime in PRIIMEK: Tamara ORLICH

Naslov magistrskega dela: Izogeometrična kolokacija

Kraj: Koper

Leto: 2022

Število listov: 83

Število slik: 24

Število tabel: 5

Število prilog: 9

Število strani prilog: 9

Število referenc: 21

Mentor: prof. dr. Vito Vitrih

UDK: 519.6(043.2)

Ključne besede: B-zlepki, aproksimacija, izogeometrična analiza, kolokacijska metoda

Math. Subj. Class. (2020): 65N35, 65D17, 68U07

Izvleček: V magistrskem delu analiziramo izogeometrično kolokacijsko metodo in aplikacijo te na različnih kolokacijskih točkah pri reševanju Poissonove parcialne diferencialne enačbe v eni in dveh dimenzijah. Prvo poglavje obravnava definicije in pomembne lastnosti izbranih baznih funkcij, ki so B-zlepki in NURBS-i (neenakomerni racionalni bazni B-zlepki). V drugem poglavju definiramo geometrije B-zlepkov in NURBS-ov z njihovimi lastnostmi, izogeometrični koncept ter  $L^2$  aproksimacijsko metodo. Poissonovo parcialno diferencialno enačbo v obeh dimenzijah in obeh formulacijah definiramo v tretjem poglavju, v katerem predstavimo tudi Galerkinovo izogeometrično metodo. Sledita glavni dve poglavji, v katerih opišemo izogeometrično kolokacijsko metodo in kolokacijske točke. Najprej opredelimo najbolj uporabljene Grevillove in Demkove kolokacijske točke in nato še najnovejše Superkonvergentne kolokacijske točke. Sledi obravnava Alternirajočih (*en. Alternating*) in Združenih (*en. Clustered*) superkonvergentnih kolokacijskih točk, ki smo jih izbrali iz množice vseh superkonvergentnih točk. Nalogo zaključimo z nekaterimi numeričnimi poskusi, s katerimi primerjamo red konvergence izogeometrične kolokacijske metode na Poissonovih parcialnih diferencialnih enačbah z omenjenimi kolokacijskimi točkami v različnih normiranih prostorih. Primerjavo redov konvergenz z Galerkinovo izogeometrično metodo povzamemo v ustrezni tabeli.

## Key document information

Name and SURNAME: Tamara ORLICH

Title of the thesis: Isogeometric Collocation

Place: Koper

Year: 2022

Number of pages: 83

Number of figures: 24

Number of tables: 5

Number of appendices: 9

Number of appendix pages: 9

Number of references: 21

Mentor: prof. dr. Vito Vitrih

UDC: 519.6(043.2)

Keywords: B-splines, approximation, isogeometric analysis, collocation method

Math. Subj. Class. (2020): 65N35, 65D17, 68U07

**Abstract:** In this master thesis we look at the isogeometric collocation method and how it can be used to solve Poisson's PDE in one and two dimensions. The first chapter defines and discusses the key properties of B-spline and NURBS basis functions. The following chapter discusses isogeometric analysis. In this chapter we define the geometries and properties of B-splines and NURBS, as well as the isogeometric concept and the  $L^2$  approximation method. In the following chapter, we define the Poisson's PDE in both dimensions and both formulations, namely weak and strong. This chapter also introduces the Galerkin isogeometric method. The following two chapters describe the isogeometric collocation method and collocation points: the most commonly used are the Greville and Demko collocation points, while the newest are the Superconvergent collocation points. And at last, from the Superconvergent points, we obtain the Alternating and Clustered superconvergent collocation points. Finally, we perform some numerical tests to determine the rates of convergence of isogeometric collocation methods on Poisson's PDEs with the aforementioned collocation points on different normed spaces. Furthermore, we compare convergence rates with the Galerkin isogeometric method and summarize the results in a table.

# List of Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2</b>	<b>B-SPLINES AND NURBS</b>	<b>4</b>
2.1	DEFINITIONS . . . . .	4
2.2	PROPERTIES . . . . .	7
<b>3</b>	<b>ISOGEOMETRIC ANALYSIS</b>	<b>10</b>
3.1	GEOMETRIES . . . . .	10
3.2	ISOGEOMETRIC SPACES . . . . .	14
3.3	REFINEMENTS . . . . .	15
3.4	$L^2$ APPROXIMATION . . . . .	16
<b>4</b>	<b>POISSON'S PARTIAL DIFFERENTIAL EQUATION</b>	<b>20</b>
4.1	WEAK FORM APPROXIMATION BASED ON ISOGEOMETRIC ANALYSIS . . . . .	21
4.2	THE GALERKIN ISOGEOMETRIC METHOD . . . . .	23
<b>5</b>	<b>ISOGEOMETRIC COLLOCATION METHOD</b>	<b>27</b>
5.1	SOLVING POISSON'S PDE . . . . .	27
5.2	GREVILLE COLLOCATION POINTS AND DEMKO COLLOCATION POINTS . . . . .	29
<b>6</b>	<b>SUPERCONVERGENT COLLOCATION POINTS</b>	<b>32</b>
6.1	METHOD FOR FINDING SUPERCONVERGENT POINTS . . . . .	33
6.1.1	Points for even $p$ . . . . .	33
6.1.2	Points for odd $p$ . . . . .	35
6.2	LEAST-SQUARES AT SUPERCONVERGENT COLLOCATION POINTS (LS-SP) . . . . .	38
6.3	ALTERNATING SUPERCONVERGENT COLLOCATION POINTS . . . . .	39
6.4	CLUSTERED SUPERCONVERGENT COLLOCATION POINTS . . . . .	41
<b>7</b>	<b>NUMERICAL TESTS</b>	<b>44</b>

<b>8 CONCLUSION</b>	<b>56</b>
<b>9 DALJŠI POVZETEK V SLOVENSKEM JEZIKU</b>	<b>58</b>
<b>10 REFERENCES</b>	<b>61</b>

# List of Tables

1	Triangular pattern of B-spline basis functions needed to prove the Local support property of B-spline basis functions. . . . .	7
2	Triangular pattern of B-spline basis functions needed to prove the second property of B-spline basis functions. . . . .	8
3	Error functions and their second derivative for different degrees. . . . .	37
4	Superconvergent points on interval $[-1, 1]$ for different degrees. . . . .	37
5	Comparisons of rate of convergence: GIM, GCP, DCP, LS-SP, CSCP, ASCP. . . . .	56

# List of Figures

1	B-spline basis functions of order $p = 5$ defined by the uniform and open knot vector which has 4 elements. . . . .	5
2	A cubic curve on $\Xi = \{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$ . Moving $\mathbf{B}_5$ (to $\mathbf{B}'_5$ ) changes the curve on the interval $[1/4, 1)$ . . . . .	12
3	Graphical representation of the isoparametric concept in 2D. . . . .	14
4	The exact solution $u(x) = \frac{1}{2} \sin(3\pi x)$ (in blue) and the approximated solution (in red) obtained by using GIM. . . . .	26
5	Distribution of GCP for $p = 3, \dots, 7$ , on a knot vector with 8 elements. . . . .	30
6	Distribution of DCP for $p = 3, \dots, 7$ , on a knot vector with 8 elements. . . . .	31
7	Distribution of SCP for $p = 3, \dots, 7$ , on a knot vector with 8 elements. . . . .	38
8	Distribution of internal ASCP for odd $p = 3, 5, 7$ with a given knot vector having 10 elements and $n = 13$ : the ASCP are in orange, the remaining superconvergent points are displayed with blue circles, while the black rounded orange points are the “additional points”. . . . .	40
9	Distribution of internal ASCP for odd $p = 3, 5, 7$ with a given knot vector having 9 elements and $n = 12$ : the ASCP are in orange, the remaining superconvergent points are displayed with blue circles, while the black rounded orange points are the “additional points”. . . . .	40
10	Distribution of internal CSCP for odd $p = 3, 5, 7$ with a given knot vector having 10 elements: the CSCP are in green, the remaining superconvergent points are displayed with blue circles, while the black rounded green points are the “additional points”. . . . .	43

---

11	1D relative errors for GCP. . . . .	45
12	1D relative errors for DCP. . . . .	46
13	1D relative errors for ASCP. . . . .	47
14	1D relative errors for CSCP. . . . .	48
15	Comparison of relative errors for different methods and norms when $p = 3$ . . . . .	49
16	2D relative errors for GCP. . . . .	50
17	2D relative errors for DCP. . . . .	51
18	2D relative errors for ASCP. . . . .	51
19	2D relative errors for CSCP. . . . .	52
20	The physical domain $\Omega$ and the exact solution of the Poisson's PDE. .	53
21	Distribution of GCP on $\Omega$ . . . . .	54
22	2D relative errors for GCP. . . . .	54
23	The physical domain $\Omega$ and the exact solution of the Poisson's PDE. .	55
24	Distribution of GCP on $\Omega$ . . . . .	55

# List of Appendices

APPENDIX A Galerkin isogeometric method

APPENDIX B Isogeometric collocation method in 1 dimension

APPENDIX C Isogeometric collocation method in 2 dimensions

APPENDIX D Superconvergent collocation points

APPENDIX E Least squares at superconvergent collocation points

APPENDIX F Alternating superconvergent collocation points

APPENDIX G Clustered superconvergent collocation points

APPENDIX H Relative errors in 1D

APPENDIX I Relative errors in 2D

# List of Abbreviations

<i>ASCP</i>	Alternating superconvergent points
<i>CAD</i>	Computer-aided design
<i>CSCP</i>	Clustered superconvergent points
<i>DCP</i>	Demko collocation points
<i>FEA</i>	Finite element analysis
<i>FEM</i>	Finite element method
<i>GCP</i>	Greville collocation points
<i>GIM</i>	Galerkin isogeometric method
<i>i.e.</i>	that is
<i>ICM</i>	Isogeometric Collocation Method
<i>IgA</i>	Isogeometric analysis
$\text{Int}(\Gamma)$	the interior of a set $\Gamma$
<i>LS – SP</i>	Least-squares method at superconvergent collocation points
<i>NURBS</i>	Non-uniform rational B-splines
<i>PDE</i>	Partial differential equation
<i>SCP</i>	Superconvergent collocation points

## Acknowledgments

Najprej bi se zahvalila prof. dr. Vitu Vitrihu, ki mi je potrpežljivo sledil pri obdelavi magistrskega dela in mi takoj priskočil na pomoč, ko sem jo potrebovala. Zahvaljujem se tudi komisiji, prof. dr. Štefku Miklaviču in prof. dr. Marku Orlu, ki sta si uzela nekaj časa in prisluhnila predstavitvi tega magistrskega dela.

Velik hvala gre seveda tudi FAMNIT-u in vsem profesorjem, ki so mi pokazali lepoto in univerzalnost matematike, ter njeno rabo v različnih aspektih, od najbolj praktičnega do najbolj abstraktnega, in kje vse se matematika pojavlja v svetu. Zahvaljujem se tudi mednarodni pisarni in predvsem Tini Franca, ki mi je potrpežljivo pomagala pri birokratskih postopkih pri obeh izkušnjah Erasmusa+ in predvsem ko me je prepričala, da sodelujem pri prvi izkušnji, ki je bila čudovita in je ne bom nikoli pozabila, kot mi je že vnaprej ona zagotovila.

Super se zahvaljujem moji super mami in moji super prijateljici Lindi za vso spodbudo, potrpeženje in motivacijo v vseh letih študija, ter njuni pomoči pri dosežku cilja.

# 1 INTRODUCTION

Partial differential equations (PDEs) [17, 19, 20] are the most commonly used tool in scientific fields. Because we do not yet know how to solve them analytically, computational mathematics and numerical approximation have been developed. The importance of computers has grown in recent decades, paralleled by the development of more powerful computing machines. This has aided in the development of effective computational and numerical methods.

The Finite element method (FEM) (see [14]) is currently the most widely used numerical method for solving PDEs. With this method, we must approximate the physical domain, which also requires a significant amount of computational time. Another numerical method has been developed in recent years to overcome this difficulty. This method is known as Isogeometric analysis (IgA), and it is a computational mechanics technology based on functions used to represent geometry that was first introduced by Hughes, Cottrell, and Bazilevs in [4, 5] with the goal of reducing the gap between Finite element analysis (FEA) and Computer-aided design (CAD). In FEM we approximate the geometry, as previously stated, whereas in IgA we use the functions describing the geometry directly. IgA is similar to FEM in many ways, but instead of piecewise polynomial functions, the basis functions of IgA are B-splines or NURBS (Non-Uniform Rational B-splines) (see [6, 7, 16, 18]), which are the same functions used to build CAD geometries. As a result, the basis functions used in IgA are highly dependent on the geometrical domain representation; this is known as the isogeometric concept.

The Galerkin isogeometric method (GIM) is the most commonly used method for solving a PDE. Firstly, the PDE is written in a weak formulation, which necessitates lower smoothness of functions than using the strong form directly. The integrals are then evaluated, and a linear system is constructed. Finally, the linear system is solved to obtain the coefficients in the approximate solution that correspond to the basis B-spline or NURBS functions. This technique, however, necessitates the evaluation of integrals using specific quadrature rules, and the accuracy of the solution is dependent on the quality of the numerical integration. The other option is to work directly on the strong formulation. Solving the strong form of the PDE via collocation eliminates integration but requires more regular spaces.

Isogeometric collocation methods (ICMs) have been first introduced by Auricchio, Da Veiga, Hughes, Reali and Sangalli in [2]. However, in terms of rate of convergence, ICM proposed thus far does not perform as well as GIM (see [1, 13, 15]). Collocating

the equation at Greville or Demko points is a common choice because they are classical interpolation points for arbitrary degree and regularity of splines, but both choices exhibit suboptimal convergence behavior. However, there is a fascinating alternative. Gomez and de Lorenzis in [11] have demonstrated that there must exist a set of collocation points that exactly reproduces the Galerkin solution, and thus the error of the ICM built from those points has the same rate of convergence as GIM. These are the zeros of the Galerkin residual and are known as Cauchy-Galerkin points (see [1, 11]). Unfortunately, these points are not available a priori; thereby [1] selects as a surrogate the points where, under suitable hypothesis, superconvergence of the second derivatives of the Galerkin solution occurs, motivated by the fact that for Poisson PDE the Galerkin residual is actually equivalent to the error of the approximation of the second derivatives (see [15]). These are known as Superconvergent points.

However, because there are more Superconvergent points than degrees of freedom, [1] proposes using a least-squares approximation to solve the overdetermined linear system. This concept is further developed in [11, 15] by using different specific subsets of the Superconvergent collocation points (SCP) to ensure that the number of collocation points corresponds to the number of degrees of freedom. These are known as Alternating superconvergent collocation points (ASCP) and Clustered superconvergent collocation points (CSCP).

The structure of the thesis is the following.

- In Chapter 2, we introduce the preliminary theory, which includes the necessary definitions of the B-splines and NURBS basis functions mentioned in this work, as well as stating and proving a number of useful properties about them.
- The third chapter is divided into four sections: the first section introduces B-splines and NURBS geometries, as well as their properties; the second section explains the isoparametric concept; the third section describes three different procedures for obtaining more precise isogeometric solutions; and the final section introduces the  $L^2$  approximation, as well as some other definitions of functional analysis.
- The Poisson's partial differential equation is discussed in Chapter 4 in both one and two dimensions, as well as the technique for obtaining the weak formulation from the strong one. The Galerkin isogeometric method is also presented, as well as its application to Poisson's partial differential equation with Dirichlet boundary conditions.
- The main section is in Chapter 5, where we describe the isogeometric collocation method in both observed dimensions. We also provide a selection of the most widely used collocation points, namely the Greville and Demko collocation points.

- We compute Superconvergent points and introduce the Least-squares method at them in Chapter 6. The Alternating and Clustered superconvergent collocation points are then obtained using Superconvergent points.
- In Chapter 7, we perform some numerical tests in one and two-dimensional spaces, computing the isogeometric collocation for all mentioned collocation points and various Poisson's partial differential equations. We compare the relative errors obtained in the  $L^2$ ,  $H_1$ , and  $H_2$  norms. Wolfram Mathematica is the programming language we use to run our algorithms and visualize the results with graphs.
- Finally, in Chapter 8, we make our conclusions.

## 2 B-SPLINES AND NURBS

In this chapter, we aim at introducing definitions of B-splines and NURBS basis functions and their properties. For the definitions and properties in this chapter, we refer to [5, 7, 9, 16].

### 2.1 DEFINITIONS

B-splines are piecewise polynomial curves composed of linear combinations of B-spline basis functions. The B-spline basis is defined through the concept of knot vectors.

**Definition 2.1.** Let  $\hat{\Omega}$  be the one-dimensional parametric space. A finite subset of  $\hat{\Omega}$  made of non-decreasing set of coordinates in this parametric space  $\hat{\Omega}$  is called *knot vector*. It is written as  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ , where  $\xi_i \in \mathbb{R}$  is the  $i$ -th knot,  $i$  is the knot index,  $i = 1, 2, \dots, n + p + 1$ ,  $p$  is the polynomial degree and  $n$  is the number of B-spline basis functions.

**Definition 2.2.** Each interval  $[\xi_i, \xi_{i+1}]$  for  $i = 1, 2, \dots, n + p$  is called a *knot span*. Each internal *knot span*, that is each interval  $[\xi_i, \xi_{i+1}]$  for  $i = 1, 2, \dots, n + p$ , such that  $\xi_{i+1} \neq \xi_1$  and  $\xi_i \neq \xi_{n+p+1}$ ,  $i = 1, 2, \dots, n + p$ , is called *element*.

**Definition 2.3.** A knot vector is said to be *uniform* if the knots are equally spaced in the parametric space. If they are unequally spaced, then the knot vector is *non-uniform*.

Knots may be repeated, that is, more than one knot is located at the same coordinate in the parametric space. Note that this implies that a knot span can have zero length.

**Definition 2.4.** A knot vector is said to be *open* if its first and last knots repeat  $p + 1$  times.

We now have all that we need to define B-spline and NURBS basis functions. Firstly, we will define the univariate B-spline basis functions, which are B-spline basis functions that are defined from a one-dimensional parametric space  $\hat{\Omega}$ .

**Definition 2.5.** Let  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$  be defined as in Def. 2.1. For  $i \in \{1, 2, \dots, n\}$ , the *univariate B-spline basis functions*  $N_{i,p}$ , are defined with a recursive formula called *Cox-de Boor recursion formula* as:

- For  $p = 0$  they are defined by

$$N_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.1)$$

where  $i = 1, 2, \dots, n$ .

- For  $p = 1, 2, 3, \dots$  they are defined by

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi), \quad (2.2)$$

where  $i = 1, 2, \dots, n$ .

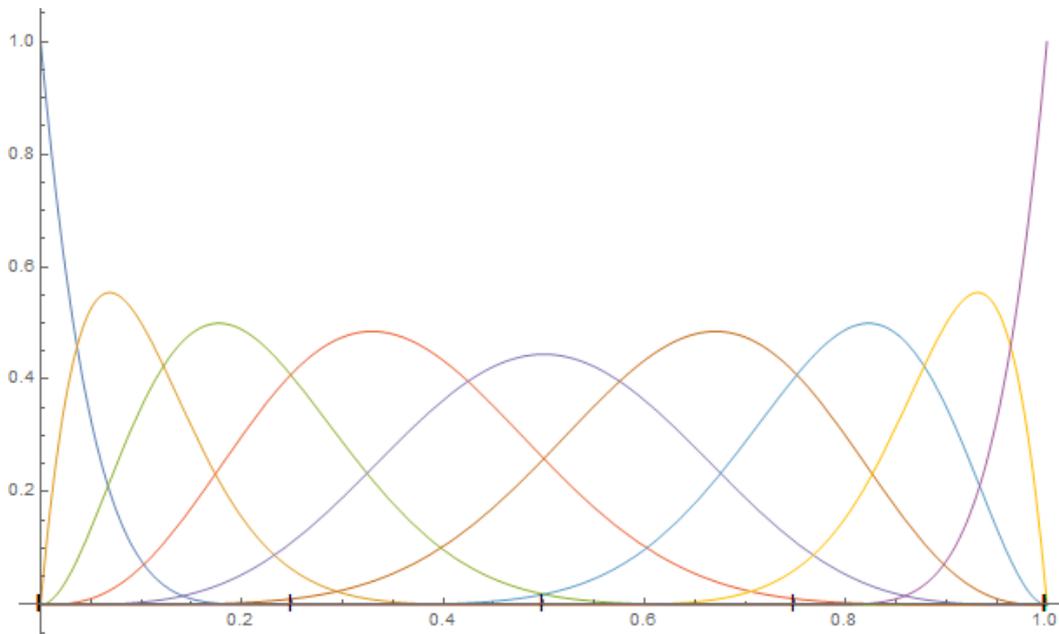


Figure 1: B-spline basis functions of order  $p = 5$  defined by the uniform and open knot vector which has 4 elements.

*Remark 2.6.* By convention, we consider that  $N_{n+1,p} = 0$  for all  $p > 0$  and that  $\frac{0}{0} = 0$  (which happens whenever  $\xi_{i+p} = \xi_i$  or if  $\xi_{i+p+1} = \xi_{i+1}$ ). Also, note that,  $N_{i,0}(\xi)$  is a step function, equal to zero everywhere except on the half open interval  $\xi \in [\xi_i, \xi_{i+1})$ , while for  $p > 0$ ,  $N_{i,p}(\xi)$  is a linear combination of two  $(p - 1)$ -degree basis functions. In the following,  $h$  will represent the size of the elements.

**Lemma 2.7.** *Let  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$  be an open knot vector, as in Def. 2.1. Let  $n_{el}$  be the number of elements of  $\Xi$ . Then  $n = n_{el} + p$ .*

*Proof.* Since  $\Xi$  is open, by definition, we know that we have  $p + 1$  knots equal to  $\xi_1$  and  $p + 1$  knots equal to  $\xi_{n+p+1}$ . Consequently, the number of internal knots is on one hand equal to  $(n + p + 1) - 2(p + 1)$ , on the other hand it is equal to  $n_{el} - 1$ . So,  $n - p - 1 = n_{el} - 1$  and therefore  $n = n_{el} + p$ .  $\square$

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  $k$ -variate B-spline basis,  $k \in \mathbb{N} \setminus \{0\}$ , is defined from a parametric space  $\hat{\Omega}$  of dimension  $k$ , that can be decomposed thanks to a cartesian product as  $\prod_{i=1}^k \hat{\Omega}_i$ , where each  $\hat{\Omega}_i$  is a one-dimensional parametric space. In this work, we will consider just the 2-variate B-spline basis and we give the following formal definition:

**Definition 2.8.** Let  $\Xi_i \subset \hat{\Omega}_i$ ,  $i = 1, 2$ , be 2 knot vectors that define  $n_i$  univariate B-spline basis functions with polynomial degree equal to  $p_i$ ,  $i = 1, 2$ , respectively. Then the corresponding 2-variate B-spline basis is

$$\{N_{\mathbf{j}, \mathbf{p}} : \mathbf{j} = (j_1, j_2), 0 \leq j_1 \leq n_1, 0 \leq j_2 \leq n_2; \mathbf{p} = (p_1, p_2)\}$$

such that for all  $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \hat{\Omega}$ ,  $N_{\mathbf{j}, \mathbf{p}}(\boldsymbol{\xi}) = \prod_{i=1}^2 N_{j_i, p_i}(\xi_i)$ . The set of 2-dimensional elements used to define B-spline basis functions is called *mesh*.

*Remark 2.9.* The basis functions formed from open knot vectors are interpolatory at the ends of the parametric space interval in one dimension and at the corners of patches in multiple dimensions but they are not, in general, interpolatory at interior knots.

Finally, non-uniform rational B-splines (NURBS) are built from B-splines and the NURBS basis functions are an extension of the B-spline basis functions, that is, they define a larger function space. The motivation of the introduction of NURBS is that in general, conic sections cannot be parameterized by polynomials, but can be parameterized with rational functions.

**Definition 2.10.** Let  $\hat{\Omega}$  be a one-dimensional parametric space, and let  $\Xi$  be a knot vector on  $\hat{\Omega}$  that generates the  $n$  univariate B-spline basis functions of order  $p$ , as in Def. 2.5. Then given  $n$  weights  $w_i \in \mathbb{R}$ ,  $i = 1, 2, \dots, n$ , we can define the set of *univariate NURBS basis functions* as

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

for all  $i \in \{1, 2, \dots, n\}$  and for all  $\xi \in \hat{\Omega}$ .

The *order* of NURBS basis functions is the order of the underlying B-spline basis functions. In general, we define NURBS basis functions using positive weights.

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, so we will omit it, but we refer the readers to see [16].

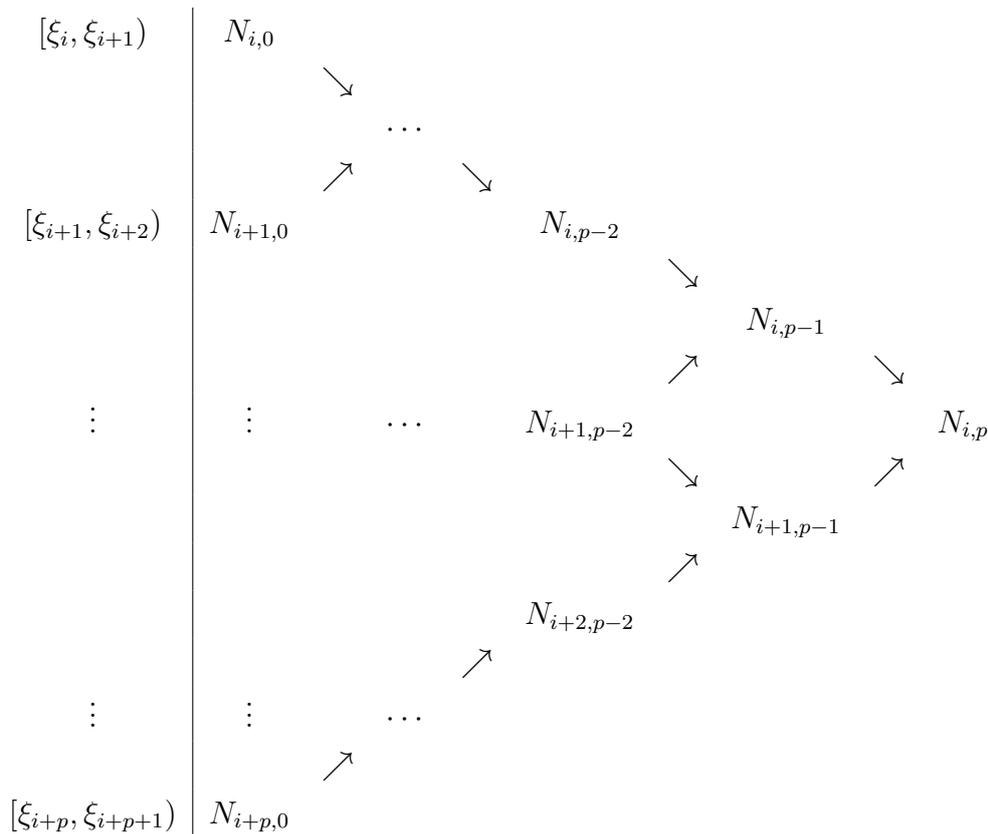
## 2.2 PROPERTIES

We now list a number of important properties of the B-spline basis functions, which determine many desirable geometric characteristics of B-spline curves and surfaces. Let us use the same notation as in Def. 2.1 and 2.5.

1. **Local support property.**  $N_{i,p}(\xi) = 0$  if  $\xi \notin [\xi_i, \xi_{i+p+1})$ .

*Proof.* Since Cox de Boor formula used to calculate B-spline basis functions is a recursion relation, a basis function of a given polynomial degree  $p$  depends on lower order and therefore this dependence forms a triangular pattern shown in Table 1. From the pattern it is clear that the basis function  $N_{i,p}$  is non-negative

Table 1: Triangular pattern of B-spline basis functions needed to prove the Local support property of B-spline basis functions.

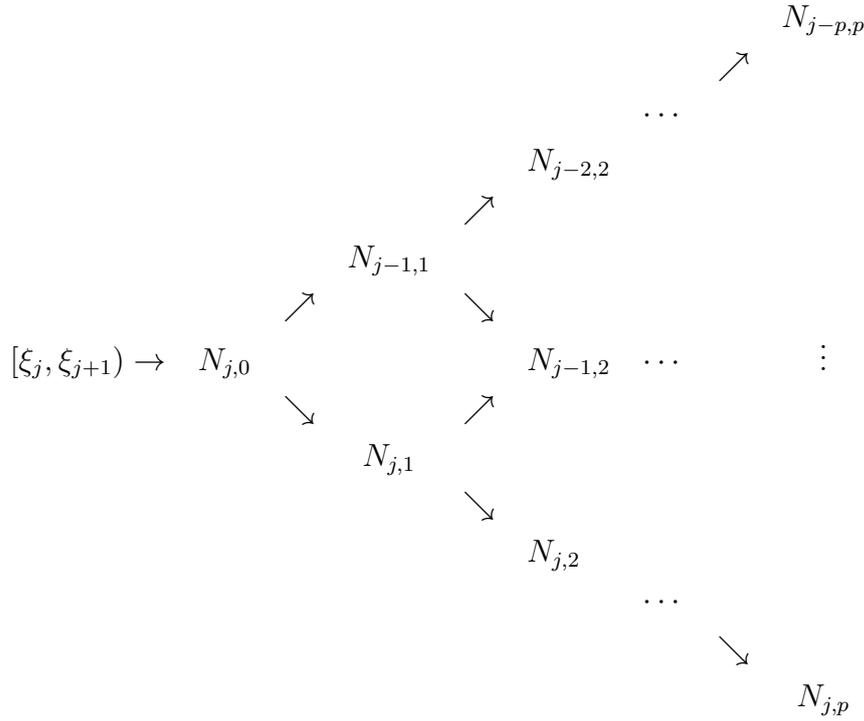


on the interval  $[\xi_i, \xi_{i+p+1})$ . □

2. In any given knot span,  $[\xi_j, \xi_{j+1})$ , at most  $p + 1$  of the  $N_{i,p}$  are non-zero, namely the functions  $N_{j-p,p}, \dots, N_{j,p}$ .

*Proof.* Using Cox de Boor recursion we get the triangular pattern shown in Table 2. From it, we can clearly conclude that the basis functions  $N_{j-p,p}, \dots, N_{j,p}$

Table 2: Triangular pattern of B-spline basis functions needed to prove the second property of B-spline basis functions.



are non zero for  $\xi \in [\xi_j, \xi_{j+1})$ . □

### 3. Non-negativity property. $N_{i,p}(\xi) \geq 0$ for all $i, p$ and $\xi$ .

*Proof.* We can prove this by induction on  $p$ . It is clearly true for  $p = 0$ . Assume it is true for  $p - 1$ ,  $p \geq 0$ , with  $i$  and  $\xi$  arbitrary. By definition

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi)$$

By the local support property,  $N_{i,p-1}(\xi) = 0$  if  $\xi \notin [\xi_i, \xi_{i+p})$ . But  $\xi \in [\xi_i, \xi_{i+p})$  implies  $\frac{\xi - \xi_i}{\xi_{i+p} - \xi_i}$  is non-negative. By assumption,  $N_{i,p-1}(\xi)$  is non-negative, and thus the first term is non-negative. The same is true for the second term, and hence  $N_{i,p}(\xi)$  is non-negative. □

### 4. Partition of unity property. For an arbitrary knot span, $[\xi_i, \xi_{i+1})$ ,

$$\sum_{j=i-p}^i N_{j,p}(\xi) = 1$$

for all  $\xi \in [\xi_i, \xi_{i+1})$ .

*Proof.* To prove this, consider interval  $[\xi_i, \xi_{i+1})$ . Here only  $N_{i-p,p}, \dots, N_{i,p}$  are non-zero. Then

$$\sum_{j=i-p}^i N_{j,p}(\xi) = \sum_{j=i-p}^i \frac{\xi - \xi_j}{\xi_{j+p} - \xi_j} N_{j,p-1}(\xi) + \sum_{j=i-p}^i \frac{\xi_{j+p+1} - \xi}{\xi_{j+p+1} - \xi_{j+1}} N_{j+1,p-1}(\xi).$$

Changing the summation index in the second sum from  $j$  to  $j+1$ , and considering that  $N_{i-p,p-1}(\xi) = N_{i+1,p-1}(\xi) = 0$ , we have

$$\sum_{j=i-p}^i N_{j,p}(\xi) = \sum_{j=i-p+1}^i \left[ \frac{\xi - \xi_j}{\xi_{j+p} - \xi_j} + \frac{\xi_{j+p} - \xi}{\xi_{j+p} - \xi_j} \right] N_{j,p-1}(\xi) = \sum_{j=i-p+1}^i N_{j,p-1}(\xi).$$

Applying the same concept recursively yields

$$\sum_{j=i-p}^i N_{j,p}(\xi) = \sum_{j=i-p+1}^i N_{j,p-1}(\xi) = \sum_{j=i-p+2}^i N_{j,p-2}(\xi) = \dots = \sum_{j=i}^i N_{j,0}(\xi) = 1.$$

□

5. All derivatives of  $N_{i,p}(\xi)$  exist in the interior of a knot span. At a knot  $N_{i,p}(\xi)$  is  $p - k$  times continuously differentiable, where  $k$  is the multiplicity of the knot. Hence, increasing degree increases continuity, and increasing knot multiplicity decreases continuity. Proof can be found in [16].
6. **Extrema property:** Except for the case  $p = 0$ ,  $N_{i,p}(\xi)$  attains exactly one maximum value. Proof can be found in [16].

The NURBS  $R_{i,p}(\xi)$  have similar properties as B-spline basis functions  $N_{i,p}(\xi)$ :

- non-negativity property,
- partition of unity property,
- extrema property and
- local support property.

*Remark 2.11.* Whenever the weights are constants, we get that  $\forall \xi \in \hat{\Omega} : R_{i,p}(\xi) = N_{i,p}(\xi)$  thanks to the partition of unity property of B-spline basis functions. This shows that polynomial B-splines are particular cases of NURBS.

### 3 ISOGEOMETRIC ANALYSIS

Following [5, 9], the root idea behind IgA is that the basis used to exactly model the geometry will also serve as the basis for the solution space of the numerical method. This concept of using the same basis for geometry and analysis is called the *isoparametric concept*. In IgA we select a basis capable of exactly representing the known geometry and use it as a basis for the domain we wish to approximate. A graphical representation of it is shown in Figure 3.

The above concepts are taken from [5, 9].

#### 3.1 GEOMETRIES

As already mentioned, in order to define B-spline or NURBS, we need a parametric space. The B-spline parametric space is local to “patches”, which play the role of subdomains within which element types and material models are assumed to be uniform. Element boundaries in the physical space are simply the images of knot lines under the B-spline mapping. The next definitions and properties are strictly following [5, 16].

B-splines curves in  $\mathbb{R}^d$  are constructed by taking a linear combination of B-spline basis functions. The coefficients of the basis functions are referred to as *control points*. Piecewise linear interpolation of the control points gives the so-called *control polygon*.

**Definition 3.1.** Given  $n$  B-spline basis functions  $N_{i,p}$ ,  $i = 1, 2, \dots, n$ , and corresponding control points  $\mathbf{B}_i \in \mathbb{R}^d$ ,  $i = 1, 2, \dots, n$ , a piecewise-polynomial *B-spline curve* is given by

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

Let  $\mathbf{C}(\xi)$  be a B-spline curve and  $p$  its polynomial degree. We now list important properties of B-spline curves:

1.  $\mathbf{C}(\xi)$  is a piecewise polynomial curve (since the  $N_{i,p}(\xi)$  are piecewise polynomials) that interpolates endpoints, that is, if  $\hat{\Omega} = [a, b]$ ,  $a, b \in \mathbb{R}$ , then  $\mathbf{C}(a) = \mathbf{B}_1$  and  $\mathbf{C}(b) = \mathbf{B}_n$ ; the number of control points is equal to the number of B-spline basis functions, i.e.  $n$ .
2. **Affine invariance.** Let us recall first the definition of an affine transformation.

**Definition 3.2.** Let  $\mathbf{x}$  be a point in  $\mathbb{R}^3$ . An *affine transformation*, denoted by  $\Phi$ , maps  $\mathbb{R}^3$  into  $\mathbb{R}^{3 \times 1}$  and has the form  $\Phi(\mathbf{x}) = A\mathbf{x} + \mathbf{v}$ , where  $A \in \mathbb{R}^{3 \times 3}$  is some matrix and  $\mathbf{v} \in \mathbb{R}^3$  is a vector. Affine transformations include translations, rotations, scalings and uniform stretching and shearings.

An affine transformation is applied to the B-spline curve by applying it to the control points. Proof can be found in [16].

3. **Strong convex hull property** For a B-spline curve of degree  $p$ , we define the convex hull as the union of all of the convex hulls formed by  $p + 1$  successive control points. The curve is contained in the convex hull of its control polygon; in fact, if  $\xi \in [\xi_i, \xi_{i+1})$ , then  $\mathbf{C}(\xi)$  is in the convex hull of the control points  $\mathbf{B}_{i-p}, \dots, \mathbf{B}_i$ .

*Proof.* Since  $\xi \in [\xi_i, \xi_{i+1})$  we know by the second property of B-spline basis functions that only  $p + 1$  B-spline basis functions are non-negative, namely  $N_{i-p,p}, \dots, N_{i,p}$ . Therefore

$$\mathbf{C}(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i = \sum_{j=i-p}^i N_{j,p}(\xi) \mathbf{B}_j.$$

By the non-negativity and partition of unity properties of B-spline basis functions  $N_{i,p}$  the proof is concluded.  $\square$

4. **Local control property.** Changing  $\mathbf{B}_i$  affects  $\mathbf{C}(\xi)$  only on the interval  $[\xi_i, \xi_{i+p+1})$ . See Figure 2.
5. The continuity and differentiability of  $\mathbf{C}(\xi)$  follow from that of the  $N_{i,p}(\xi)$  (since  $\mathbf{C}(\xi)$  is just a linear combination of the  $N_{i,p}(\xi)$ ). Thus,  $\mathbf{C}(\xi)$  is infinitely differentiable in the interior of knot intervals, and it is at least  $p - k$  times continuously differentiable at a knot of multiplicity  $k$ .

*Remark 3.3.* In general, control points are not interpolated by B-spline curves. The curve is interpolatory at the first and last points, due to the fact that the knot vector is open and also at the control points that have multiplicity equal to the polynomial degree  $p$ . The curve is tangent to the control polygon at these points.

B-spline geometries of dimension greater than one are defined by means of tensor product rules from the B-spline curves.

**Definition 3.4.** Given a *control net*  $\{\mathbf{B}_{i,j}\}$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ , polynomial degrees  $p$  and  $q$  and knot vectors  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ , and  $H = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$ ,

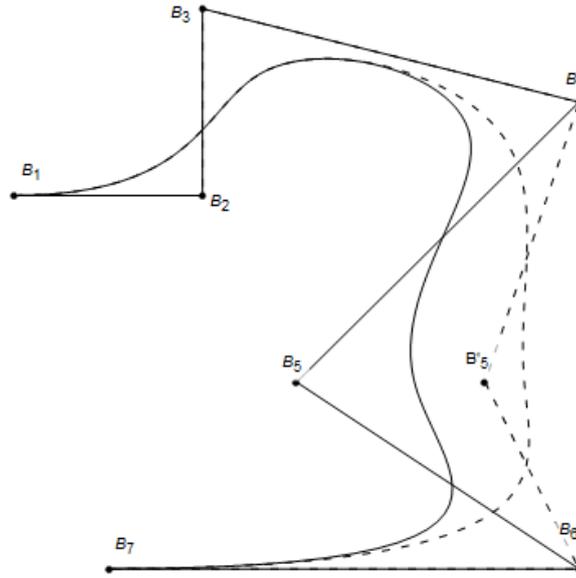


Figure 2: A cubic curve on  $\Xi = \{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$ . Moving  $\mathbf{B}_5$  (to  $\mathbf{B}'_5$ ) changes the curve on the interval  $[1/4, 1)$ .

a tensor product *B-spline surface* is defined by

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) N_{j,q}(\eta) \mathbf{B}_{i,j},$$

where  $N_{i,p}(\xi)$  and  $N_{j,q}(\eta)$  are univariate B-spline basis functions of order  $p$  and  $q$ , corresponding to knot vectors  $\Xi$  and  $H$ , respectively.

The properties of the tensor product basis functions follow from the corresponding properties of the univariate B-spline basis functions (see [16]). Let  $\mathbf{S}(\xi, \eta)$  be a B-spline surface, where  $\Xi$  has  $n + p + 1$  knots and  $H$  has  $m + q + 1$  knots as in the Def. 3.4. B-spline surfaces have the following properties:

1. The surface interpolates the four corner control points:  $\mathbf{S}(0, 0) = \mathbf{B}_{0,0}$ ,  $\mathbf{S}(1, 0) = \mathbf{B}_{n,0}$ ,  $\mathbf{S}(0, 1) = \mathbf{B}_{0,m}$  and  $\mathbf{S}(1, 1) = \mathbf{B}_{n,m}$ , for  $\Omega = [0, 1] \times [0, 1]$ . This follows from the partition of unity property of the tensor product basis functions ( $\sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) N_{j,q}(\eta) = 1$  for all  $(\xi, \eta) \in [0, 1] \times [0, 1]$ ) and the identities

$$N_{1,p}(0)N_{1,q}(0) = N_{n,p}(1)N_{1,q}(0) = N_{1,p}(0)N_{m,q}(1) = N_{n,p}(1)N_{m,q}(1) = 1.$$

2. **Affine invariance:** an affine transformation is applied to the surface by applying it to the control points. This follows from the partition of unity property of the tensor product B-spline basis functions.
3. **Strong convex hull property.** If  $(\xi, \eta) \in [\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}]$ , then  $\mathbf{S}(\xi, \eta)$  is in the convex hull of the control points  $\mathbf{B}_{h,k}$ ,  $i - p \leq h \leq i$  and  $j - q \leq k \leq j$ .

This follows from the non-negativity property of the tensor product B-spline basis functions ( $N_{i,p}(\xi)N_{j,q}(\eta) \geq 0$  for all  $i, j, p, q, \xi, \eta$ ), partition of unity property of the tensor product basis functions and from the property that says: in any given rectangle  $[\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}]$  at most  $(p+1)(q+1)$  basis functions are non-zero.

4. **Local control property.** If  $\mathbf{B}_{i,j}$  is changed, this affects the surface only in the rectangle  $[\xi_i, \xi_{i+p+1}] \times [\eta_j, \eta_{j+p+1}]$ . This follows from the property that  $N_{i,p}(\xi)N_{j,q}(\eta) = 0$  if  $(\xi, \eta)$  is outside the rectangle  $[\xi_i, \xi_{i+p+1}] \times [\eta_j, \eta_{j+p+1}]$ .
5. The continuity and differentiability of  $\mathbf{S}(\xi, \eta)$  follows from that of the basis functions. In particular, if a knot  $\xi_i$  has multiplicity  $k$ ,  $\mathbf{S}(\xi, \eta)$  is  $p - k$  times differentiable in the  $\xi$  and it is  $q - k$  times differentiable in the  $\eta$  direction if a knot  $\eta_j$  has multiplicity  $k$ .

We will now proceed to the definitions and properties of NURBS geometries. Before that, let us consider the formal definition of the control points for a NURBS curve.

**Definition 3.5.** Let  $\{\mathbf{B}_i^w\}$  be the set of control points for a B-spline curve in  $\mathbb{R}^{d+1}$  with knot vector  $\Xi$ . The set of *control points for a NURBS curve* in  $\mathbb{R}^d$  is defined as

$$(\mathbf{B}_i)_j = \frac{(\mathbf{B}_i^w)_j}{w_i}, \quad j = 1, 2, \dots, d, \quad w_i = (\mathbf{B}_i^w)_{d+1},$$

where  $(\mathbf{B}_i)_j$  is the  $j$ -th component of the vector  $\mathbf{B}_i$  and  $w_i$  is the  $i$ -th weight.

Similarly, as for B-spline curves and surfaces, we can define the NURBS curves and surfaces.

**Definition 3.6.** Given  $n$  NURBS basis functions  $R_{i,p}$ ,  $i = 1, 2, \dots, n$  and corresponding control points  $\mathbf{B}_i$  (as in Def. 3.5),  $i = 1, 2, \dots, n$ , a *NURBS curve* is given by

$$\mathbf{C}(\xi) = \sum_{i=1}^n R_{i,p}(\xi) \mathbf{B}_i.$$

**Definition 3.7.** Given a control net  $\{\mathbf{B}_{i,j}\}$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ , polynomial degrees  $p$  and  $q$  and knot vectors  $\Xi$  and  $H$ , a *NURBS surface* is given by

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m R_{i,p}(\xi) R_{j,q}(\eta) \mathbf{B}_{i,j},$$

where  $R_{i,p}(\xi)$  and  $R_{j,q}(\eta)$  are univariate NURBS basis functions of order  $p$  and  $q$  corresponding to knot vectors  $\Xi$  and  $H$ , respectively.

Similarly, also NURBS curves/surfaces have the same properties as B-spline curves/surfaces:

- they are interpolatory at endpoints,

- they have affine invariance property,
- they have strong convex hull property,
- they have differentiability property and
- they have local control property.

## 3.2 ISOGEOMETRIC SPACES

This section is about isogeometric spaces and the definitions are taken from [2, 9].

Let us denote the domain in the physical space by  $\Omega$ . Similarly, let us denote the domain in the parametric space by  $\hat{\Omega}$ . In our work,  $\hat{\Omega} = [0, 1]^d$  depending on the dimension  $d = 1, 2$  that we are observing. Thus,  $\mathbf{G} : \hat{\Omega} \rightarrow \Omega$  is the geometry mapping, taking points in the parametric space,  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_d)$ , and returning the corresponding points in the physical space,  $\mathbf{x} = (x_1, \dots, x_d)$ . Also we assume that this map is invertible, so  $\mathbf{G}^{-1} : \Omega \rightarrow \hat{\Omega}$  takes points in the physical domain and identifies their corresponding parameter values.

We define also the functions  $\varphi : \hat{\Omega} \rightarrow \mathbb{R}$  and  $\psi : \Omega \rightarrow \mathbb{R}$  as it can be seen in Figure 3. The characteristic of isogeometry is: to go from  $\Omega$  to  $\mathbb{R}$  we don't use the  $\psi$  function, but we go around using  $\varphi \circ \mathbf{G}^{-1}$ . So for  $\mathbf{x} = (x_1, x_2) \in \Omega$  and  $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \hat{\Omega}$  it holds that:

$$\psi(x_1, x_2) = \varphi \circ \mathbf{G}^{-1}(x_1, x_2) = \varphi(\mathbf{G}^{-1}(x_1, x_2)) = \varphi(\xi_1, \xi_2).$$

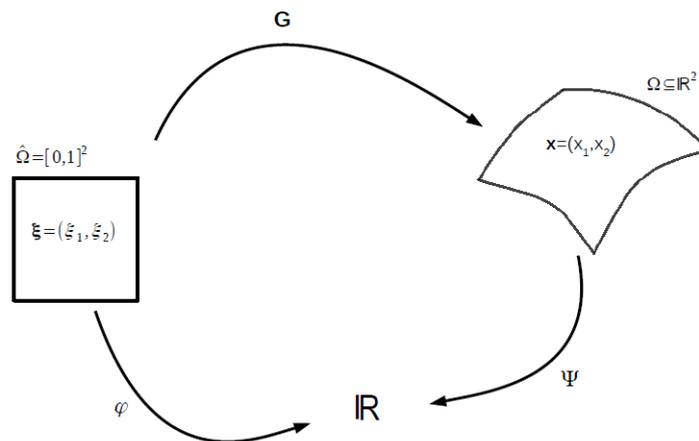


Figure 3: Graphical representation of the isoparametric concept in 2D.

B-splines or NURBS-based isogeometric analysis uses the B-splines or the NURBS basis functions in order to approximate the solution of a PDE.

More precisely, in one dimension, the geometry mapping  $\mathbf{G}$  couldn't be other than the linear mapping since  $\hat{\Omega} = [0, 1]$  and  $\Omega$  are intervals. Therefore the finite approximation space in which the numerical solution lies is

$$\mathbb{B}_p = \text{span}\{\varphi : \varphi \text{ is a linearly reparametrized B-spline or NURBS basis function of degree } p \}.$$

In two dimensions, let  $\mathbf{G} : [0, 1]^2 \rightarrow \Omega$  be the parametrization of the physical domain of the form  $\mathbf{G}(\boldsymbol{\xi}) = (G_1(\xi_1, \xi_2), G_2(\xi_1, \xi_2))$ . Recall that the Jacobian is defined as:

$$J_{\mathbf{G}}(\xi_1, \xi_2) = \begin{bmatrix} \frac{\partial G_1(\xi_1, \xi_2)}{\partial \xi_1} & \frac{\partial G_2(\xi_1, \xi_2)}{\partial \xi_1} \\ \frac{\partial G_1(\xi_1, \xi_2)}{\partial \xi_2} & \frac{\partial G_2(\xi_1, \xi_2)}{\partial \xi_2} \end{bmatrix}.$$

We assume that the Jacobian of  $\mathbf{G}$  and its inverse are non-singular. The space of B-spline or NURBS basis functions on  $\Omega$  is defined as:

$$\mathbb{B}_{p,q} = \text{span}\{\varphi \circ \mathbf{G}^{-1} : \varphi \text{ is a tensor product of two B-spline or NURBS basis functions of degree } p \text{ and } q\}.$$

### 3.3 REFINEMENTS

In order to have more precise isogeometric solutions to the PDE using B-splines or NURBS basis functions, we use three different procedures called *refinements*. With refinements we enrich the basis, but the underlying geometry and its parametrization remain intact. By doing refinements we have control over the element size, the degree of the basis and we can control the continuity of the basis. We will present three refinements: knot insertion, degree elevation and k-refinement.

The first mechanism by which one can enrich the basis is *knot insertion*. Knots may be inserted without changing a curve geometrically or parametrically. Given a knot vector  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ , we introduce the notion of an extended knot vector  $\bar{\Xi} = \{\bar{\xi}_1 = \xi_1, \bar{\xi}_2, \dots, \bar{\xi}_{n+m+p+1} = \xi_{n+p+1}\}$  such that  $\Xi \subset \bar{\Xi}$ , which has  $m$  new knots. Moreover, let  $\mathbf{B}$  be the original set of control points of cardinality  $n$ . Then the new set of control points, say  $\bar{\mathbf{B}}$  that one has to take in order to obtain the exact same geometry is defined by the transformation:  $\bar{\mathbf{B}} = T^p \mathbf{B}$ , where  $T^p$  is defined recursively as:

- for  $q = 0$ :

$$T_{i,j}^0 = \begin{cases} 1, & \text{if } \xi_j \leq \bar{\xi}_i < \xi_{j+1} \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

- for  $q = 1, 2, \dots, p$ :

$$T_{i,j}^q = \frac{\bar{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j} T_{i,j}^{q-1} + \frac{\xi_{j+q+1} - \bar{\xi}_{i+q}}{\xi_{j+q+1} - \xi_{j+1}} T_{i,j+1}^{q-1} \quad (3.2)$$

for all  $i = 1, 2, \dots, n + m$  and for all  $j = 1, 2, \dots, n$ .

Knot values already present in the knot vector may be repeated in this way, thereby increasing their multiplicity, the continuity of the basis will be reduced. However, continuity of the curve is preserved by choosing the control points in this way. Each unique internal knot value may appear no more than  $p$  times or the curve becomes discontinuous. Each time that we do knot insertion, we increase the number of control points, elements and basis function, but the curve remains unchanged.

The second mechanism by which one can enrich the basis is *degree elevation*. The process involves raising the polynomial degree of the basis functions used to represent the geometry. Recall that the continuity of the basis at each knot is  $p - k$ , where  $k$  is the multiplicity of the knot. Consequently, when  $p$  is increased,  $k$  must also be increased if we want to preserve the discontinuities in the various derivatives already existing in the original curve. During degree elevation, the multiplicity of each knot value is increased by one, but no new knot values are added. The locations of the control points change, but the elevated curve is geometrically and parametrically identical to the original curve. The process for doing this involves subdividing the curve into many B-spline curves by knot insertion, degree elevating each of these individual segments, and then removing the unnecessary knots to combine the segments into one, degree-elevated, B-spline curve.

Finally, *k-refinement* is a combination of degree elevation and knot insertion and it stems from the fact that the processes of degree elevation and knot insertion do not commute. More precisely, it consists first of elevating the degree from some  $p$  to some  $q > p$ , and then of adding a knot  $\bar{\xi}$  into the knot vector so that the basis functions at  $\bar{\xi}$  are  $C^{q-1}$  continuous. While, if a knot  $\bar{\xi}$  is inserted before elevating the degree from  $p$  to  $q > p$ , then the basis functions at  $\bar{\xi}$  would only be  $C^{p-1}$  continuous.

These definitions were taken from [4, 9]. For more details about refinements, we refer the interested reader to [5].

### 3.4 $L^2$ APPROXIMATION

In this sub-chapter, we present the theory needed to construct the best approximant using  $L^2$  approximation. We also present the normed spaces which we will use in the next chapters. The next definitions and theorems are taken from [5, 17, 20].

**Definition 3.8.** The space of square-integrable functions on  $\Omega \subset \mathbb{R}^d$  is defined as

$$L^2(\Omega) = \left\{ u : \Omega \mapsto \mathbb{R} \text{ such that } \int_{\Omega} u^2 \, d\Omega < +\infty \right\}.$$

The corresponding norm is defined as

$$\|u\|_{L^2(\Omega)} = \sqrt{\langle u, u \rangle_{L^2(\Omega)}} = \sqrt{\int_{\Omega} u^2 d\Omega}.$$

The aim of the  $L^2$  approximation is: given a function  $f \in X$  we want to approximate it with another function  $\tilde{f} \in S \subseteq X$ . Let  $X = L^2(\Omega)$  be equipped with the norm as in Def. 3.8 and let  $S \subseteq X$ . For any functions  $f, g \in X$  we define the continuous inner product as

$$\langle f, g \rangle_{L^2(\Omega)} = \int_{\Omega} f \cdot g d\Omega.$$

In the subset  $S$  we have ensured the existence and uniqueness of the best approximant (see [20], Chapter 9).

**Theorem 3.9.** *Let  $X = L^2(\Omega)$  and  $S \subseteq X$ . The element  $\tilde{f} \in S$  is the best approximant by the  $L^2$  approximation method for  $f \in X$  if and only if  $\langle f - \tilde{f}, s \rangle_{L^2(\Omega)} = 0$  for all  $s \in S$ .*

*Proof.* For brevity we will write  $\|\cdot\|$  and  $\langle \cdot, \cdot \rangle$  instead of  $\|\cdot\|_{L^2(\Omega)}$  and  $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$ , respectively.

Suppose  $\langle f - \tilde{f}, s \rangle = 0$  for each  $s \in S$ . Then

$$\begin{aligned} \|f - s\|^2 &= \|f - \tilde{f} + \tilde{f} - s\|^2 = \int_{\Omega} (f - \tilde{f} + \tilde{f} - s)^2 d\Omega = \\ &= \langle f - \tilde{f}, f - \tilde{f} \rangle + 2\langle f - \tilde{f}, \tilde{f} - s \rangle + \langle \tilde{f} - s, \tilde{f} - s \rangle = \\ &= \|f - \tilde{f}\|^2 + \underbrace{2\langle f - \tilde{f}, \tilde{f} - s \rangle}_{=0, \text{ since } \tilde{f} - s \in S} + \|\tilde{f} - s\|^2 = \\ &= \|f - \tilde{f}\|^2 + \underbrace{\|\tilde{f} - s\|^2}_{\geq 0} \geq \|f - \tilde{f}\|^2. \end{aligned}$$

For the other direction, let  $s \in S$ , let  $\lambda > 0$  be an arbitrary real number and let  $\tilde{f}$  be the best approximant. We need to prove that  $\langle f - \tilde{f}, s \rangle = 0$  for all  $s \in S$ . Since  $\tilde{f} + \lambda s$  is worse than  $\tilde{f}$ , we have:

$$\begin{aligned} 0 &\leq \|f - \tilde{f} + \lambda s\|^2 - \|f - \tilde{f}\|^2 = \\ &= \|f - \tilde{f}\|^2 + 2\lambda\langle f - \tilde{f}, s \rangle + \lambda^2\|s\|^2 - \|f - \tilde{f}\|^2 = \\ &= 2\lambda\langle f - \tilde{f}, s \rangle + \lambda^2\|s\|^2. \end{aligned}$$

When  $\lambda$  converges to 0, also  $\lambda^2$  converges even faster to zero. Thus,  $\lambda^2\|s\|^2$  can be neglected and we have that  $\langle f - \tilde{f}, s \rangle \geq 0$ . Since  $s$  has been chosen arbitrarily and we know that  $-s \in S$  we have also that  $\langle f - \tilde{f}, -s \rangle \geq 0$ , which multiplied by -1 becomes  $\langle f - \tilde{f}, s \rangle \leq 0$ . So,  $\langle f - \tilde{f}, s \rangle = 0$  as desired.  $\square$

Let  $\{N_{i,p}\}$ ,  $i = 1, 2, \dots, n$  be the basis of the space  $S$ . It follows that we can write  $\tilde{f} \in S$  as  $\tilde{f} = \sum_{i=1}^n \alpha_i N_{i,p}$  for  $\alpha_i \in \mathbb{R}$ . By Thm. 3.9, it must hold that  $f - \sum_{i=1}^n \alpha_i N_{i,p} \perp S$ , which we can write as

$$\sum_{i=1}^n \alpha_i \langle N_{i,p}, N_{j,p} \rangle = \langle f, N_{j,p} \rangle \quad \text{for } j = 1, 2, \dots, n. \quad (3.3)$$

Now, let us denote by  $S_{ij} = \langle N_{i,p}, N_{j,p} \rangle$  and  $F_j = \langle f, N_{j,p} \rangle$ , where the inner product is defined as in Def 3.8. Moreover, let  $\mathbf{S} = [S_{ij}]$ ,  $\mathbf{F} = [F_j]$  and  $\mathbf{a} = [\alpha_i]$ . For  $i, j = 1, 2, \dots, n$  we can then write Eq. (3.3) in the matrix form as

$$\mathbf{S}\mathbf{a} = \mathbf{F}$$

where  $\mathbf{S}$  is a symmetric positive definite *Gram matrix*.

*Remark 3.10.* Note that if  $\Omega \subseteq \mathbb{R}^2$ , the geometry mapping  $\mathbf{G}$  is not a linear map and therefore we need to be careful, when we reparametrize the domain  $\hat{\Omega} = [0, 1]^2$  into  $\Omega$ . So the inner products in Eq. (3.3) become:

$$\begin{aligned} S_{ij} &= \langle N_{i,p}, N_{j,p} \rangle = \int_{\Omega} N_{i,p}(x_1, x_2) \cdot N_{j,p}(x_1, x_2) \, d\mathbf{x} = \\ &= \int_{[0,1]^2} N_{i,p}(\xi_1, \xi_2) \cdot N_{j,p}(\xi_1, \xi_2) \cdot |\det J_{\mathbf{G}}(\xi_1, \xi_2)| \, d\xi, \\ F_i &= \langle f, N_{i,p} \rangle = \int_{\Omega} f(x_1, x_2) \cdot N_{i,p}(x_1, x_2) \, d\mathbf{x} = \\ &= \int_{[0,1]^2} f(\xi_1, \xi_2) \cdot N_{i,p}(\xi_1, \xi_2) \cdot |\det J_{\mathbf{G}}(\xi_1, \xi_2)| \, d\xi. \end{aligned}$$

Our aim in this work is to estimate the error between the exact solution and the approximated solution obtained using the isogeometric collocation method. Mathematically, the notion of the measure of the error implies a particular choice of norm and therefore of the inner product. Next, we will define other normed spaces that we will observe in the next chapters, specially in Chapter 7, where we will use this norms to calculate the relative errors of the approximated functions. But firstly, let us define the derivative operator.

**Definition 3.11.** Let  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_d) \in \mathbb{N}^d$ , where  $d$  is the spatial dimension, be a multi-index (that is:  $d$ -tuple of non-negative integers) such that the order  $|\boldsymbol{\alpha}| = \sum_{i=1}^d \alpha_i$  is at most  $k \in \mathbb{N}$ . The *derivative operator* is denoted as

$$D^{\boldsymbol{\alpha}}u = \frac{\partial^{|\boldsymbol{\alpha}|}u}{\partial x_1^{\alpha_1} \cdot \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}} = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \cdot \frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} \cdots \frac{\partial^{\alpha_d}}{\partial x_d^{\alpha_d}}u.$$

**Definition 3.12.** Let  $k = 1$ . The *Sobolev space*  $H_1(\Omega)$  is the space of functions that have square integrable derivatives and it is defined as

$$H_1(\Omega) = \{u : D^{\boldsymbol{\alpha}}u \in L^2(\Omega), |\boldsymbol{\alpha}| \leq 1\}.$$

The associated norm is given by

$$\|u\|_{H_1(\Omega)} = \sqrt{\langle u, u \rangle_{H_1(\Omega)}},$$

while the inner product is defined as:

$$\langle u, v \rangle_{H_1(\Omega)} = \sum_{|\alpha| \leq 1} \int_{\Omega} D^{\alpha} u \cdot D^{\alpha} v \, d\Omega,$$

where  $u$  and  $v$  are two functions in  $\Omega$ .

**Definition 3.13.** Let  $k = 2$ . The Sobolev space  $H_2(\Omega)$  is the space that have, not just square integrable first derivatives, but also square integrable second derivatives, and it is defined as

$$H_2(\Omega) = \{u : D^{\alpha} u \in L^2(\Omega), |\alpha| \leq 2\}.$$

The norm associated with  $H_2(\Omega)$  is given by

$$\|u\|_{H_2(\Omega)} = \sqrt{\langle u, u \rangle_{H_2(\Omega)}},$$

while the inner product is defined as:

$$\langle u, v \rangle_{H_2(\Omega)} = \sum_{|\alpha| \leq 2} \int_{\Omega} D^{\alpha} u \cdot D^{\alpha} v \, d\Omega.$$

## 4 POISSON'S PARTIAL DIFFERENTIAL EQUATION

We will now define the strong form of Poisson's PDE as boundary value problem, which is an elliptic, linear, second-order PDE. The definitions are taken from [5, 17, 19].

**Definition 4.1.** Consider a domain  $\Omega \subset \mathbb{R}^d$ , i.e. an open, bounded and connected set, and let  $\partial\Omega$  be its boundary. Also let  $\mathbf{x} = (x_1, x_2, \dots, x_d)$  be a  $d$ -tuple. The *strong formulation of the Poisson equation* is

$$\Delta u(\mathbf{x}) = f(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega,$$

where  $f = f(\mathbf{x})$  is a given function and the symbol  $\Delta$  denotes the Laplacian operator, which is defined as

$$\Delta u(\mathbf{x}) = \sum_{i=1}^d \frac{\partial^2 u}{\partial x_i^2}.$$

To obtain a unique solution, we need to add suitable boundary conditions, which means that we need information about the behaviour of the solution  $u = u(\mathbf{x})$  at the domain boundary  $\partial\Omega$ . For instance, we can assign different boundary conditions:

- *Dirichlet boundary condition:* the value of the solution  $u$  on the boundary is

$$u = g \text{ on } \partial\Omega,$$

where  $g : \partial\Omega \rightarrow \mathbb{R}$  is a given function. If  $g = 0$ , the boundary condition is said to be homogeneous.

- *Neumann boundary condition:* the value of the normal derivative of  $u$  can be imposed as

$$\Delta u \cdot \mathbf{n} = \frac{\partial u}{\partial \mathbf{n}} = h \text{ on } \partial\Omega,$$

where  $\mathbf{n}$  is the outward unit normal vector on  $\partial\Omega$  and  $h : \partial\Omega \rightarrow \mathbb{R}$  is a given function. If  $h = 0$ , the condition is said to be homogeneous.

- *Mixed boundary conditions:* different types of conditions can be assigned to different portions of the boundary of the computational domain  $\Omega$ . For instance, if  $\partial\Omega = \Gamma_D \cup \Gamma_N$  such that  $\text{Int}(\Gamma_D) \cap \text{Int}(\Gamma_N) = \emptyset$ , we can impose the mixed boundary conditions:

$$\begin{cases} u = g \text{ on } \Gamma_D \\ \frac{\partial u}{\partial \mathbf{n}} = h \text{ on } \Gamma_N. \end{cases}$$

From now on, we will only consider the Poisson's PDE with Dirichlet boundary condition and we will focus specially on one and two dimensional cases.

## 4.1 WEAK FORM APPROXIMATION BASED ON ISOGEOMETRIC ANALYSIS

In this sub-chapter, we will derive the weak formulation of the Poisson's equation with Dirichlet boundary condition for the one and two dimensional cases, which we will need for the Galerkin method. To get the weak formulation, we need to characterize two classes of functions: the collection of trial solutions and the set of weighting functions.

**Definition 4.2.** The collection of *trial solutions* is the collection of functions defined as:

$$S = \{u | u \in H^1(\Omega), u|_{\partial\Omega} = g\}.$$

The collection of weighting functions is a collection of functions which is similar to the trial solutions, but in this case we require the homogeneous counterpart of the Dirichlet boundary condition. The collection of *weighting functions* is defined as:

$$V = \{v | v \in H^1(\Omega), v|_{\partial\Omega} = 0\}.$$

Let us consider the next non-homogeneous one dimensional Poisson's PDE:

$$\begin{cases} u''(x) = f(x) \text{ for } x \in \Omega \\ u(x) = g(x) \text{ for } x \in \partial\Omega \end{cases} \quad (4.1)$$

where  $\Omega \subseteq \mathbb{R}$  is the physical space,  $u : \Omega \rightarrow \mathbb{R}$  is the solution, while  $f : \Omega \rightarrow \mathbb{R}$  and  $g : \partial\Omega \rightarrow \mathbb{R}$  are given sufficiently regular functions.

**Lemma 4.3.** *The weak formulation of the Poisson's PDE in one dimension is:*

$$-\int_{\Omega} u'(x)v'(x) dx = \int_{\Omega} f(x)v(x) dx$$

where  $f$  is a given function and  $u \in S$  is the solution that holds for all  $v \in V$ .

*Proof.* To obtain a weak formulation of the Poisson's equation we need to first multiply the equation with a test function  $v \in V$  and then we need to integrate it by parts:

$$\begin{aligned} \int_{\Omega} u''(x)v(x) dx &= \int_{\Omega} f(x)v(x) dx \\ [u'(x)v(x)] \Big|_{\partial\Omega} - \int_{\Omega} u'(x)v'(x) dx &= \int_{\Omega} f(x)v(x) dx. \end{aligned}$$

The boundary term vanishes because  $v|_{\partial\Omega} = 0$  for  $v \in V$  and so we get the desired.  $\square$

We can write the above equation in a more concise form as:

$$-a(u, v) = L(v)$$

where

$$a(u, v) = \int_{\Omega} u'(x)v'(x) dx \quad \text{and} \quad L(v) = \int_{\Omega} f(x)v(x) dx.$$

Let us now observe the two dimensional Poisson's PDE:

$$\begin{cases} \Delta u(\mathbf{x}) = f(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega \\ u(\mathbf{x}) = g(\mathbf{x}) \text{ for } \mathbf{x} \in \partial\Omega \end{cases} \quad (4.2)$$

where  $\Omega \subseteq \mathbb{R}^2$  is the physical domain,  $\mathbf{x} = (x_1, x_2)$ ,  $u : \Omega \rightarrow \mathbb{R}$  is the solution, while  $f : \Omega \rightarrow \mathbb{R}$  and  $g : \partial\Omega \rightarrow \mathbb{R}$  are given sufficiently regular functions.

**Lemma 4.4.** *The weak formulation of the Poisson's PDE in two dimensions is:*

$$- \int_{\Omega} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) d\mathbf{x},$$

where  $f$  is a given function and  $u \in S$  is the solution that holds for all  $v \in V$ .

*Proof.* As in the one-dimensional case, to get the weak form of the PDE, we have to multiply the equation with a test function  $v \in V$  and then integrate by parts:

$$\int_{\Omega} \Delta u v d\Omega = \int_{\Omega} f v d\Omega.$$

Note that for brevity we omit the variables, i.e. we write just the functions. Since we are in two dimensions, we need an extension of the formula of partial integration. Recall the divergence (Gauss) theorem that says:

$$\int_{\Omega} \nabla \circ \mathbf{a} d\Omega = \int_{\partial\Omega} \mathbf{a} \cdot \mathbf{n} d\gamma,$$

where  $\mathbf{a}(\mathbf{x}) = (a_1(\mathbf{x}), a_2(\mathbf{x}))^T$  is a sufficiently regular vector-valued function,  $\mathbf{n}(\mathbf{x}) = (n_1(\mathbf{x}), n_2(\mathbf{x}))^T$  is the outward unit normal vector on  $\partial\Omega$  and the symbol  $\nabla$  denotes the *divergence*, which in 2 dimensions is defined as

$$\nabla \circ \mathbf{a}(\mathbf{x}) = \sum_{i=1}^2 \frac{\partial a_i}{\partial x_i}.$$

If we apply the Gauss theorem first to the function  $\mathbf{a} = (uv, 0)^T$  and then to  $\mathbf{a} = (0, uv)^T$  and if we recall the product rule:  $\frac{\partial}{\partial x_i}(fg) = g \frac{\partial f}{\partial x_i} + f \frac{\partial g}{\partial x_i}$ , we get the relations:

$$\int_{\Omega} u \frac{\partial v}{\partial x_i} d\Omega + \int_{\Omega} v \frac{\partial u}{\partial x_i} d\Omega = \int_{\partial\Omega} uv n_i d\gamma, \quad i = 1, 2.$$

Using these last relations, the fact that  $\Delta u = \nabla \cdot \nabla \circ u = \sum_{i=1}^2 \frac{\partial}{\partial x_i} \left( \frac{\partial u}{\partial x_i} \right)$  and that  $v|_{\partial\Omega} = 0$ , since  $v \in V$ , we can modify the first integral in the next way:

$$\begin{aligned} \int_{\Omega} \Delta u v \, d\Omega &= \int_{\Omega} \sum_{i=1}^2 \frac{\partial}{\partial x_i} \left( \frac{\partial u}{\partial x_i} \right) v \, d\Omega = \\ &= - \sum_{i=1}^2 \int_{\Omega} \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i} \, d\Omega + \sum_{i=1}^2 \underbrace{\int_{\partial\Omega} \frac{\partial u}{\partial x_i} v n_i \, d\gamma}_{=0} = \\ &= - \int_{\Omega} \sum_{i=1}^2 \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i} \, d\Omega = - \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega \end{aligned}$$

□

Similarly as in the one dimensional case, we can write the weak formulation of the Poisson's PDE in two dimensions in a more concise form as:

$$-a(u, v) = L(v)$$

where in this case

$$a(u, v) = \int_{\Omega} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x} \quad \text{and} \quad L(v) = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x}.$$

Here we see that despite in the strong form solution  $u$  must have well defined second derivatives, the weak form only requires that first derivatives are square-integrable. Also it can be shown that the weak solution and the strong solution are equivalent almost everywhere (see [14]).

## 4.2 THE GALERKIN ISOGEOMETRIC METHOD

Galerkin method consists of constructing finite-dimensional approximations of  $S$  and  $V$ , denoted by  $S^h \subset S$  and  $V^h \subset V$ , which will be associated with subsets of the space spanned by the isoparametric basis. Let  $g^h$  be the approximation function of  $g$  obtained using the  $L^2$  approximation. We can characterize  $S^h$  by recognizing that if we have the given function  $g^h \in S^h$  such that  $g^h = g$  on  $\partial\Omega$ , then for every  $u^h \in S^h$  there exists a unique  $v^h \in V^h$  such that  $u^h = v^h + g^h$ . We can now write a variational equation of the weak form of the Poisson's PDE. The Galerkin form of the problem is:

$$\begin{aligned} \text{Given } g^h, \text{ find } u^h = v^h + g^h, \text{ where } v^h \in V^h, \text{ such that for all } w^h \in V^h: \\ -a(w^h, u^h) = L(w^h). \end{aligned}$$

Since the functions spaces used in the Galerkin method are finite-dimensional, we are dealing with a coupled system of linear algebraic equations. Let  $\mathbb{B}_p$  be the solution

space defined in Chapter 3.2. Without loss of generality we can assume a numbering for these functions such that there exists an integer  $n_{eq} < n$  such that  $\forall i = 1, 2, \dots, n_{eq}$ ,  $N_{i,p} = 0$  on  $\partial\Omega$ . Thus, for all  $w^h \in V^h$ , there exist constants  $c_i$ ,  $i = 1, 2, \dots, n_{eq}$  such that

$$w^h = \sum_{i=1}^{n_{eq}} c_i N_{i,p}.$$

Now, we can write that for any  $u^h \in S^h$  there exist  $d_i$ ,  $i = 1, 2, \dots, n_{eq}$  such that

$$u^h = v^h + g^h = \sum_{i=1}^{n_{eq}} d_i N_{i,p} + g^h. \quad (4.3)$$

Now we need to recall some definitions of functional analysis (which are taken from [17]).

**Definition 4.5.** Given a function space  $V$ , a *functional*  $F$  on  $V$  is an operator associating a real number to each element of  $V$ , that is:

$$F : V \mapsto \mathbb{R}.$$

A functional is said to be linear if

$$F(\lambda v + \mu w) = \lambda F(v) + \mu F(w) \quad \forall \lambda, \mu \in \mathbb{R}, \forall v, w \in V,$$

and it is said to be bounded if there is a constant  $c > 0$  such that

$$|F(v)| \leq c \|v\|_V \quad \forall v \in V.$$

**Definition 4.6.** Given a normed functional space  $V$  we call *form* an application which associates to each pair of elements of  $V$  a real number,  $a : V \times V \mapsto \mathbb{R}$ . A form is called *bilinear* if it is linear with respect to both arguments, that is, if

$$\begin{aligned} a(\lambda u + \mu w, v) &= \lambda a(u, v) + \mu a(w, v) & \forall \lambda, \mu \in \mathbb{R}, \forall u, v, w \in V, \\ a(u, \lambda w + \mu v) &= \lambda a(u, w) + \mu a(u, v) & \forall \lambda, \mu \in \mathbb{R}, \forall u, v, w \in V. \end{aligned}$$

A form is called *symmetric* if  $a(u, v) = a(v, u)$ ,  $\forall u, v \in V$ .

Now, since  $a(\cdot, \cdot)$  is bilinear, we can derive from the Galerkin form that for all  $w^h \in V^h$ :

$$a(w^h, v^h) = -L(w^h) - a(w^h, g^h) \longrightarrow \sum_{i=1}^{n_{eq}} c_i \left( \sum_{j=1}^{n_{eq}} a(N_{i,p}, N_{j,p}) d_j + L(N_{i,p}) + a(N_{i,p}, g^h) \right) = 0.$$

As the  $c_i$ 's are arbitrary, it follows that the term in parentheses must vanish; thus, for  $i = 1, 2, \dots, n_{eq}$  we have

$$\sum_{j=1}^{n_{eq}} a(N_{i,p}, N_{j,p}) d_j = -L(N_{i,p}) - a(N_{i,p}, g^h). \quad (4.4)$$

Now, define  $K_{ij} = a(N_{i,p}, N_{j,p})$ ,  $F_i = L(N_{i,p}) + a(N_{i,p}, g^h)$  and moreover  $\mathbf{K} = [K_{ij}]$ ,  $\mathbf{F} = [F_i]$  and  $\mathbf{d} = [d_i]$ . For  $i, j = 1, 2, \dots, n_{eq}$  we can rewrite the Eq. (4.4) as a linear system

$$\mathbf{K}\mathbf{d} = -\mathbf{F},$$

where  $\mathbf{K}$  is usually called the stiffness matrix,  $\mathbf{F}$  force vector and  $\mathbf{d}$  the displacement vector. We can now get  $d_i$  for  $i = 1, 2, \dots, n_{eq}$  as  $\mathbf{d} = \mathbf{K}^{-1}(-\mathbf{F})$  and inserting them in Eq. (4.3) we can finally write the solution  $u^h$ :

$$u^h = \sum_{i=1}^{n_{eq}} d_i N_{i,p} + g^h.$$

If we are in a two dimensional space, we need to use the geometry mapping to parametrize our domain, similarly as we showed in the  $L^2$  approximation.

**Lemma 4.7.** *Let us use the same notation as in the Chapter 3.2. Let  $\psi_i : \Omega \rightarrow \mathbb{R}$  and  $\varphi_i : [0, 1]^2 \rightarrow \mathbb{R}$  be elements in the finite dimensional space  $V^h$  for  $i = 1, 2$ . Let  $\mathbf{G} : [0, 1]^2 \rightarrow \Omega$  be the geometry mapping. Then the next equality holds:*

$$\int_{\Omega} \nabla \psi_i(\mathbf{x}) \cdot \nabla \psi_j(\mathbf{x}) \, d\mathbf{x} = \int_{[0,1]^2} (J_{\mathbf{G}}(\boldsymbol{\xi})^{-1} \nabla \varphi_i(\boldsymbol{\xi})) \cdot (J_{\mathbf{G}}(\boldsymbol{\xi})^{-1} \nabla \varphi_j(\boldsymbol{\xi})) \cdot |\det J_{\mathbf{G}}(\boldsymbol{\xi})| \, d\boldsymbol{\xi}$$

where  $J_{\mathbf{G}}$  denotes the Jacobian of  $\mathbf{G}$ .

*Proof.* We need to show that  $\nabla \varphi(\boldsymbol{\xi}) = J_{\mathbf{G}}(\boldsymbol{\xi}) \nabla \psi(\mathbf{x})$ . Let us look at

$$\nabla \varphi(\boldsymbol{\xi}) = \begin{bmatrix} \frac{\partial \varphi(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_1} \\ \frac{\partial \varphi(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \psi(\mathbf{G}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2))}{\partial \xi_1} \\ \frac{\partial \psi(\mathbf{G}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2))}{\partial \xi_2} \end{bmatrix}$$

Using the chain rule  $\frac{\partial f(x(t), y(t))}{\partial t} = \frac{\partial f}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dt}$  we get

$$\begin{aligned} \nabla \varphi(\boldsymbol{\xi}) &= \begin{bmatrix} \frac{\partial \psi(x_1, x_2)}{\partial x_1} \cdot \frac{\partial G_1(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_1} + \frac{\partial \psi(x_1, x_2)}{\partial x_2} \cdot \frac{\partial G_2(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_1} \\ \frac{\partial \psi(x_1, x_2)}{\partial x_1} \cdot \frac{\partial G_1(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_2} + \frac{\partial \psi(x_1, x_2)}{\partial x_2} \cdot \frac{\partial G_2(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_2} \end{bmatrix} = \\ &= \begin{bmatrix} \frac{\partial G_1(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_1} & \frac{\partial G_2(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_1} \\ \frac{\partial G_1(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_2} & \frac{\partial G_2(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)}{\partial \xi_2} \end{bmatrix} \begin{bmatrix} \frac{\partial \psi(x_1, x_2)}{\partial x_1} \\ \frac{\partial \psi(x_1, x_2)}{\partial x_2} \end{bmatrix} = J_{\mathbf{G}}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) \nabla \psi(x_1, x_2). \end{aligned}$$

□

In our case  $N_{i,p} \in V^h$  so by using Lemma 4.7 on the two forms in Eq. (4.4), we get

$$\begin{aligned} a(N_{i,p}, N_{j,p}) &= \int_{\Omega} \nabla N_{i,p}(\mathbf{x}) \cdot \nabla N_{j,p}(\mathbf{x}) \, d\mathbf{x} = \\ &= \int_{[0,1]^2} (J_{\mathbf{G}}(\boldsymbol{\xi})^{-1} \nabla N_{i,p}(\boldsymbol{\xi})) \cdot (J_{\mathbf{G}}(\boldsymbol{\xi})^{-1} \nabla N_{j,p}(\boldsymbol{\xi})) \cdot |\det J_{\mathbf{G}}(\boldsymbol{\xi})| \, d\boldsymbol{\xi} \\ L(N_{i,p}) &= \int_{\Omega} f(\mathbf{x}) N_{i,p}(\mathbf{x}) \, d\mathbf{x} = \int_{[0,1]^2} f(\mathbf{G}(\boldsymbol{\xi})) \cdot N_{i,p}(\boldsymbol{\xi}) \cdot |\det J_{\mathbf{G}}(\boldsymbol{\xi})| \, d\boldsymbol{\xi}. \end{aligned}$$

Clearly, if we will have NURBS instead of B-splines basis functions, the procedure would be the same.

**Example 4.8.** Let  $\Xi = \{0, 0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1, 1\}$ . In Figure 4 we approximate the one dimensional homogeneous Poisson's PDE:

$$\begin{cases} u''(x) = -\frac{9}{2}\pi^2 \sin(3\pi x), & x \in (0, 1) \\ u(0) = u(1) = 0, \end{cases},$$

with GIM, using  $\Xi$  as the uniform and open knot vector.

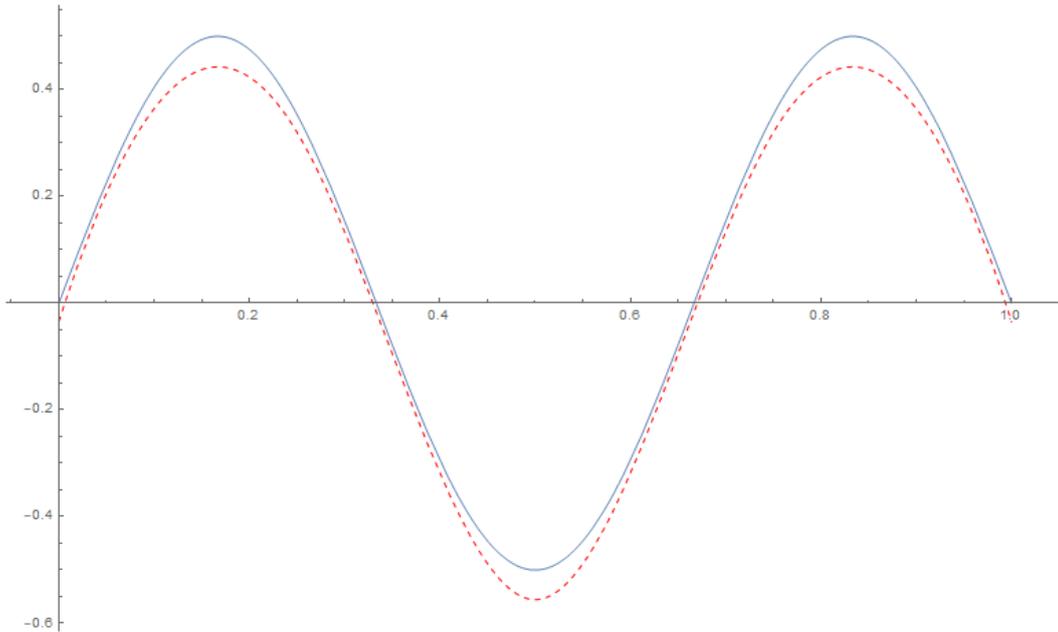


Figure 4: The exact solution  $u(x) = \frac{1}{2} \sin(3\pi x)$  (in blue) and the approximated solution (in red) obtained by using GIM.

## 5 ISOGEOMETRIC COLLOCATION METHOD

In contrast to the Galerkin isogeometric method, isogeometric collocation is based on the evaluation of the strong formulation of the PDE at a set of discrete points called *collocation points*. This method requires basis functions that are sufficiently smooth to handle possibly high order differential operators. In this work we use B-spline (or NURBS) basis functions as basis functions for the method. Let us consider now the ICM for Poisson's PDE.

### 5.1 SOLVING POISSON'S PDE

Let us consider the next Poisson's PDE:

$$\begin{cases} \Delta u = f \text{ on } \Omega \\ u = g \text{ on } \partial\Omega \end{cases}$$

where  $\Omega \subseteq \mathbb{R}^d$ ,  $d = 1, 2$ , is the physical domain,  $u : \Omega \rightarrow \mathbb{R}$  is the solution, while  $f : \Omega \rightarrow \mathbb{R}$  and  $g : \partial\Omega \rightarrow \mathbb{R}$  are a given sufficiently regular functions. We assume that the problem has a unique solution  $u$ . The collocation method used to solve this problem is based on the choice of a finite set of collocation points and the choice of them directly influence the stability and convergence properties of the collocation scheme. In general, collocation points are defined in the parametric space  $\hat{\Omega}$  from which the B-spline or NURBS geometry  $\Omega$  has been built. So, let  $\hat{C} := \{\hat{\tau}_i\}_{i \in I} \subset \hat{\Omega}$  and let  $\tau_i := \mathbf{G}(\hat{\tau}_i)$  for all  $i \in I$ , where  $\mathbf{G}$  is the geometry mapping  $\hat{\Omega} \rightarrow \Omega$ . We then define  $C := \{\tau_i\}_{i \in I} \subset \Omega$ . Let us separate  $C$  into two distinct sets:  $C_B$  corresponds to the set of collocation points belonging to the boundary  $\partial\Omega$ , and  $C_I$  corresponds to the set of collocation points belonging to the interior of  $\Omega$ . This is equivalent to separating  $\hat{C}$  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  $\Omega$  through  $\mathbf{G}$ . Then the collocation problem becomes

$$\begin{cases} \Delta u(\tau) = f(\tau), \text{ for each } \tau \in C_I \\ u(\tau) = g(\tau), \text{ for each } \tau \in C_B. \end{cases}$$

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 degree of the underlying B-splines, and let  $n$  be the dimension of  $\mathbb{B}_p$ . Moreover, let  $N_{i,p}$ ,  $i = 1, 2, \dots, n$  be the B-spline basis functions of  $\mathbb{B}_p$ . Then there exist  $c_i \in \mathbb{R}$ ,  $i = 1, 2, \dots, n$ , such that  $u = \sum_{i=1}^n c_i N_{i,p}$ . Consequently, the collocation problem is transformed into the following linear system whose unknowns are the coefficients  $c_i$ :

$$\begin{cases} \sum_{i=1}^n c_i \Delta N_{i,p}(\tau) = f(\tau), \text{ for each } \tau \in C_I, \\ \sum_{i=1}^n c_i N_{i,p}(\tau) = g(\tau), \text{ for each } \tau \in C_B. \end{cases} \quad (5.1)$$

The linear system for the one dimensional case can be written in a matrix form as  $\mathbf{A}\mathbf{c} = \mathbf{f}$ , where  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{c} \in \mathbb{R}^n$ ,  $\mathbf{f} \in \mathbb{R}^n$  and

$$\mathbf{A} = \begin{bmatrix} N_{1,p}(\tau_1) & N_{2,p}(\tau_1) & \cdots & N_{n,p}(\tau_1) \\ N''_{1,p}(\tau_2) & N''_{2,p}(\tau_2) & \cdots & N''_{n,p}(\tau_2) \\ \vdots & \vdots & \ddots & \vdots \\ N''_{1,p}(\tau_{n-1}) & N''_{2,p}(\tau_{n-1}) & \cdots & N''_{n,p}(\tau_{n-1}) \\ N_{1,p}(\tau_n) & N_{2,p}(\tau_n) & \cdots & N_{n,p}(\tau_n) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} \quad \text{and} \quad \mathbf{f} = \begin{bmatrix} g(\tau_1) \\ f(\tau_2) \\ \vdots \\ f(\tau_{n-1}) \\ g(\tau_n) \end{bmatrix}.$$

This formulation gives rise in general to a non-symmetric (but diagonally dominant) system matrix and if the choice of the collocation points leads to a well-posed problem, it can be easily solved to find the solution  $u \in \mathbb{B}_p$ .

If two dimensional case is considered, we need to use the geometry mapping  $\mathbf{G}$  to parametrize the domain.

**Lemma 5.1.** *Let  $\varphi_i : \hat{\Omega} \rightarrow \mathbb{R}$  and  $\psi_i : \Omega \rightarrow \mathbb{R}$  for  $i = 1, 2$  be two functions (as in Figure 3). Moreover, let  $J_{\mathbf{G}}(\boldsymbol{\xi})$  be the Jacobian of the geometry mapping  $\mathbf{G} : \hat{\Omega} \rightarrow \Omega$ . Then:*

$$\Delta \psi_i(\mathbf{x}) = \frac{1}{\sqrt{\det([J_{\mathbf{G}}(\boldsymbol{\xi})]^T J_{\mathbf{G}}(\boldsymbol{\xi}))}} \nabla(M(\boldsymbol{\xi}) \nabla \varphi_i(\boldsymbol{\xi})),$$

where

$$M(\boldsymbol{\xi}) = \sqrt{\det([J_{\mathbf{G}}(\boldsymbol{\xi})]^T J_{\mathbf{G}}(\boldsymbol{\xi}))} \cdot ([J_{\mathbf{G}}(\boldsymbol{\xi})]^T J_{\mathbf{G}}(\boldsymbol{\xi}))^{-1}.$$

The proof of Lemma 5.1 can be found in [3]. By Lemma 5.1, the linear system (5.1) in two dimensions becomes:

$$\begin{cases} \sum_{i=1}^n \sum_{j=1}^n \frac{c_{i,j}}{\sqrt{\det([J_{\mathbf{G}}(\boldsymbol{\tau})]^T J_{\mathbf{G}}(\boldsymbol{\tau}))}} \nabla(M(\boldsymbol{\tau}) \nabla(N_{i,p}(\boldsymbol{\tau}) N_{j,p}(\boldsymbol{\tau}))) = f(\boldsymbol{\tau}), \text{ for each } \boldsymbol{\tau} \in C_I, \\ \sum_{i=1}^n \sum_{j=1}^n c_{i,j} N_{i,p}(\boldsymbol{\tau}) N_{j,p}(\boldsymbol{\tau}) = g(\boldsymbol{\tau}), \text{ for each } \boldsymbol{\tau} \in C_B, \end{cases}$$

where  $\boldsymbol{\tau} = (\tau_1, \tau_2)$ . The solution  $u$  of a 2 dimensional Poisson's PDE using ICM is then sought in the space  $\mathbb{B}_{p,p}$ . Note that, in two dimensions, 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. Similarly as in the one dimensional case, the linear system can be written in a matrix form, where the matrix on the left side has dimensions  $n^2 \times n^2$ , while the vector of unknowns,  $\mathbf{c}$ , and the vector on the right side have dimension  $n^2$ .

## 5.2 GREVILLE COLLOCATION POINTS AND DEMKO COLLOCATION POINTS

The most widely used isogeometric collocation points are the Greville collocation points.

**Definition 5.2.** Let  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$  be a given knot vector as in the Def. 2.1 and let us consider univariate B-splines, then the *Greville collocation points (GCP)* are defined as the mean of  $p$  consecutive knots, that is:

$$\hat{\tau}_i := \frac{\xi_{i+1} + \xi_{i+2} + \dots + \xi_{i+p}}{p}, \quad \tau_i = \mathbf{G}(\hat{\tau}_i), \quad i = 1, 2, \dots, n.$$

If we use open knots,  $\xi_1 = \xi_2 = \dots = \xi_{p+1}$  and  $\xi_{n+1} = \xi_{n+2} = \dots = \xi_{n+p+1}$ , then it is easy to see that in this case,  $\hat{\tau}_1 = \xi_1$  and  $\hat{\tau}_n = \xi_{p+n+1}$ . Figure 5 shows the Greville collocation points for  $p = 3, \dots, 7$  on a knot vector with 8 elements. In the case of bivariate B-splines, Greville points are defined by means of a tensor product rule.

**Definition 5.3.** Let  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$  and  $H = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$  be two knot vectors, where  $p$  and  $q$  are polynomial degrees,  $n$  and  $m$  are the numbers of B-spline basis functions for  $\Xi$  and  $H$ , respectively. Then, the *Greville collocation points for a two dimensional space* are defined as  $\hat{\tau}_{ij} := (\hat{\xi}_i, \hat{\eta}_j)$ , where

$$\hat{\xi}_i := \frac{\xi_{i+1} + \xi_{i+2} + \dots + \xi_{i+p}}{p}, \quad \hat{\eta}_j := \frac{\eta_{j+1} + \eta_{j+2} + \dots + \eta_{j+q}}{q}.$$

Further we can map these points with the geometry mapping  $\mathbf{G} : \hat{\Omega} \rightarrow \Omega$  to get

$$\tau_{ij} = \mathbf{G}(\hat{\tau}_{ij}), \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m.$$

The procedure is the same if instead of taking B-splines basis functions, we take NURBS.

The other very widely used isogeometric collocation points are the *Demko collocation points (DCP)*. Those points correspond to the extrema of the Chebyshev splines, that are the splines whose extrema take the values  $\pm 1$  and that have the maximum number of oscillations. Demko points can be obtained thanks to Remez iterative algorithm. The Remez iterative algorithm is an efficient algorithm that constructs a unique minmax polynomial given an initial set of points. The minmax polynomial is such a

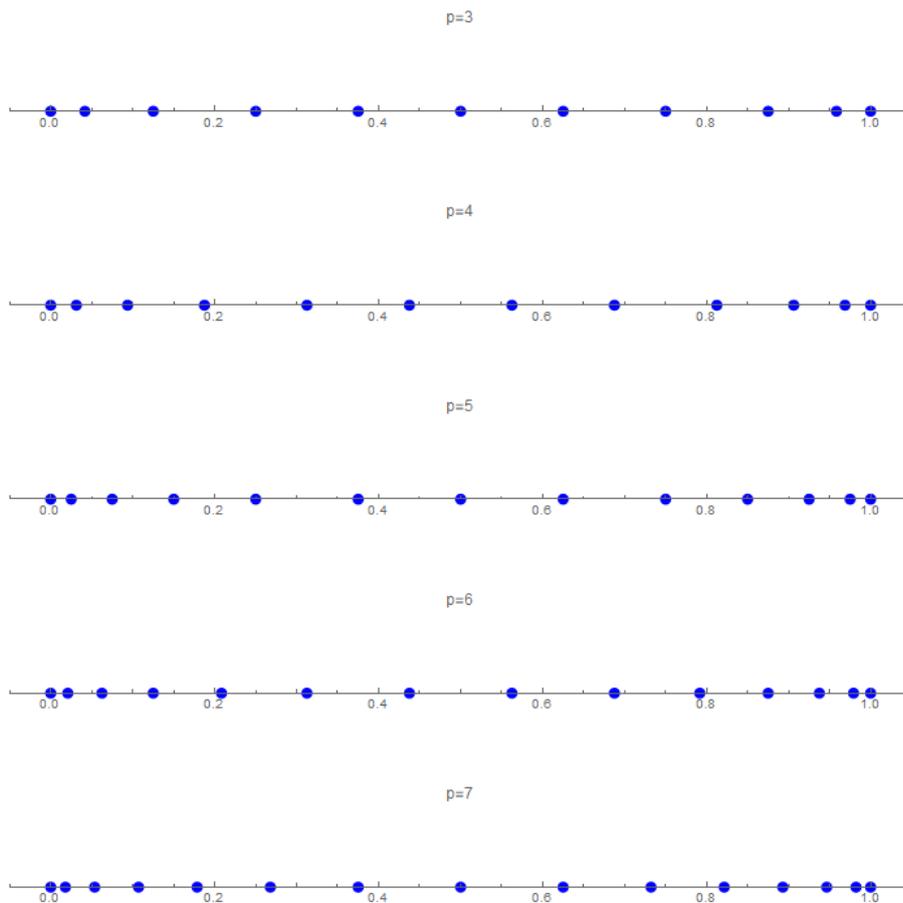


Figure 5: Distribution of GCP for  $p = 3, \dots, 7$ , on a knot vector with 8 elements.

polynomial that minimizes the maximum vertical distance between the polynomial and the function in consideration. Unlike the Greville points, there is no explicit formula for the Demko points, but it can be easily determined using the built in function `chbpnt` in MATLAB (see [8]). This built in function uses as initial guess the spline that takes alternatively the values 1 and  $-1$  at the sequence, which is composed of averages of successive  $p$  knots. Figure 6 shows the Demko collocation points for  $p = 3, \dots, 7$  on a knot vector with 8 elements. We refer the interested readers to [6, 10].

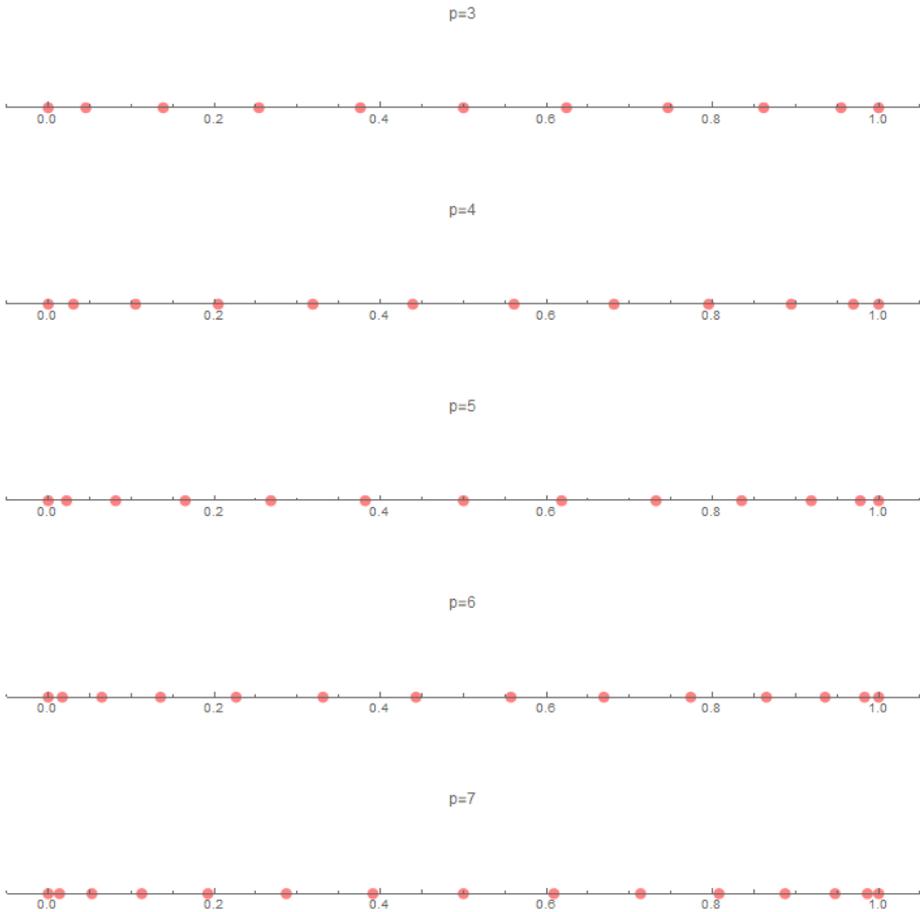


Figure 6: Distribution of DCP for  $p = 3, \dots, 7$ , on a knot vector with 8 elements.

## 6 SUPERCONVERGENT COLLOCATION POINTS

In this chapter, we discuss a choice of collocation points for which the solution obtained using isogeometric collocation method behaves similarly to the standard Galerkin isogeometric approximation. The main idea of this method is that we seek the collocation points for which the error in the second derivatives of Galerkin approximation is small. We will see that there are at least two superconvergent points per element, so if we take all of them, we obtain an overdetermined system of equations. This is why, we will, later on, consider the Alternating and Clustered superconvergent collocation points, which are subsets of superconvergent collocation points with cardinality equal to the number of B-spline basis functions ( $n$ ).

Let  $u_h$  be the approximated solution of a Poisson's PDE with Dirichlet boundary condition obtained by the Galerkin isogeometric method based on B-splines and let  $u$  be its exact solution.

The *superconvergent collocation points* are the zeros of the Galerkin residual, that is, the zeros of  $D^2(u - u_h)$ . In [11] it is proven that there exist at least  $n$  distinct superconvergent points for a space of dimension  $n$  and the collocation at those points produces the Galerkin solution exactly. We also refer the readers to [1], where it is investigated the use of the superconvergent points, in the second derivative, in collocation.

Since in practice we don't know the exact solution, and therefore neither the "exact" superconvergent points, we will use a method to find the "surrogate" superconvergent points (see [1, 11, 15]). We assume that the superconvergent points are element invariant, which means, that the points do not change their properties, if we map them with an affine mapping from a reference domain to another one. Their locations are presented in the Table 4 for the reference domain  $[-1, 1]$ . From now on, we will call the "surrogate" superconvergent points just as superconvergent points for simplicity, even if this is not really true.

## 6.1 METHOD FOR FINDING SUPERCONVERGENT POINTS

We will now see a general method (which is described in [11]) to find superconvergent points for B-splines of degree  $p$  and maximum continuity  $\mu = p - 1$ . In the next sections, we will assume that the knot vector is open and uniform. As before, let  $u$  be the exact solution of the Poisson's PDE,  $u_h$  the solution obtained using the GIM and define  $r = p + 1$ . Firstly, we need to perform the Taylor expansion of the error  $e = u - u_h$  up to order  $r$ . The method is based on the a priori assumption that the coefficients of the error expansion are the same in each knot span so that the expansion reads

$$e = c_0 P_0 + c_1 P_1 + \dots + c_r P_r + R \quad \text{in } [-1, 1] \quad (6.1)$$

with  $R = \mathcal{O}(h^{r+1})$  as the remainder. Here  $P_m$  denotes the Legendre polynomial of degree  $m$  on the normalized domain  $[-1, 1]$ , and  $c_m$  is the corresponding coefficient in the Taylor expansion, for  $m = 0, 1, 2, \dots, r$ . In this case  $h$  is a mesh length scale, moreover, the knot span length. Our goal is to define a set of equations that allows us to derive the coefficients of the Taylor expansion to eventually find the roots of  $e''$  on each knot span. So we need to find for a knot span the  $r + 1$  coefficients in the expansion up to a scaling factor. These conditions stem from

- the conditions on superconvergence of  $e$  (for even  $p$ ) or of  $e'$  (for odd  $p$ ) at knots and midpoints (see [21]) and
- the continuity conditions of  $e$  and its derivatives up to order  $\mu$  at the knots.

As follows, we will see how to find the coefficients  $c_0, c_1, c_2, \dots, c_r$  for the cases of even and odd  $p$ .

For multi-dimensional problems on a B-spline or NURBS single-patch geometry, the superconvergent points can be obtained by further mapping the tensor product of one-dimensional superconvergent points through the geometry map  $\mathbf{G}$  in the physical domain. Clearly, the same considerations of the one-dimensional case are valid.

### 6.1.1 Points for even $p$

For  $p = 2$ , it is known (see [21]) that superconvergent points for the second derivative are located at the midpoints of each knot span. These points coincide with Greville points. For  $p > 2$ , it is known (see [21]) that  $e$  is superconvergent at the knots and at the midpoint of each knot span. This can be written as  $e(-1) = e(0) = e(1) = \mathcal{O}(h^{r+1})$  and leads to the first set of equations:

$$e(1) + e(-1) = c_0 P_0(1) + c_2 P_2(1) + \dots + c_{r-1} P_{r-1}(x) = c_0 + c_2 + \dots + c_{r-1} = \mathcal{O}(h^{r+1}), \quad (6.2)$$

$$e(1) - e(-1) = c_1 P_1(1) + c_3 P_3(1) + \dots + c_r P_r(1) = c_1 + c_3 + \dots + c_r = \mathcal{O}(h^{r+1}), \quad (6.3)$$

$$e(0) = c_0 P_0(0) + c_2 P_2(0) + \dots + c_{r-1} P_{r-1}(0) = \mathcal{O}(h^{r+1}). \quad (6.4)$$

The second set of conditions is about the continuity. The continuity of  $e$  is already taken in consideration in Eq. (6.3), but we also need that the continuity holds for  $e'$ ,  $e''$  and all subsequent derivatives up to order  $\mu$ . At this point it is important to recall one of the identities regarding Legendre polynomials:  $P_m(-x) = (-1)^m P_m(x)$ . Using this identity we get the next continuity equations:

$$\begin{aligned} e'(1) - e'(-1) &= c_2 P_2'(1) + c_4 P_4'(1) + \dots + c_{r-1} P_{r-1}'(1) = \mathcal{O}(h^{r+1}) \\ e''(1) - e''(-1) &= c_3 P_3''(1) + c_5 P_5''(1) + \dots + c_r P_r''(1) = \mathcal{O}(h^{r+1}) \\ &\vdots \\ e^{(\mu)}(1) - e^{(\mu)}(-1) &= P_{r-1}^{(\mu)} c_{r-1} = \mathcal{O}(h^{r+1}) \Rightarrow c_{r-1} = \mathcal{O}(h^{r+1}). \end{aligned}$$

The last equation, in combination with continuity conditions of the odd derivatives  $e'$ ,  $e''$ , ...,  $e^{(\mu)}$  and with the Eq. (6.4), gives

$$c_0 = c_2 = \dots = c_{r-1} = \mathcal{O}(h^{r+1}),$$

which eliminates all terms with even index from the expansion Eq. (6.1) and the Eq. (6.2) becomes useless. Therefore, we are left with Eq. (6.3), and all the continuity conditions of the even derivatives  $e''$ ,  $e^{(4)}$ , ...,  $e^{(\mu-1)}$ . This means that we have  $1 + \frac{r-3}{2} = \frac{r-1}{2}$  equations and  $\frac{r+1}{2}$  unknowns, and so the number of unknowns is one less than the number of equations, as desired. The wanted coefficients can be then computed by solving:

$$\begin{bmatrix} P_1(1) & P_3(1) & P_5(1) & \cdots & P_{r-2}(1) & P_r(1) \\ 0 & P_3''(1) & P_5''(1) & \cdots & P_{r-2}''(1) & P_r''(1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & P_{r-2}^{(\mu-1)}(1) & P_r^{(\mu-1)}(1) \end{bmatrix} \begin{bmatrix} c_1 \\ c_3 \\ \vdots \\ c_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (6.5)$$

from which we get the coefficients  $c_3, c_5, \dots, c_r$  as functions of  $c_1$ . Moreover we obtain that

$$e(x) = c_1 (P_1(x) + k_1 P_3(x) + \dots + k_{\frac{r-1}{2}} P_r(x)) + \mathcal{O}(h^{r+1}), \quad (6.6)$$

where  $k_1, \dots, k_{\frac{r-1}{2}}$  are the coefficients of  $c_1$  obtained from the system (6.5). So now, we found the coefficients of the error function of a knot span. To standardize the location of superconvergent points, we need to map the Eq. (6.6) to the interval  $[-1, 1]$ , which we can do, since we assumed the element invariant property of superconvergent points. The parameter  $c_1$  can be simply thought of as a scaling factor which does not alter the

roots of  $e''$ , so we can eliminate it. If we now call  $e_R$ , the error function on the interval  $[-1, 1]$  we conclude that

$$e_R(\eta) \approx P_1(\eta) + k_1 P_3(\eta) + \dots + k_{\frac{r-1}{2}} P_r(\eta), \quad \eta \in [-1, 1]. \quad (6.7)$$

The last step is to calculate  $e''_R(\eta)$  and to solve  $e''_R(\eta) = 0$ . In this case we get three searching superconvergent points on the interval  $[-1, 1]$ . The pseudo-code of the algorithm for finding superconvergent points for  $p$  even is presented in **Algorithm 1**.

---

**Algorithm 1:** Superconvergent points for even  $p$  on  $[-1, 1]$ .

---

**Input:** Polynomial degree  $p$ .

**Output:** Vector  $s$  of superconvergent points on  $[-1, 1]$ .

```

1  $r := p + 1$ ;
2  $\mu := p - 1$ ;
3  $\mathbf{x} := \{x_i | i = 1, 2, \dots, \frac{p}{2} + 1\}$ ;
4 for  $j = 0, 2, 4, \dots, \mu - 1$ 
5   for  $i = 1, 3, 5, \dots, r$ 
6     //  $A =$  matrix on the right-hand side of dimension  $\frac{r+1}{2} \times \frac{r-1}{2}$ .
6      $A[i, j] :=$   $j$ -th derivative of  $P_i(x)$ ;
7  $c \leftarrow$  solution of  $A \cdot \mathbf{x} = 0$ ;
8  $e(x) = \sum_{i=1}^{\frac{r-1}{2}} c[i] \cdot P_{2i-1}(x)$ ;
9  $s \leftarrow$  solution of  $e''(x) = 0$ ;
10 return  $s$ ;
```

---

### 6.1.2 Points for odd $p$

For odd  $p$ , it is known (see [21]) that  $e'$  is superconvergent at the knots and at the midpoint of each knot span. This can be written as  $e'(-1) = e'(0) = e'(1) = \mathcal{O}(h^r)$  and leads to the first set of equations:

$$e'(1) + e'(-1) = c_1 + c_3 P'_3(1) + \dots + c_{r-1} P'_{r-1}(1) = \mathcal{O}(h^{r+1}) \quad (6.8)$$

$$e'(1) - e'(-1) = c_2 P'_2(1) + c_4 P'_4(1) + \dots + c_r P'_r(1) = \mathcal{O}(h^{r+1}) \quad (6.9)$$

$$e(0) = c_1 + c_3 P'_3(0) + \dots + c_{r-1} P'_{r-1}(0) = \mathcal{O}(h^{r+1}) \quad (6.10)$$

Also in this case, the second set of conditions is about the continuity. The continuity of  $e'$  is already taken in consideration in Eq. (6.9), but we also need that the continuity holds for  $e$ ,  $e''$  and all subsequent derivatives up to order  $\mu$ . Using the Legendre identity

that we have mentioned before, we get the next continuity equations:

$$\begin{aligned} e(1) - e(-1) &= c_1 + c_3 + \dots + c_{r-1} = \mathcal{O}(h^{r+1}) \\ e''(1) - e''(-1) &= c_3 P_3''(1) + c_5 P_5''(1) + \dots + c_r P_r''(1) = \mathcal{O}(h^{r+1}) \\ &\vdots \\ e^{(\mu)}(1) - e^{(\mu)}(-1) &= P_{r-1}^{(\mu)} c_{r-1} = \mathcal{O}(h^{r+1}) \Rightarrow c_{r-1} = \mathcal{O}(h^{r+1}). \end{aligned}$$

The last equation, in combination with continuity conditions of  $e$  and of its even derivatives  $e'', \dots, e^{(\mu-2)}$ , gives

$$c_1 = c_3 = \dots = c_{r-1} = \mathcal{O}(h^{r+1}),$$

which eliminates all terms with odd index from the expansion Eq. (6.1) and the Eqs. (6.8) and (6.10) become useless. Therefore, we are left with Eq. (6.9), and all the continuity conditions of the odd derivatives  $e''', e^{(5)}, \dots, e^{(\mu-1)}$ . This means that we have  $1 + \frac{r-4}{2} = \frac{r-2}{2}$  equations and  $\frac{r+2}{2}$  unknowns. In this case we obtain that the number of unknowns is two less than the number of equations. It can be shown that  $c_0 = 0$ . We omit the proof, because we are only interested in the derivatives of the error, for which the coefficient  $c_0$  does not play any role. The desired coefficients can be then computed, similarly as in the case for even  $p$ , by solving:

$$\begin{bmatrix} P_2'(1) & P_4'(1) & P_6'(1) & \dots & P_{r-2}'(1) & P_r'(1) \\ 0 & P_4'''(1) & P_6'''(1) & \dots & P_{r-2}'''(1) & P_r'''(1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & P_{r-2}^{(\mu-1)}(1) & P_r^{(\mu-1)}(1) \end{bmatrix} \begin{bmatrix} c_2 \\ c_4 \\ \vdots \\ c_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (6.11)$$

from which we get the coefficients  $c_4, c_6, \dots, c_r$  as functions of  $c_2$ . Therefore we obtain that

$$e(x) = c_2(P_2(x) + k_1 P_4(x) + \dots + k_{\frac{r-2}{2}} P_r(x)) + \mathcal{O}(h^{r+1}), \quad (6.12)$$

where  $k_1, \dots, k_{\frac{r-2}{2}}$  are the coefficients of  $c_2$  obtained from the system (6.11). So as before, we found the coefficients of the error function of a knot span. Again, to standardize the location of superconvergent points, we need to map the Eq. (6.12) to the interval  $[-1, 1]$ . So, in this case,  $e_R$ , the error function on the interval  $[-1, 1]$ , rescaled to eliminate the missing constant  $c_2$ , can be approximated as

$$e_R(\eta) \approx P_2(\eta) + k_1 P_4(\eta) + \dots + k_{\frac{r-2}{2}} P_r(\eta), \quad \eta \in [-1, 1]. \quad (6.13)$$

The last step is identical as the one for even  $p$ : we calculate  $e_R''(\eta)$  and solve  $e_R''(\eta) = 0$  so we get the two searching superconvergent points on the interval  $[-1, 1]$ . The pseudo-code of the algorithm for finding superconvergent points for  $p$  odd is presented in **Algorithm 2**.

The error functions and their second derivatives for even and odd  $p$  are presented in Table 3, while the zeros of  $e_R''$  are presented in Table 4.

---

**Algorithm 2:** Superconvergent points for odd  $p$  on  $[-1, 1]$ .

---

**Input:** Polynomial degree  $p$ .**Output:** Vector  $s$  of superconvergent points on  $[-1, 1]$ .

```

1  $r := p + 1$ ;
2  $\mu := p - 1$ ;
3  $\mathbf{x} := \{x_i | i = 1, 2, \dots, \frac{p+1}{2}\}$ ;
4 for  $j = 1, 3, 5, \dots, \mu - 1$ 
5   for  $i = 2, 4, 6, \dots, r$ 
6     //  $A =$  matrix on the right-hand side of dimension  $\frac{r}{2} \times \frac{r-2}{2}$ .
7      $A[i, j] :=$   $j$ -th derivative of  $P_i(1)$ ;
7  $c \leftarrow$  solution of  $A \cdot x = 0$ ;
8  $e(x) = \sum_{i=1}^{\frac{r-2}{2}} c[i] \cdot P_{2i-2}(x)$ ;
9  $s \leftarrow$  solution of  $e''(x) = 0$ ;
10 return  $s$ ;
```

---

Table 3: Error functions and their second derivative for different degrees.

$p$	$e_R(\eta)$	$e''_R(\eta)$
2	$-\frac{5}{2}\eta(\eta^2 - 1)$	$-15\eta$
3	$-\frac{7}{80}(15\eta^4 - 30\eta^2 + 7)$	$\frac{21}{4}(1 - 3\eta^2)$
4	$\frac{7}{16}\eta(3\eta^4 - 10\eta^2 + 7)$	$\frac{105}{4}\eta(\eta^2 - 1)$
5	$\frac{1}{48}(21\eta^6 - 105\eta^4 + 147\eta^2 - 31)$	$\frac{7}{8}(15\eta^4 - 30\eta^2 + 7)$
6	$-\frac{5}{48}\eta(3\eta^6 - 21\eta^4 + 49\eta^2 - 31)$	$-\frac{35}{8}\eta(3\eta^4 - 10\eta^2 + 1)$
7	$-\frac{33}{6400}(15\eta^8 - 140\eta^6 + 490\eta^4 - 620\eta^2 + 127)$	$-\frac{33}{160}(21\eta^6 - 105\eta^4 + 147\eta^2 - 31)$

Table 4: Superconvergent points on interval  $[-1, 1]$  for different degrees.

$p$	zeros of $e''$
2	0
3	$\pm \frac{1}{\sqrt{3}}$
4	$-1, 0, 1$
5	$\pm 0.51932962$
6	$-1, 0, 1$
7	$\pm 0.50491857$

## 6.2 LEAST-SQUARES AT SUPERCONVERGENT COLLOCATION POINTS (LS-SP)

As mentioned above, with the just described method, we get approximately twice as many SCP points as needed to define a collocation scheme. Let the superconvergent points be denoted as  $\{\psi_i\}_{i \in I}$ . In details, let  $\Xi = \{\xi_1, \dots, \xi_{n+p+1}\}$  be a knot vector with notation as in Def. 2.1, then we have found  $|I| = 2(n - p)$  SCP for odd polynomial degree  $p$  and  $|I| = 3(n - p)$  SCP for even polynomial degree  $p$ . Note that we get enough SCP only if  $|I| \geq n$ , i.e. it must hold  $n \geq 2p$  for  $p$  odd and  $2n \geq 3p$  for  $p$  even. Figure 7 shows the SCP for  $p = 3, 4, \dots, 7$  on a knot vector with 8 elements.

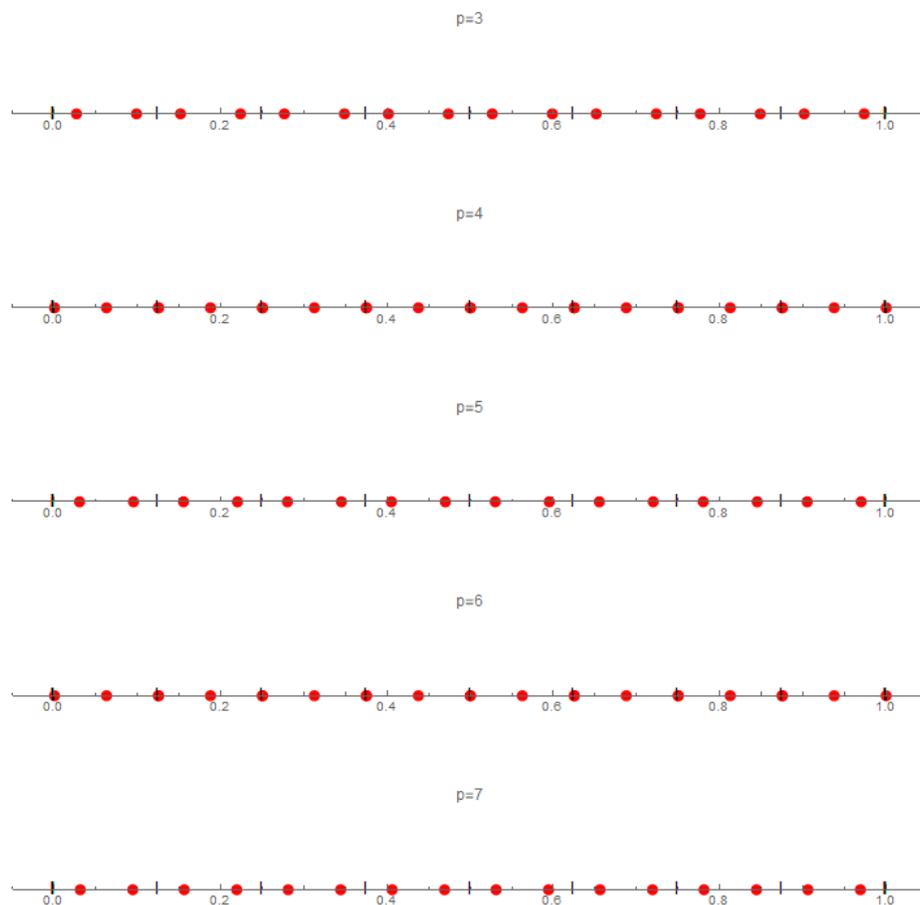


Figure 7: Distribution of SCP for  $p = 3, \dots, 7$ , on a knot vector with 8 elements.

If we take all of them, we get an overdetermined system of equations, which we can solve in a least squares sense, leading to a method which is not strictly a collocation method, but we mention it as a sake of completeness. We refer the interested readers to [12]. Recall from Chapter 5.1, that the isogeometric collocation problem in one

dimension, can be written in a matrix form as  $\mathbf{A}\mathbf{c} = \mathbf{f}$ . In this case we will have a non-square matrix  $\mathbf{A}$  of dimensions  $m \times n$  and a longer vector  $\mathbf{f}$  of dimension  $m$ , where  $m = 2(n - p)$  if  $p$  is odd and  $m = 3(n - p)$  if  $p$  is even. To obtain the solution by the *method of least squares*, we firstly need to multiply both sides by  $\mathbf{A}^T$  and then solve the system:

$$\mathbf{A}^T \mathbf{A} \mathbf{c} = \mathbf{A}^T \mathbf{f} \quad \longrightarrow \quad \mathbf{c} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{f}$$

to get the desired coefficients  $c_i$  for  $i = 1, 2, \dots, n$ . Similarly we can do for the two dimensional case.

### 6.3 ALTERNATING SUPERCONVERGENT COLLOCATION POINTS

In [11], it's observed that any subset of SCP will produce the Galerkin solution, provided that there is at least one point on the support of each B-spline or NURBS basis functions. Now we need to select as many collocation points as B-spline basis functions, i.e.  $n$ .

The Alternating Superconvergent collocation points (ASCP) is a selected subset of the superconvergent points, say  $\{\tau_i\}_{i=1}^n$ . This set of points is selected in such a way that every element  $[\xi_i, \xi_{i+1})$  of the knot span contains at least one superconvergent point. This roughly means to consider every second superconvergent point, hence the name of the method. We want to impose Dirichlet boundary conditions strongly; therefore we define  $\tau_1 = 0$  and  $\tau_n = 1$ . To find the internal points,  $\{\tau_i\}_{i=2}^{n-1}$ , we consider separately the case when  $p$  is odd and  $p$  is even. For  $p$  is odd we need to separate the cases when  $n$  is odd and  $n$  is even. When  $n$  is odd, we delete every second SCP  $\psi_i$  starting from  $i = 2$  up to  $n - p$  and then, to get a symmetric stencil, we delete every second point from  $2(n - p) - 1$  back, up to  $n - p + 3$ . While, when  $n$  is even, we delete every second SCP  $\psi_i$  starting from  $i = 2$  up to  $n - p$ , as before, and then every second point from  $2(n - p) + 1$  back, up to  $n - p + 3$ . By deleting these points, we obtain  $\frac{2(n-p)}{2} + 1 = n - p + 1$  points, which is less than  $n - 2$  if  $p > 3$ . Therefore, for  $p > 3$ , we need to add  $x$  points, where the number  $x$  is found by solving  $n - p + 1 + x = n - 2$ . These  $p - 3$  points are added close to the boundary of our domain. For  $p$  is even, as in [11], we take the Greville points, since there is no subset that gives better order of convergence. The algorithm for finding interior ASCP is presented in **Algorithm 3**. The ASCP for odd  $p$  and different  $n$  are presented in Figures 8 and 9.

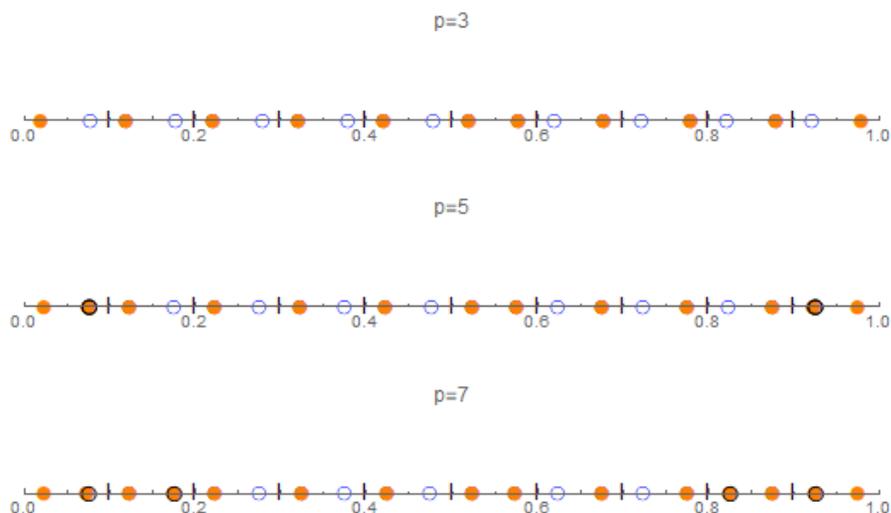


Figure 8: Distribution of internal ASCP for odd  $p = 3, 5, 7$  with a given knot vector having 10 elements and  $n = 13$ : the ASCP are in orange, the remaining superconvergent points are displayed with blue circles, while the black rounded orange points are the “additional points”.

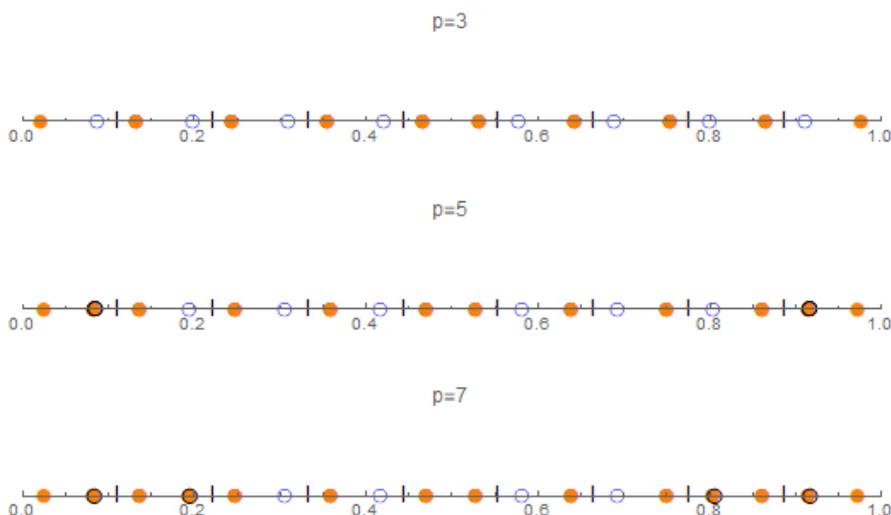


Figure 9: Distribution of internal ASCP for odd  $p = 3, 5, 7$  with a given knot vector having 9 elements and  $n = 12$ : the ASCP are in orange, the remaining superconvergent points are displayed with blue circles, while the black rounded orange points are the “additional points”.

**Algorithm 3:** Alternating superconvergent collocation points.

---

**Input:** Vector of SCP ( $s$ ), number of B-spline basis functions ( $n$ ) and polynomial degree ( $p$ ).

**Output:** Vector of ASCP ( $a$ ).

```

1 if  $p$  is odd then
2   if  $n$  is odd then
3     for  $i = 2, 4, 6, \dots, n - p$ 
4        $\lfloor$  delete  $s[i]$  from  $s$ ;
5     for  $i = 2(n - p) - 1, 2(n - p) - 3, \dots, n - p + 3$ 
6        $\lfloor$  delete  $s[i]$  from  $s$ ;
7   else
8     for  $i = 2, 4, 6, \dots, n - p$ 
9        $\lfloor$  delete  $s[i]$  from  $s$ ;
10    for  $i = 2(n - p) + 1, 2(n - p) - 1, \dots, n - p + 3$ 
11       $\lfloor$  delete  $s[i]$  from  $s$ ;
12     $a \leftarrow s$ ;
13    for  $i = 2, 4, 6, \dots, p - 3$ 
14       $\lfloor$  add  $s[q]$  and  $s[2(n - p) - q + 1]$  to  $a$ ;
15    add 0 and 1 at the beginning and at the end of  $a$ , respectively ;
16 else
17   for  $i = 1, 2, \dots, n$ 
18      $\lfloor$   $a[i] = \frac{\sum_{j=i+1}^{i+p} \xi_j}{p}$ ;
19 return  $a$ ;
```

---

## 6.4 CLUSTERED SUPERCONVERGENT COLLOCATION POINTS

Another choice of collocation points among the superconvergent points, alternative to ASCP, are the Clustered Superconvergent collocation points (CSCP).

In this case, the idea is to select two superconvergent points in an element and then skip the following one. As for the ASCP, we want to impose Dirichlet boundary conditions strongly, so again, we define  $\tau_1 = 0$  and  $\tau_n = 1$ . To find the internal points,  $\{\tau_i\}_{i=2}^{n-1}$ , we again consider separately the case when  $p$  is odd and  $p$  is even. For  $p$  is odd (see Figure 10), we delete every fourth SCP  $\{\psi_i, \psi_{i+1}\}$  starting from  $i = 3$  up to  $2(n - p) - 1$ . By deleting these points, we obtain  $\frac{2(n-p)}{2} = n - p$  points, which is less

than  $n - 2$  for  $p > 2$ . Similarly as for ASCP, by solving  $n - p + x = n - 2$ , we get that we need to add  $p - 2$  points close to the boundary of the chosen interval (which in Figure 10, are referred as the “additional points”). Again, from [11] we know that there is no subset that gives better order of convergence as that of Greville points. Therefore, for  $p$  is even we choose the Greville points. The algorithm for finding interior CSCP is presented in Algorithm 4.

---

**Algorithm 4:** Clustered superconvergent collocation points.
 

---

**Input:** Vector of SCP ( $s$ ), number of B-spline basis functions ( $n$ ) and polynomial degree ( $p$ ).

**Output:** Vector of CSCP ( $c$ ).

```

1  $q := 0$ ;
2 if  $p$  is odd then
3   for  $i = 3, 7, \dots, 2(n - p) - 1$ 
4      $\lfloor$  delete  $s[i]$  and  $s[i + 1]$  from  $s$ ;
5    $c \leftarrow s$ ;
6   while  $\text{length}(c) \neq n - 2$  do
7      $\lfloor$  add  $s[2(n - p) - q]$  and  $s[q + 1]$  to  $c$ ;
8      $q ++$ ;
9   add 0 and 1 at the beginning and at the end of  $c$ , respectively ;
10 else
11   for  $i = 1, 2, \dots, n$ 
12      $\lfloor$   $c[i] = \frac{\sum_{j=i+1}^{i+p} \xi_j}{p}$ ;
13 return  $c$ ;
```

---

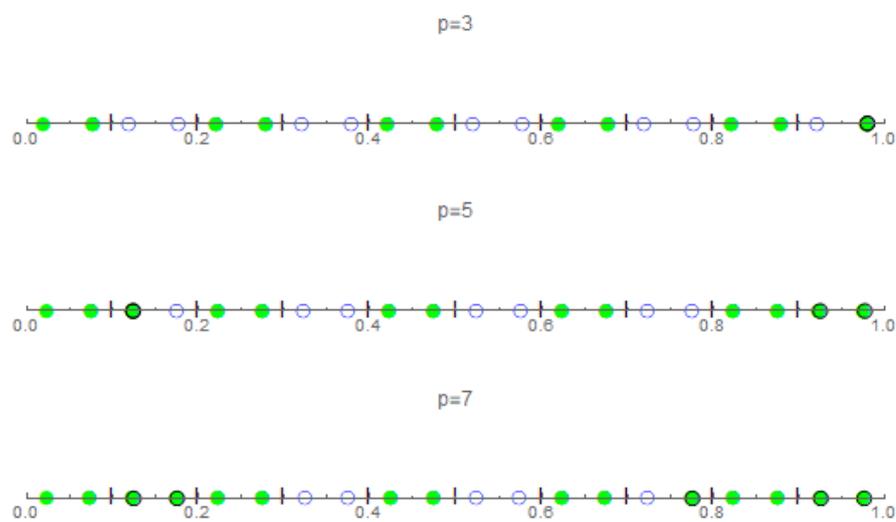


Figure 10: Distribution of internal CSCP for odd  $p = 3, 5, 7$  with a given knot vector having 10 elements: the CSCP are in green, the remaining superconvergent points are displayed with blue circles, while the black rounded green points are the “additional points”.

## 7 NUMERICAL TESTS

This section is devoted to the numerical tests of the ICM for Poisson's PDE with Dirichlet boundary conditions in one and two dimensions for different collocation points  $\{\tau_i\}_{i \in I}$ . With the numerical tests we observe the rate of convergence depicted with graphs.

As till now, let  $u$  be the exact solution of the Poisson's PDE and let  $u_h^*$  be the approximated solution obtained with the ICM. The *relative error* is then calculated as

$$e_X^h = \frac{\|u - u_h^*\|_X}{\|u\|_X},$$

where  $X$  represent the normed space and  $h$  is the length of the knot span of the considered uniform and open knot vector  $\Xi$ . In the next, we calculate relative errors considering  $L^2$ ,  $H_1$  and  $H_2$  norms. Each graph has on the  $y$ -axis  $\log_2(e_X^h)/\log_2(10)$  and on the  $x$ -axis the levels. The level represents the number of elements of  $\Xi$  and each time we do knot insertion refinement (in particular, we double the elements), we increase the level. More precisely, the set of levels is  $\{2^i \cdot h_0\}_{i=0}^3$ , where  $h_0 = \frac{1}{h}$  and in the graphs are presented with number 1, 2, 3 and 4, respectively. In the graphs we draw also the reference lines which helps us to see the rate of convergence. To find the rate of convergence, we have to compute the relative errors and they can be written as

$$e_X^h = Ch^r + O(h^{r+1}), \quad e_X^{\frac{h}{2}} = C \left(\frac{h}{2}\right)^r + O\left(\left(\frac{h}{2}\right)^{r+1}\right), \quad \dots$$

where  $C \in \mathbb{R}$  is a constant and  $h$  represent the initial knot span, i.e.  $h = \frac{1}{h_0}$ . The rate of convergence for the initial  $h$  is then

$$r_h = \log_2 \left( \frac{e_X^h}{e_X^{\frac{h}{2}}} \right)$$

and similarly we can compute also  $r_{\frac{h}{2}}, r_{\frac{h}{4}}, \dots$ . We expect that this sequence converges to a number, which we denote by  $r$ . The reference line is then defined as  $l(h) = h^{-r}$ , where  $r$  is the rate of convergence.

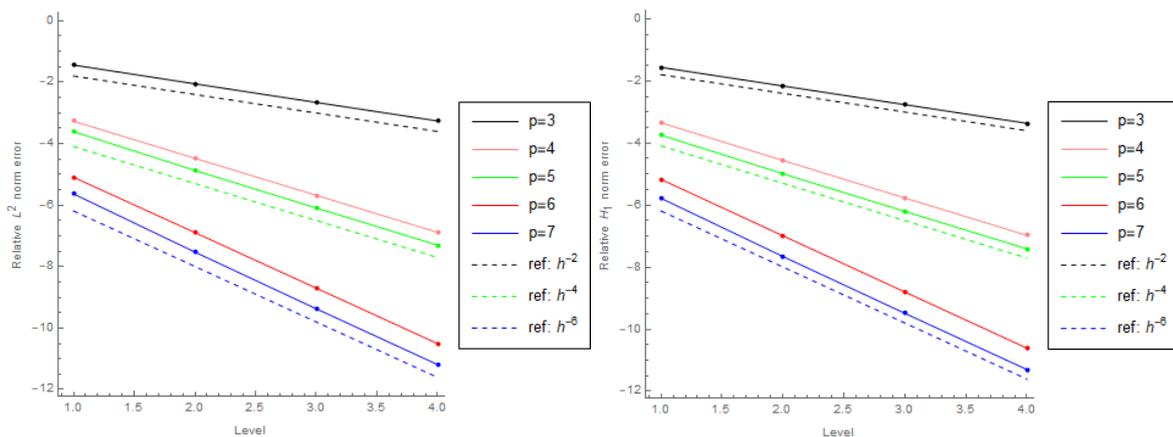
**Example 7.1.** For the first example, let  $\Omega = \hat{\Omega} = [0, 1]$  and let us consider the one dimensional Poissons's PDE:

$$\begin{cases} u''(x) = -\frac{9}{2}\pi^2 \cos(3\pi x), & \text{for } x \in (0, 1), \\ u(0) = \frac{1}{2}, & u(1) = -\frac{1}{2}, \end{cases}$$

whose exact solution is  $u(x) = \frac{1}{2} \cos(3\pi x)$ . In this example we take  $h = \frac{1}{16}$  and so  $h_0 = 16$ .

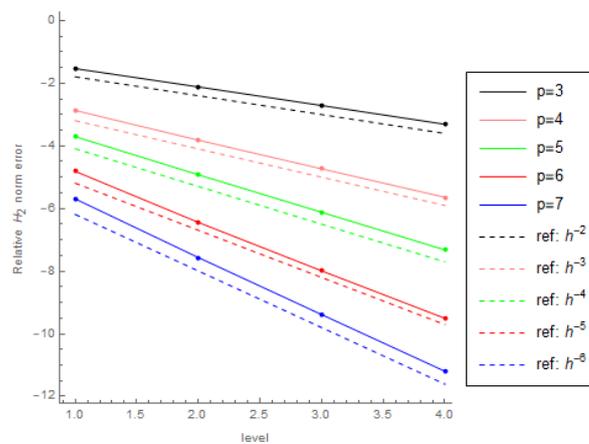
In Figure 11 we observe the rate of convergence of ICM using GCP:

- the rate of convergence in  $L^2$  norm is  $p - 1$  for odd polynomial degree and  $p$  for even polynomial degree (see Figure 11a),
- the rate of convergence is the same in  $H_1$  norm as in  $L_2$  norm (see Figure 11b) and
- the rate of convergence decreases in  $H_2$  norm for even polynomial degree, indeed, regardless of parity, it is  $p - 1$  (see Figure 11c).



(a) Relative error in  $L^2$  norm.

(b) Relative error in  $H_1$  norm.



(c) Relative error in  $H_2$  norm.

Figure 11: 1D relative errors for GCP.

In Figure 12 we can see the rate of convergence of ICM using DCP. In this case the errors converge with the same orders as for the Greville points:

- in  $L^2$  and  $H_1$  norm the rate of convergence is  $p - 1$  for  $p$  odd and  $p$  for  $p$  even (see Figure 12a, 12b) and
- in  $H_2$  norm the rate of convergence is  $p - 1$  (see Figure 12c).

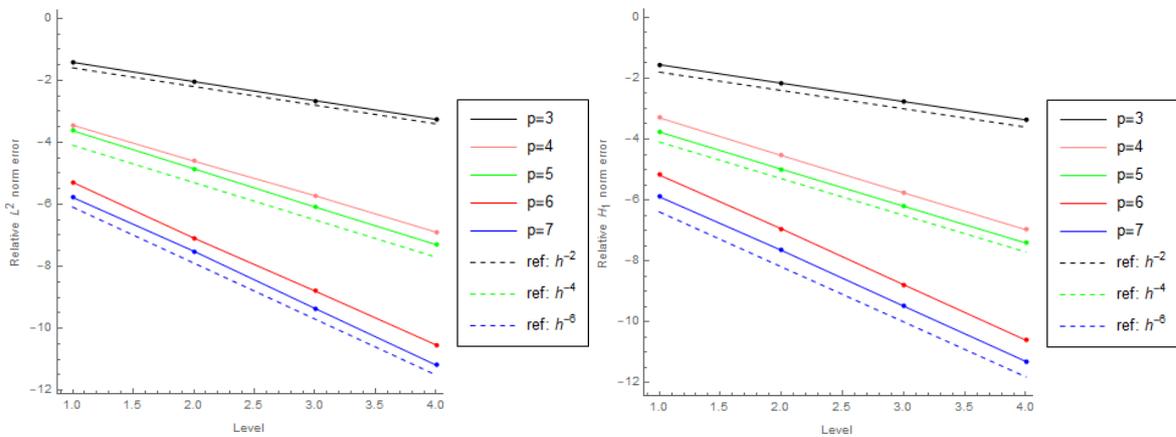
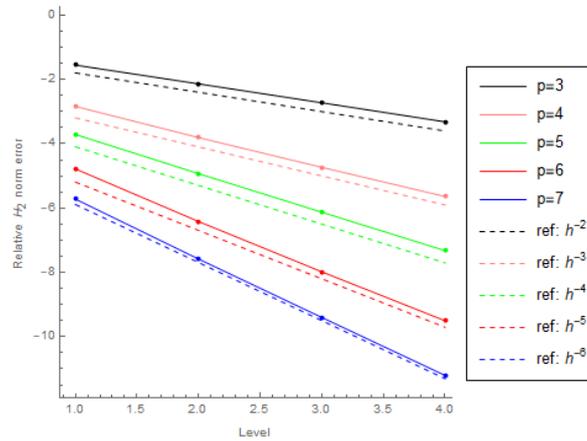
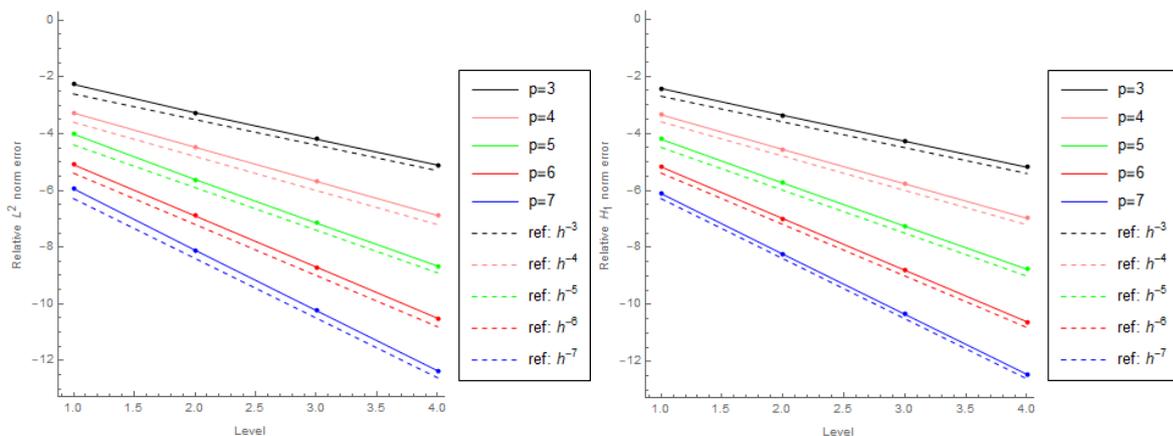
(a) Relative error in  $L^2$  norm.(b) Relative error in  $H_1$  norm.(c) Relative error in  $H_2$  norm.

Figure 12: 1D relative errors for DCP.

The convergence orders of ASCP are shown in Figure 13:

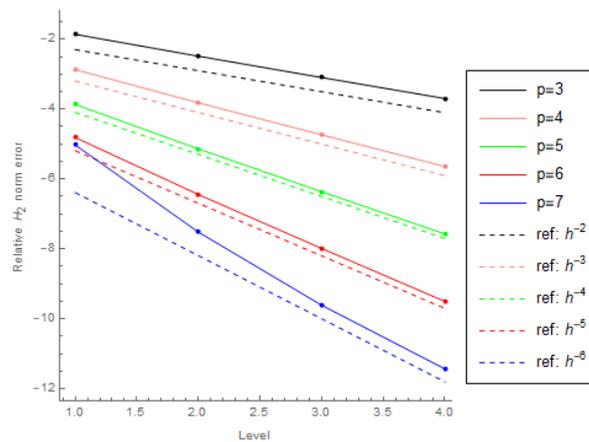
- in  $L^2$  and  $H_1$  norm the rate of convergence is  $p$  (see Figures 13a and 13b), regardless of the parity of  $p$ ,
- in  $H_2$  norm the rate is  $p - 1$ , again regardless of the parity of  $p$  (see Figure 13c).

Therefore we can observe that with ASCP we get a better approximation for odd  $p$  than using GCP or DCP.



(a) Relative error in  $L^2$  norm.

(b) Relative error in  $H_1$  norm.



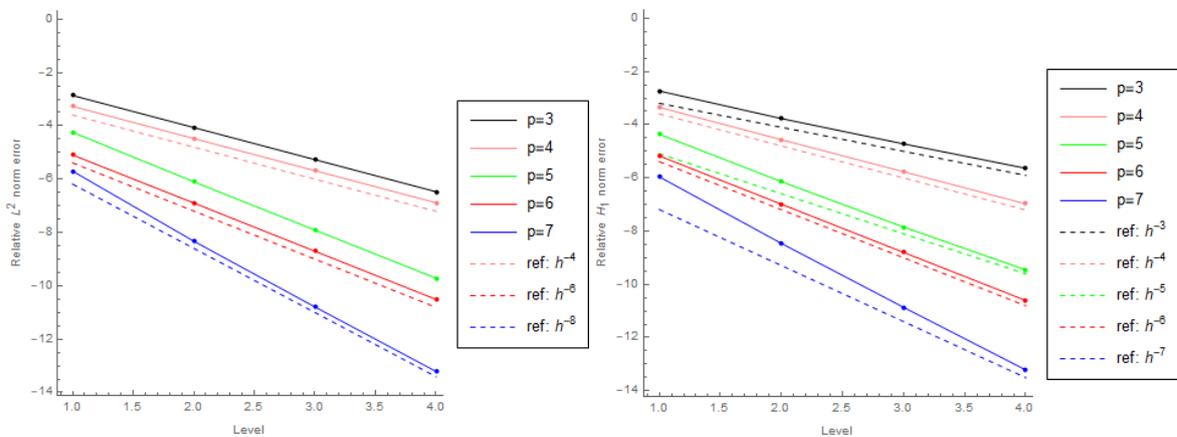
(c) Relative error in  $H_2$  norm.

Figure 13: 1D relative errors for ASCP.

In Figure 14 the rate of convergence for CSCP using ICM is illustrated:

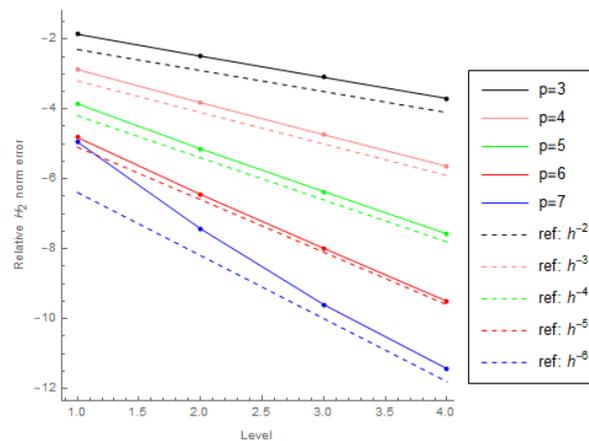
- in  $L^2$  norm the rate of convergence is  $p + 1$  for odd polynomial degree and  $p$  for even polynomial degree (see Figure 14a),
- in  $H_1$  norm the rate of convergence is  $p$  (see Figure 14b) and
- in  $H_2$  norm the rate of convergence is  $p - 1$  (see Figure 14c).

With CSCP the rate of convergence for odd  $p$  is even better in comparison with the ASCP, GCP and DCP.



(a) Relative error in  $L^2$  norm.

(b) Relative error in  $H_1$  norm.



(c) Relative error in  $H_2$  norm.

Figure 14: 1D relative errors for CSCP.

In Figure 15 we can observe the comparison between different collocation points and the Galerkin isogeometric method when  $p = 3$ . Clearly, our aim is to achieve the rate of convergence of the Galerkin isogeometric method. Since DCP result in a very slightly differences in accuracy compared to GCP, we omit them in the graphs. We can summarize the results from another point of view:

- in  $L^2$  norm and  $p$  odd, the convergence rate is optimal for CSCP, for ASCP is one-order suboptimal, and it is two-orders suboptimal for GCP (and DCP). For even degree, in all types of collocation points we get one-order suboptimal convergence;
- in  $H_1$  norm the convergence rate is optimal for all collocation points except for GCP (and DCP), when the polynomial degree is odd;
- in  $H_2$  norm all mentioned collocation points gives the optimal rate of convergence.

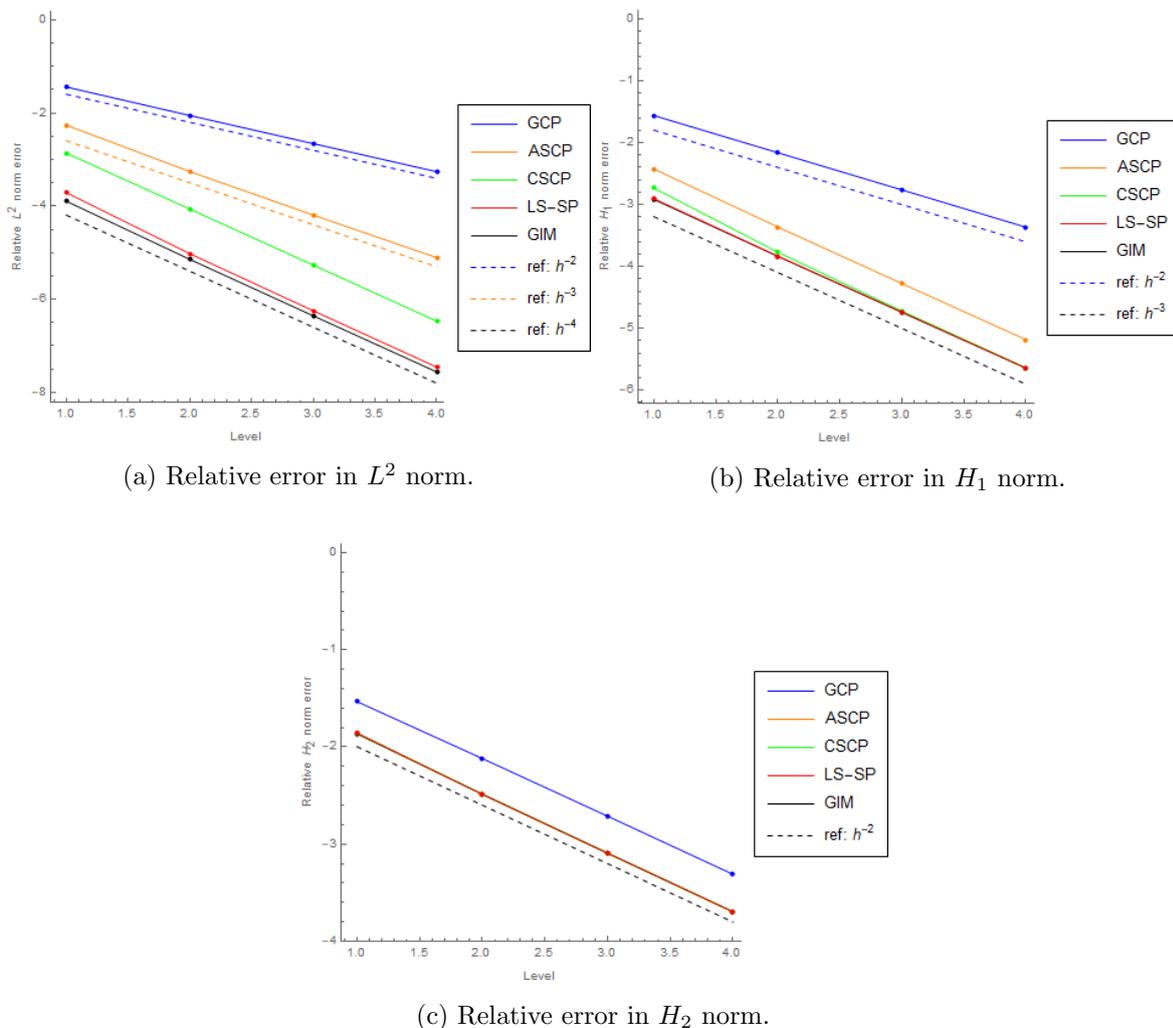


Figure 15: Comparison of relative errors for different methods and norms when  $p = 3$ .

**Example 7.2.** For the second example, let  $\Omega = \hat{\Omega} = [0, 1]^2$  and let us consider the two dimensional homogeneous Poissons's PDE:

$$\begin{cases} \Delta u(x_1, x_2) = -\pi^2 \sin(\pi x_1) \cdot \sin(\pi x_2), (x_1, x_2) \in \Omega \\ u(x_1, x_2) = 0, (x_1, x_2) \in \partial\Omega, \end{cases}$$

whose exact solution is  $u(x_1, x_2) = \frac{1}{2} \sin(\pi x_1) \sin(\pi x_2)$ . In this example we take  $h_0 = 4$  and so  $h = \frac{1}{4}$ . Also, the geometry mapping  $\mathbf{G}$  remains for now the identity. Note that to find the B-spline basis functions, we do the tensor product of the chosen knot vector, i.e.  $\Xi \times \Xi$ . Similarly, we do also for collocation points, that is to say, that we have the same collocation points in both directions.

In this case we calculated the relative errors for  $p = 3, 4, 5$ , because the time complexity of the algorithm is much bigger. In Figures 16, 17, 18 and 19 the rates of convergence for GCP, DCP, ASCP and CSCP, are shown respectively. It is possible to observe that the same orders of convergence in the  $L^2$ ,  $H_1$  and  $H_2$  norms expected in 1D case are attained also in 2D case.

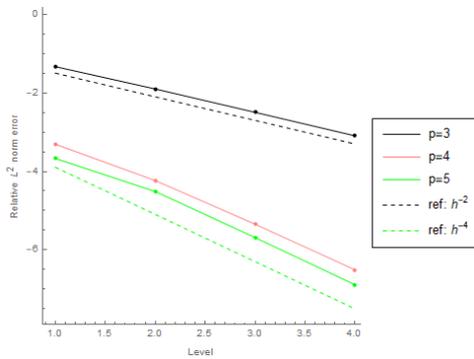
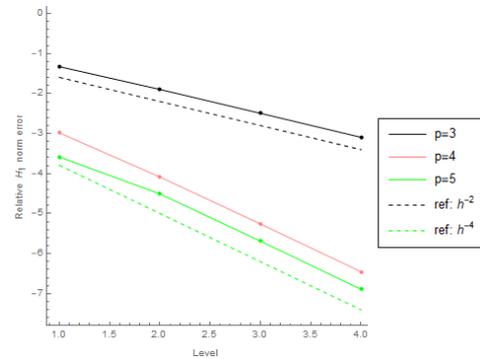
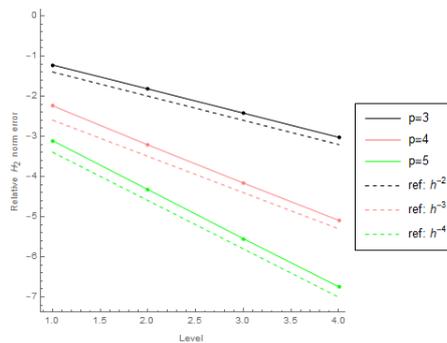
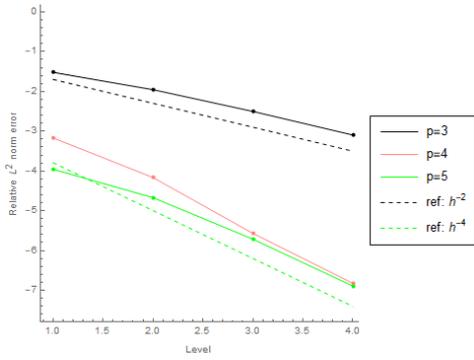
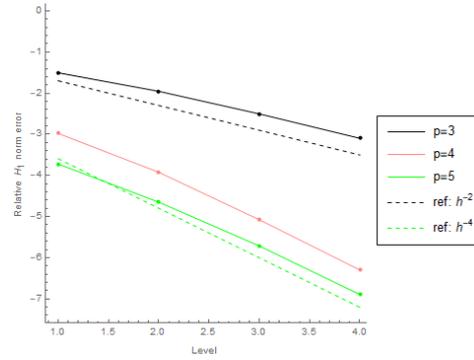
(a) Relative error in  $L^2$  norm.(b) Relative error in  $H_1$  norm.(c) Relative error in  $H_2$  norm.

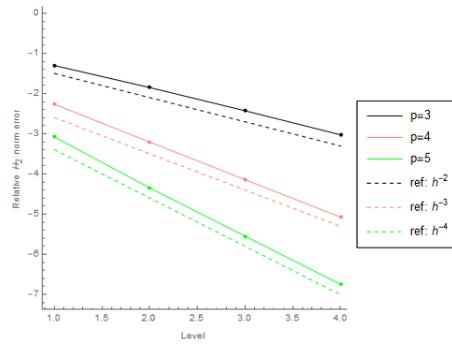
Figure 16: 2D relative errors for GCP.



(a) Relative error in  $L^2$  norm.

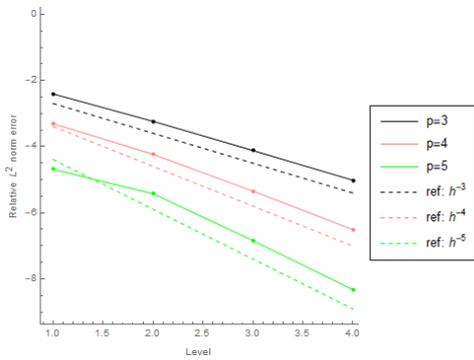


(b) Relative error in  $H_1$  norm.

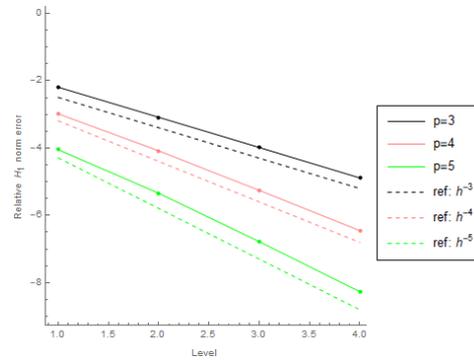


(c) Relative error in  $H_2$  norm.

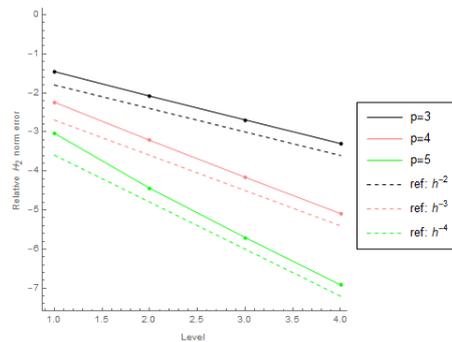
Figure 17: 2D relative errors for DCP.



(a) Relative error in  $L^2$  norm.



(b) Relative error in  $H_1$  norm.



(c) Relative error in  $H_2$  norm.

Figure 18: 2D relative errors for ASCP.

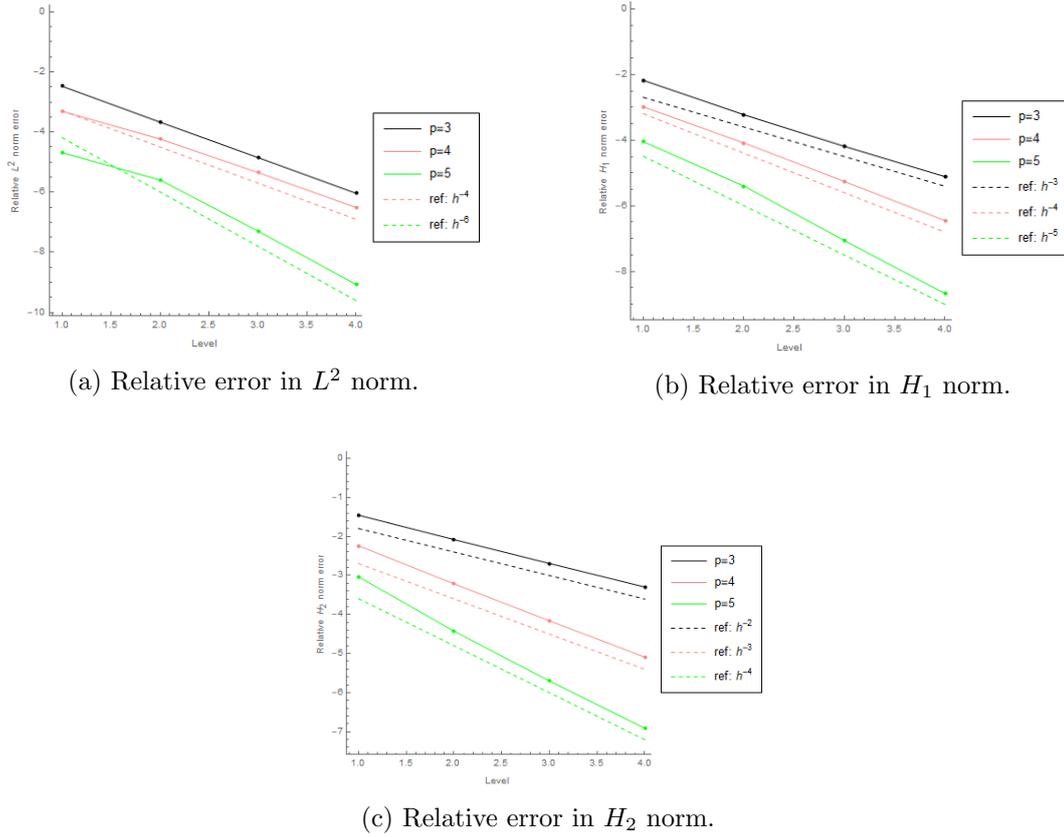


Figure 19: 2D relative errors for CSCP.

**Example 7.3.** Let  $\Omega$  be a domain different from the parametric one and let the geometry mapping be the so called bilinear mapping. Moreover, let  $\Omega$  be the rhombus with vertices  $(0, 0)$ ,  $(1/3, 1)$ ,  $(1, 1/2)$  and  $(3/2, 3/2)$ . The geometry mapping is defined using Bernstein basis polynomials. Recall that the Bernstein basis polynomials are of the form:

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i},$$

and so the tensor product of two Bernstein basis polynomials having the same degree  $n$ , is defined as:

$$B_{i,j}^n(x_1, x_2) = \binom{n}{i} x_1^i (1-x_1)^{n-i} \cdot \binom{n}{j} x_2^j (1-x_2)^{n-j}.$$

The geometry map,  $\mathbf{G} : [0, 1]^2 \rightarrow \Omega$ , used in this example is then defined as:

$$\mathbf{G}(\xi_1, \xi_2) = \sum_{i,j=0}^1 \mathbf{B}_{i,j} \cdot B_{i,j}^1(\xi_1, \xi_2),$$

where  $\mathbf{B}_{i,j}$  are the control points, i.e. the vertices of the rhombus:  $\mathbf{B}_{0,0} = (0, 0)$ ,  $\mathbf{B}_{0,1} = (1/3, 1)$ ,  $\mathbf{B}_{1,0} = (1, 1/2)$  and  $\mathbf{B}_{1,1} = (3/2, 3/2)$ . Let us analyse the next two-

dimensional Poisson's PDE:

$$\begin{cases} \Delta u(x_1, x_2) = -\pi^2 \sin(\pi x_1) \cdot \cos(\pi x_2), (x_1, x_2) \in \Omega \\ u(x_1, x_2) = \begin{cases} \frac{1}{2} \cos(3\pi x_1) \sin(\pi x_1) & \text{for } x_2 = 3x_1, x_1 \in [0, \frac{1}{3}] \\ \frac{1}{2} \cos(\frac{3}{7}\pi(2 + x_1)) \sin(\pi x_1) & \text{for } x_2 = \frac{3}{7}x_1 + \frac{6}{7}, x_1 \in [\frac{1}{3}, \frac{3}{2}] \\ -\frac{1}{2} \cos(2\pi x_1) \sin(\pi x_1) & \text{for } x_2 = 2x_1 - 3, x_1 \in [1, \frac{3}{2}] \\ \frac{1}{2} \cos(\frac{\pi x_1}{2}) \sin(\pi x_1) & \text{for } x_2 = \frac{1}{2}x_1, x_1 \in [0, 1] \end{cases}, (x_1, x_2) \in \partial\Omega, \end{cases}$$

whose exact solution is  $u(x_1, x_2) = \frac{1}{2} \cdot \cos(\pi x_2) \cdot \sin(\pi x_1)$ . The physical domain with the exact solution can be seen in Figure 20.

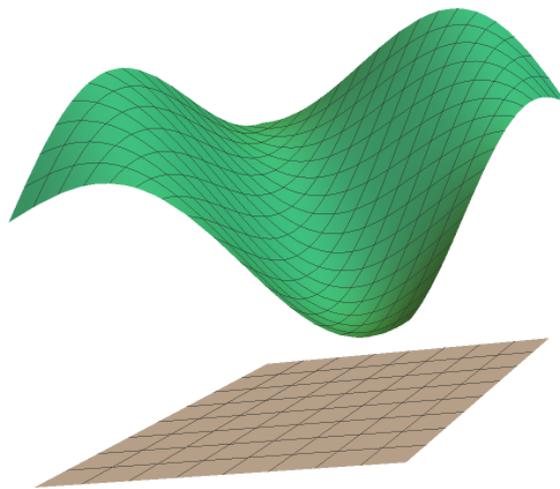


Figure 20: The physical domain  $\Omega$  and the exact solution of the Poisson's PDE.

Since we obtain the same rates of convergence as in the one dimensional case, in this work, we present the relative errors for the ICM using just GCP. In Figure 21 are shown the GCP on the physical domain  $\Omega$ , using B-spline basis functions of degree  $p = 3$  and knot vector with 8 elements. GCP are  $\tau_{ij} = \mathbf{G}(\hat{\tau}_{ij})$ , where  $\hat{\tau}_{ij} := (\hat{\xi}_i, \hat{\xi}_j)$  for  $i, j = 1, 2, \dots, 11$ , as defined in Def. 5.3.

In Figure 22 are presented the relative errors in the  $L^2$  and  $H_1$  norm for  $p = 3, 4, 5$  using GCP. Also in this example, as in the previous one, we take  $h_0 = 4$  and so  $h = \frac{1}{4}$ .

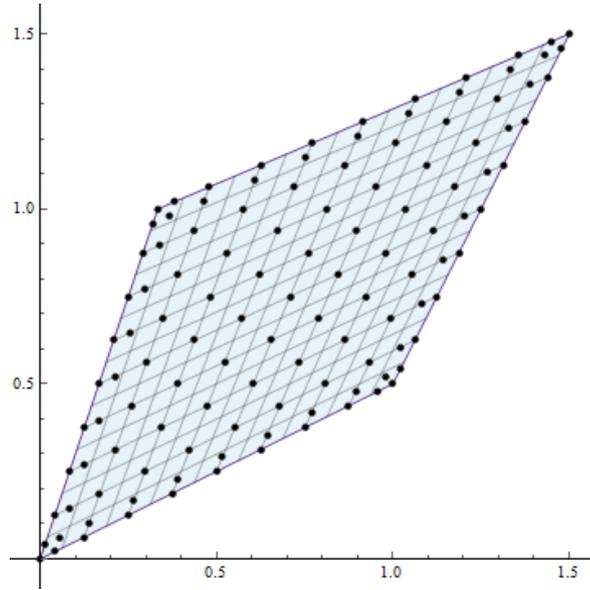
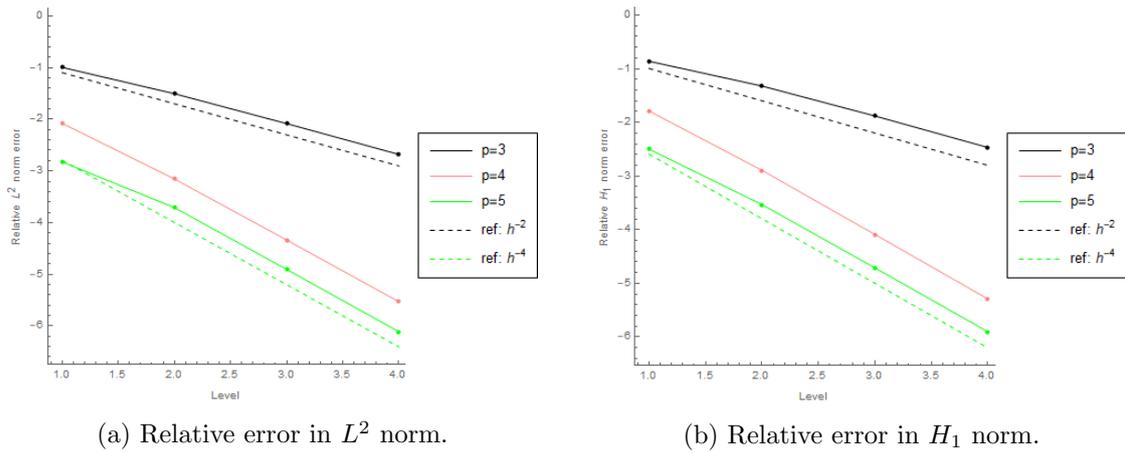
Figure 21: Distribution of GCP on  $\Omega$ .

Figure 22: 2D relative errors for GCP.

**Example 7.4.** Finally, in the last example, we consider as a computational domain  $\Omega$  the quarter of an annulus. Again we need the geometry mapping  $\mathbf{G}$  that maps the points from the domain  $[0, 1]^2$  to the domain  $\Omega$  and we define it using Bernstein basis polynomials. Firstly, we define two sets  $K^1$  and  $K^2$  in the next way:

$$K^1 = \left\{ \left( \cos\left(\frac{\pi}{18}i\right), \sin\left(\frac{\pi}{18}i\right) \right) \text{ for } i = 0, 1, \dots, 9 \right\}$$

$$K^2 = \left\{ \left( 2 \cos\left(\frac{\pi}{18}i\right), 2 \sin\left(\frac{\pi}{18}i\right) \right) \text{ for } i = 0, 1, \dots, 9 \right\}$$

Then the control points are defined as

$$\mathbf{B}_{i,j} = \begin{bmatrix} \cos\left(\frac{\pi}{18}i\right) + jk_i^1 \\ \sin\left(\frac{\pi}{18}i\right) + jk_i^2 \end{bmatrix}$$

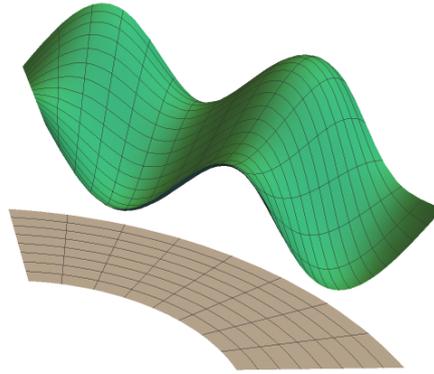


Figure 23: The physical domain  $\Omega$  and the exact solution of the Poisson's PDE.

for  $i, j = 0, 1, 2, \dots, 9$  and  $k_i^1, k_i^2 \in \mathbb{R}$  are such that  $\cos(\frac{\pi}{18}i) + 9k_i^1 = 2\cos(\frac{\pi}{18}i)$  and  $\sin(\frac{\pi}{18}i) + 9k_i^2 = 2\sin(\frac{\pi}{18}i)$ . The geometry mapping  $\mathbf{G}(u, v)$  is defined as:

$$\mathbf{G}(u, v) = \sum_{i,j=0}^9 \mathbf{B}_{i,j} B_{i,j}^9(u, v).$$

We observe the two dimensional Poisson's PDE with the same exact solution as in the previous example; note that the boundary conditions are clearly different. The physical domain  $\Omega$  and the exact solution of the Poisson's PDE are shown in Figure 23; while in Figure 24 we can see the distribution of GCP on  $\Omega$ , using B-spline basis functions with polynomial degree  $p = 3$  and knot vector with 16 elements. Since the rate of convergence remains unchanged and the time complexity is big, we omit the graphs.

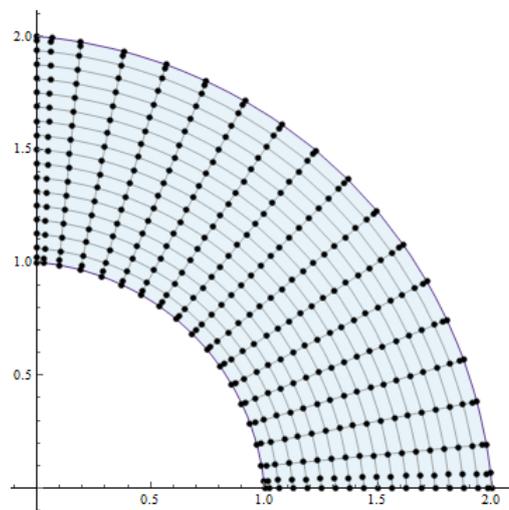


Figure 24: Distribution of GCP on  $\Omega$ .

## 8 CONCLUSION

In this thesis we investigated the rate of convergence of various collocation points on ICM on Poisson’s PDE in one and two dimensions. We began by reviewing the most significant properties of B-spline and NURBS basis functions, curves, and surfaces. Then we introduced a recent method for solving PDE, namely the IgA, as well as refinements, i.e. techniques for obtaining better isogeometric solutions. Then we described the first method,  $L^2$  approximation, in which we saw the geometry mapping in action for the first time. Then we moved on to PDEs, specifically to the Poisson’s PDE with various boundary conditions, in both dimensions and both formulations. In this thesis we focused exclusively on the Poisson’s PDE with Dirichlet boundary conditions. We observed the second approach, the GIM, thanks to these definitions. After that, we got to the heart of the issue: the ICM, which, unlike the GIM, employs the strong PDE formulation. Then we started to notice some collocation point families. GCP and DCP are the most widely utilized. The SCP are the most recent collocation points, with which we attempt to reach the same solution as the GIM by requiring the collocation residual to be zero at superconvergent points. Yet, because we do not have access to the actual location of the superconvergent points, we must approximate them; however, these “surrogate” points do not accurately approximate the Galerkin residual zeros throughout the domain. Then we select two subsets of superconvergent points with orders equal to degrees of freedom, these are the ASCP and CSCP. Finally, we ran several numerical tests of ICM of Poisson’s PDE in one and two dimensions, examining the rate of convergence on  $L^2$ ,  $H_1$  and  $H_2$  normed spaces. The following table summarizes our findings:

	GIM	GCP and DCP		LS-SP and CSCP		ASCP
		$p$ odd	$p$ even	$p$ odd	$p$ even	
$L^2$	$p + 1$	$p - 1$	$p$	$p + 1$	$p$	$p$
$H_1$	$p$	$p - 1$	$p$	$p$	$p$	$p$
$H_2$	$p - 1$	$p - 1$	$p - 1$	$p - 1$	$p - 1$	$p - 1$

Table 5: Comparisons of rate of convergence: GIM, GCP, DCP, LS-SP, CSCP, ASCP.

According to the Table 5, CSCP are the best choice for collocation points because the rate of convergence is optimal for all norms except the  $L^2$  norm when  $p$  is even. In

fact, finding a set of collocation points for even  $p$  such that the rate of convergence in the  $L^2$  norm is optimal, is still an open problem.

## 9 DALJŠI POVZETEK V SLOVENSKEM JEZIKU

V magistrski nalogi obravnavamo izogeometrično kolokacijsko metodo pri reševanju Poissonove parcialne diferencialne enačbe.

Parcialne diferencialne enačbe (v nadaljevanju PDE, glej [17, 19, 20]) se pojavljajo na več znanstvenih področjih. Ker so lahko tudi zelo kompleksne in jih zato ne znamo rešiti analitično, so v zadnjih letih razvili računske in numerične metode.

Najbolj uporabljena numerična metoda za reševanje PDE je metoda končnih elementov (FEM, glej [14]), katere slabost je, da moramo fizično domeno aproksimirati, njen izračun pa potrebuje veliko računskega časa. Da bi premostili to težavo, se je v zadnjih letih razvila nova numerična metoda, ki jo imenujemo Izogeometrična analiza (IgA) in so jo prvič obravnavali Hughes, Cottrell in Bazilevs v [4, 5]. Njihov cilj je bil zmanjšati razkorak med analizo končnih elementov (FEA) in računalniško podprtim oblikovanjem (CAD). Medtem ko s FEM domeno aproksimiramo z mrežo, v IgA uporabimo funkcije, ki direktno opišejo geometrijo. Značilnost IgA je ta, da uporabimo za bazne funkcije B-zlepke ali NURBS (neenakomerni racionalni B-zlepki) (glej [6, 7, 16, 18]). Te bazne funkcije oziroma B-zlepke in NURBS-e definiramo v drugem poglavju, kjer dokažemo tudi njihove najpomembnejše lastnosti kot so nenegativnost, particija enote in lokalna nosilnost. Ker so te funkcije enake tistim, ki jih uporabljamo pri grajenju CAD geometrij, so te zelo odvisne od prikaza geometrijske domene. Ta koncept imenujemo izogeometrični koncept in ga definiramo na začetku tretjega poglavja. Magistrsko delo nadaljujemo z definicijami geometrij B-zlepkov in NURBS-ov, ki so krivulje in ploskve. Kot pomembne lastnosti krivulj in ploskev B-zlepkov navajamo: afina invariantnost, lastnost konveksne ovojnice in lokalna kontrola. Podobne lastnosti imajo tudi krivulje in ploskve NURBS-ov. V nadaljevanju opišemo teorijo IgA in spoznamo geometrijsko preslikavo. Sledi prikaz treh različnih tehnik izboljšav za boljše izogeometrično rešitev PDE, ki so: vstavljanje vozlov, višanje stopnje in k-zgoščevanje. Nadalje predstavimo prvo obravnavano metodo in sicer  $L^2$  aproksimacijo, s katero definiramo tudi  $L^2$  normiran prostor in kasneje tudi  $H_1$  in  $H_2$  normirane prostore, ki jih uporabljamo pri izračunu relativne napake v sedmem poglavju. V tem poglavju je izogeometrija omenjena v opombi 3.10, v kateri upoštevamo geometrijsko preslikavo pri parametrizaciji dvo-dimenzionalne domene.

V četrtem poglavju se osredotočimo na Poissonovo PDE in definiramo različne robne

pogoje in sicer Dirichletov robni pogoj, Neumannov robni pogoj in mešani robni pogoj. V magistrski nalogi se osredinimo na Poissonovo PDE z Dirichletovimi robnimi pogoji. Nato opišemo Galerkinovo izogeometrično metodo (GIM), ki je najbolj uporabljena izogeometrična metoda za reševanje PDE. Postopek te metode predvideva, da najprej pretvorimo PDE v šibko obliko (glej Lema 4.3 in Lema 4.4), ki potrebuje manj gladke funkcije kot krepka oblika. Nato izračunamo integrale in dobimo linearen sistem. Zaključimo tako, da rešimo linearen sistem, iz katerega dobimo koeficiente aproksimirane rešitve, ki pripadajo baznim funkcijam (to so B-zlepki ali NURBS funkcije). Ko obravnavamo Poissonovo PDE v dveh dimenzijah, moramo še upoštevati reparametrizacijo domene (glej Lema 4.7). Vendar ta metoda potrebuje pri izračunu integralov specifična integracijska pravila in natančnost rešitve je odvisna od kvalitete metode numerične integracije.

Glavna tema magistrske naloge je alternativna izogeometrična metoda za reševanje PDE. Ta metoda uporablja krepko formulacijo PDE na množici diskretnih točk, ki jih imenujemo kolokacijske točke. S kolokacijsko metodo rešujemo krepko obliko PDE, kar odpravi integracijo, potrebuje pa bolj regularen prostor. Izogeometrično kolokacijsko metodo (ICM) so prvič predstavili Auricchio, Da Veiga, Hughes, Reali in Sangalli v [2].

V petem poglavju predstavimo ICM za Poissonovo PDE v eni in dveh dimenzijah in obravnavamo dve različni vrsti kolokacijskih točk. Najbolj uporabljene kolokacijske točke so Grevillove in Demkove točke, ker so to klasične interpolacijske točke za poljubno stopnjo in regularnost zlepkov. Porazdelitev Grevillovih in Demkovih kolokacijskih točk je prikazana na slikah 5 in 6. Poznamo pa tudi novejšo navdušujočo alternativo. Gomez in de Lorenzis v [11] sta dokazala, da mora obstajati množica kolokacijskih točk, ki eksaktno reproducira Galerkinovo rešitev, kar posledično pomeni, da ima napaka ICM isti red konvergence kot GIM. Množico kolokacijskih točk sestavljajo ničle Galerkinovega residuala in jih imenujemo Cauchy-Galerkinove točke (glej [1, 11]). Žal te točke niso znane a priori, zato [1] izbere za nadomestne točke take, pri katerih se pod ustreznimi pogoji pojavi superkonvergenca drugega odvoda Galerkinove rešitve. Do tega prihaja, ker je Galerkinov reidual z uporabo Poissonove PDE ekvivalenten napaki aproksimacije drugega odvoda (glej [15]). Te točke imenujemo superkonvergentne točke in jih predstavimo v šestem poglavju. V tem sta navedena dva algoritma, s katerima izračunamo superkonvergentne točke v primeru, da je stopnja baznih funkcij soda (glej Algoritem 1) oziroma liha (glej Algoritem 2).

Ker je število superkonvergentnih točk večje od prostostne stopnje, [1] predlaga uporabo aproksimacije najmanjših kvadratov za reševanje predoločenega linearnega sistema. Ta sistem rešimo z metodo najmanjših kvadratov, ki je predstavljena v podpoglavju 6.2. Ta metoda pa ni ravno prava kolokacijska metoda. V [11, 15] se predlaga izbiro dveh različnih podmnožic superkonvergentnih točk, ki imajo kardinalnost enako prostostni stopnji. Te dve podmnožici sta sestavljeni iz Alternirajočih (en. Alternat-

ing) in Združenih (en. Clustered) kolokacijskih točk. Alternirajoče superkonvergentne kolokacijske točke tvorimo tako, da izberemo vsako drugo superkonvergentno kolokacijsko točko (glej Algoritem 3), Združene superkonvergentne kolokacijske točke pa tako, da izberemo vsako drugo dvojico točk (glej Algoritem 4).

Magistrsko nalogo zaključimo z analizo konvergenčnega reda ICM za Poissonovo PDE na omenjenih kolokacijskih točkah. V prvem primeru si ogledamo nehomogeno Poissonovo PDE v eni dimenziji, v drugem primeru pa homogeno Poissonovo PDE v dveh dimenzijah. V obeh primerih je fizična domena enaka parametrični domeni, ki je  $[0, 1]^2$ . V zadnjih dveh primerih pa fizično domeno spremenimo: fizična domena v enem primeru je bilinearna preslikava enotskega kvadrata (glej Sliko 21), v drugem pa je četrt kolobarja (glej Sliko 24). Kot je opisano v [1, 13, 15], tudi mi opazimo, da so redi konvergence do sedaj predstavljenih ICM vsaj za sode stopnje nekoliko slabši od redov konvergence GIM. Redi konvergenca za  $L^2$ ,  $H_1$  in  $H_2$  normirane prostore so prikazani v tabeli 5, iz katere je razvidno, da imajo, na primer, Alternirajoče superkonvergentne točke sub-optimalen red v  $L^2$  normi, medtem ko imajo vse obravnavane kolokacijske točke optimalen red v  $H_2$  normi. Prav tako pa imajo Združene superkonvergentne točke vsaj za lihe stopnje optimalne rede konvergenca, kar jih dela uporabne v praksi. Za sode stopnje ostaja odprt problem, kako poiskati družino kolokacijskih točk, ki bi tudi v  $L^2$  normi imela optimalen red.

## 10 REFERENCES

- [1] C. ANITESCU, Y. JIA, Y.J. ZHANG and T. RABCZUK, An isogeometric collocation method using superconvergent points. *Comput. Methods Appl. Mech. Engrg.* 284 (2015) 1073–1097. (Cited on pages 1, 2, 32, 59 in 60.)
- [2] F. AURICCHIO, L. BEIRÃO DA VEIGA, T.J.R. HUGHES, A. REALI and G. SANGALLI, Isogeometric Collocation Methods. *Mathematical Models and Methods in Applied Sciences* 20 (2010) 2075–2107. (Cited on pages 1, 14 in 59.)
- [3] A. BARTEZZAGHI, L. DEDÈ and A. QUARTERONI, Isogeometric Analysis of high order Partial Differential Equations on surfaces. *Computer Methods in Applied Mechanics and Engineering* 295 (2015) 446–469. (Cited on page 28.)
- [4] Y. BAZILEVS, J.A. COTTRELL and T.J.R. HUGHES, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg.* 194 (2005) 4135–4195. (Cited on pages 1, 16 in 58.)
- [5] Y. BAZILEVS, J.A. COTTRELL and T.J.R. HUGHES, *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009. (Cited on pages 1, 4, 10, 16, 20 in 58.)
- [6] C. DE BOOR, A Practical Guide to Splines. In *J.E. Mersden, L. Sirovich (eds)*, Applied Mathematical Sciences, Volume 27, Springer-Verlag New York, 2001. (Cited on pages 1, 30 in 58.)
- [7] C. DE BOOR, On calculating with B-splines. *Journal of Approximation Theory* 6 (1972) 50–62. (Cited on pages 1, 4 in 58.)
- [8] C. DE BOOR and MATHWORKS, INC, *Spline Toolbox for Use with MATLAB: User's Guide, Version 3*. MathWorks, 2005. (Cited on page 30.)
- [9] O. CHANON, *A numerical Study of the Isogeometric Collocation Method*, Master thesis, Politecnico di Milano, 2017. (Cited on pages 4, 10, 14 in 16.)
- [10] S. DEMKO, On the Existence of Interpolating Projections onto Spline Spaces. *Journal of Approximation Theory* 43 (1985) 151–156. (Cited on page 30.)

- [11] H. GOMEZ and L. LORENZIS, The variational collocation method. *Computer Methods in Applied Mechanics and Engineering* 309 (2016) 152–181. (Cited on pages 2, 32, 33, 39, 42 in 59.)
- [12] L. HONGWEI, X. YUNYANG, W. XIAO, H. QIANQIAN and R. JINGWEN, Iso-geometric Least-Squares Collocation Method with Consistency and Convergence Analysis . *Journal of Systems Science and Complexity* 5 (2020) 1656–1693. (Cited on page 38.)
- [13] T.J.R. HUGHES and A. REALI, An Introduction to Isogeometric Collocation Methods. V: G. Beer, S. Bordas (eds), *Isogeometric Methods for Numerical Simulation, Volume 561*, Springer Vienna, 2015, 173–204. (Cited on pages 1 in 60.)
- [14] T.J.R. HUGHES, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Inc., 1987. (Cited on pages 1, 23 in 58.)
- [15] M. MONTARDINI, G. SANGALLI and L. TAMELLINI, Optimal-order isogeometric collocation at Galerkin superconvergent points. *Computer Methods in Applied Mechanics and Engineering* 316 (2017) 741–757. (Cited on pages 1, 2, 32, 59 in 60.)
- [16] L. PIEGL and W. TILLER, *The NURBS Book*. Springer-Verlag Berlin, Second Edition, 1995. (Cited on pages 1, 4, 6, 9, 10, 11, 12 in 58.)
- [17] A. QUARTERONI, *Numerical Models for Differential Problems*, Springer-Verlag Italia, Second Edition, 2014. (Cited on pages 1, 16, 20, 24 in 58.)
- [18] D.F. ROGERS, *An Introduction to NURBS with Historical Perspective*, Academic Press, 2001. (Cited on pages 1 in 58.)
- [19] S. SALSA, *Partial Differential Equations in Action*, Springer, Cham, Second Edition, 2015. (Cited on pages 1, 20 in 58.)
- [20] E. SÜLI and D. MAYERS, *An Introduction to Numerical Analysis*. Cambridge University Press, 2003. (Cited on pages 1, 16, 17 in 58.)
- [21] L. WAHLBIN, *Superconvergence in Galerkin Finite Element Methods*, Springer, Berlin, 1995. (Cited on pages 33 in 35.)

# Appendices

# APPENDIX A GIM

(\* INPUT: open knot vector ( $\Xi$ ), polynomial degree of the B-splines basis functions ( $p$ ), exact solution of Poisson's PDE ( $U$ )  
OUTPUT: galerkin approximate solution (galerkin) \*)

```
GIM[p_,  $\Xi$ _, U_] := Module[{f, g, n, B, rhs, lhs, coeffgh, gh, stiff, force, coeff, galerkin},
```

```
  (*f(x) and g(x) of the Poisson's PDE*)
```

```
  f[x_] := D[U[x], {x, 2}];
```

```
  g[x_] := U[x];
```

```
  n = Length[ $\Xi$ ] - p - 1; (*number of B-spline basis functions*)
```

```
  B[i_,  $\xi$ _] := BSplineBasis[{p,  $\Xi$ }, i,  $\xi$ ]; (*B-spline basis functions  $N_{i,p}(\xi)$ *)
```

```
  (*approximation function  $g^h$  using  $L^2$  approximation*)
```

```
  (* matrix on left side *)
```

```
  lhs = Table[Table[NIntegrate[B[i, u] × B[j, u], {u, 0, 1}], {i, 0, n-1}], {j, 0, n-1}];
```

```
  (* vector on right side *)
```

```
  rhs = Table[NIntegrate[g[u] × B[k, u], {u, 0, 1}], {k, 0, n-1}];
```

```
  (*coefficients  $a_i$ *)
```

```
  coeffgh = LinearSolve[lhs // N, rhs // N];
```

```
  (*approximated solution*)
```

```
  gh[u_] := Sum[(coeffgh[[i+1]] × B[i, u]), {i, 0, (n-1)}];
```

```
  (*galerkin method*)
```

```
  (*stiffness matrix*)
```

```
  stiff = Table[Table[NIntegrate[D[B[i, u], u] × D[B[j, u], u], {u, 0, 1}], {i, 1, n-2}], {j, 1, n-2}];
```

```
  (*force vector*)
```

```
  force = Table[(-NIntegrate[f[u] × B[k, u], {u, 0, 1}] - NIntegrate[D[gh[u], u] × D[B[k, u], u], {u, 0, 1}]), {k, 1, n-2}];
```

```
  (*coefficients  $d_i$ *)
```

```
  coeff = LinearSolve[stiff // N, force // N];
```

```
  (*approximated solution*)
```

```
  galerkin[u_] := Sum[coeff[[i]] × B[i, u], {i, 1, n-2}] + gh[u];
```

```
  Return[galerkin]
```

```
]
```

# APPENDIX B ICM in 1D

```
(* INPUT: open knot vector ( $\Xi$ ), polynomial degree of the B-splines basis functions ( $p$ ), collocation points ( $\tau$ ),
exact solution of Poisson's PDE ( $U$ )
OUTPUT: numerical approximate solution (app) *)
ICM1D[ $\Xi$ _,  $p$ _,  $\tau$ _,  $U$ _] := Module[{f, g, n, B, rhs, row, fun, lhs, coeff, app},

  (*f(x) and g(x) of the Poisson's PDE*)
  f[x_] = D[U[x], {x, 2}];
  g[x_] := U[x];

  n = Length[ $\Xi$ ] -  $p$  - 1; (*number of B-spline basis functions*)
  B[i_,  $\xi$ _] := BSplineBasis[{ $p$ ,  $\Xi$ ], i,  $\xi$ ]; (*B-spline basis functions  $N_{i,p}(\xi)$ *)

  (* matrix on left side *)
  lhs = {};
  Table[
    row = Flatten[
      Table[
        (*local support property of  $N_{i,p}(\xi)$ *)
        If[ $\tau[[k]] \geq \Xi[[i+1]] \&\& \tau[[k]] \leq \Xi[[i+p+2]]$ ],
          If[( $k = 1 \mid\mid k = n$ ),
            B[i,  $\xi$ ] /.  $\xi \rightarrow \tau[[k]]$ ,
            (D[B[i,  $\xi$ ], { $\xi$ , 2}]) /.  $\xi \rightarrow \tau[[k]]$ 
          ]
        ], {i, 0, n-1}
      ];
    AppendTo[lhs, row];
    , {k, 1, n}];
  lhs = ReplaceAll[Null  $\rightarrow$  0][lhs];

  (* vector on right side *)
  rhs = {};
  Table[
    If[ $k = 1 \mid\mid k = n$ ,
      fun = g[x] /.  $x \rightarrow \tau[[k]]$ ,
      fun = f[x] /.  $x \rightarrow \tau[[k]]$ 
    ]
  ] *
  AppendTo[rhs, fun], {k, 1, n}];

  (*coefficients  $c_i$ *)
  coeff = LinearSolve[lhs // N, rhs // N];

  (* approximated solution *)
  app[u_] := Sum[(coeff[[i+1]] * B[i, u]), {i, 0, (n-1)}];

  Return[app]
]
```

# APPENDIX C ICM in 2D

```
(* INPUT: open knot vector ( $\Xi$ ), polynomial degree of the B-splines basis functions (p), collocation points in 1D ( $\tau$ ),
exact solution of Poisson's PDE (U), geometry mapping (G)
OUTPUT: numerical approximate solution (app) *)
ICM2D[ $\Xi$ _, p_,  $\tau$ _, U_, G_] := Module[{f, g, n, B,  $\tau$ 2, JacG, M, L, rhs, row, fun, lhs, coeff, app, errorL2},

  (*f(x,y) and g(x,y) of the Poisson's PDE*)
  f[x_, y_] := D[U[x, y], {x, 2}] + D[U[x, y], {y, 2}];
  g[x_, y_] := U[x, y];

  n = Length[ $\Xi$ ] - p - 1; (*number of B-spline basis functions in each direction*)

  (*tensor product of univariate B-spline basis functions:  $N_{i,p}(\xi) \times N_{i,p}(\eta)$ *)
  B[i_, j_,  $\xi$ _,  $\eta$ _] := BSplineBasis[{p,  $\Xi$ }, i,  $\xi$ ] * BSplineBasis[{p,  $\Xi$ }, j,  $\eta$ ];

   $\tau$ 2 = Parallelize[Table[{ $\tau$ [[i]],  $\tau$ [[j]]}, {i, 1, n}, {j, 1, n}]; (*tensor product of collocation points  $\tau$ *)

  JacG[ $\xi$ _,  $\eta$ _] := Grad[G[ $\xi$ ,  $\eta$ ], { $\xi$ ,  $\eta$ ]; (*Jacobian of geomtery mapping G*)

  M[ $\xi$ _,  $\eta$ _] := Sqrt[Det[Transpose[JacG[ $\xi$ ,  $\eta$ ]].JacG[ $\xi$ ,  $\eta$ ]] * Inverse[Transpose[JacG[ $\xi$ ,  $\eta$ ]].JacG[ $\xi$ ,  $\eta$ ]];

  L[ $\xi$ _,  $\eta$ _, i_, j_] := M[ $\xi$ ,  $\eta$ ].Grad[B[i, j,  $\xi$ ,  $\eta$ ], { $\xi$ ,  $\eta$ ];

  (*matrix on left side*)
  lhs = {};
  Table[
    row = Flatten[
      Table[
        (*local support property of  $N_{i,p}(\xi)$  and  $N_{i,p}(\eta)$ *)
        If[ $\tau$ 2[[k, 1]][[1]]  $\geq$   $\Xi$ [[i+1]] &&  $\tau$ 2[[k, 1]][[1]]  $\leq$   $\Xi$ [[i+p+2]] &&
           $\tau$ 2[[k, 1]][[2]]  $\geq$   $\Xi$ [[j+1]] &&  $\tau$ 2[[k, 1]][[2]]  $\leq$   $\Xi$ [[j+p+2]],
          If[(k == 1 || k == n || l == 1 || l == n),
            B[i, j,  $\xi$ ,  $\eta$ ] /.  $\xi \rightarrow \tau$ 2[[k, 1]][[1]] /.  $\eta \rightarrow \tau$ 2[[k, 1]][[2]],
             $\frac{(D[L[\xi, \eta, i, j][[1]], \{\xi\}] + D[L[\xi, \eta, i, j][[2]], \{\eta\}])}{(Sqrt[Det[Transpose[JacG[\xi, \eta]].JacG[\xi, \eta]])}$  /. ( $\xi \rightarrow \tau$ 2[[k, 1]][[1]],  $\eta \rightarrow \tau$ 2[[k, 1]][[2]]) // N
          ]
        ], {i, 0, n-1}, {j, 0, n-1}]]];
    AppendTo[lhs, row];, {k, 1, n}, {l, 1, n}];
  lhs = ReplaceAll[lhs, Null  $\rightarrow$  0];

  (* vector on right side *)
  rhs = {};
  Table[If[k == 1 || k == n || l == 1 || l == n,
    fun = g[x, y] /. x  $\rightarrow$  G[ $\tau$ 2[[k, 1]][[1]],  $\tau$ 2[[k, 1]][[2]]][[1]] /. y  $\rightarrow$  G[ $\tau$ 2[[k, 1]][[1]],  $\tau$ 2[[k, 1]][[2]]][[2]],
    fun = f[x, y] /. x  $\rightarrow$  G[ $\tau$ 2[[k, 1]][[1]],  $\tau$ 2[[k, 1]][[2]]][[1]] /. y  $\rightarrow$  G[ $\tau$ 2[[k, 1]][[1]],  $\tau$ 2[[k, 1]][[2]]][[2]]
  ] *
  AppendTo[rhs, fun], {k, 1, n}, {l, 1, n}];

  (*coefficients  $c_i$ *)
  coeff = LinearSolve[lhs // N, rhs // N];

  (* approximated solution *)
  app[u_, v_] := Sum[(coeff[[{j+1, n i}]] * B[i, j, u, v]), {j, 0, (n-1)}, {i, 0, (n-1)}];

  Return[app]
]
```

# APPENDIX D SCP

```
(* INPUT: open knot vector ( $\Xi$ ), polynomial degree of the B-splines basis functions (p)
   OUTPUT: superconvergent collocation points (scp) *)
SCP[ $\Xi$ _, p_] := Module[{r,  $\mu$ , variables, coeff1, numbrOfVariables, var, matrix, sol, vec, er, solution,
  points, n, elts,  $\xi$ , lengthelts, param, paramPoints, scp},
  r := p + 1;
   $\mu$  := p - 1;
  variables = {a, b, c, d, e, f}; (* $c_i$ *)
  coeff1 = ConstantArray[0, r]; (* $k_i$ *)
  If[Mod[p, 2] == 0,
    (numbrOfVariables := p / 2 + 1;
     var = Table[variables[[i]], {i, 1, numbrOfVariables}];
     (*matrix on left side*)
     matrix = Table[Table[D[LegendreP[i, x], {x, j}] /. x -> 1, {i, 1, r, 2}], {j, 0,  $\mu$ , 2}];
     sol := Solve[matrix.var == 0, var];
     (*vector with the obtained  $c_1, c_3, \dots, c_r$ *)
     vec := Flatten[{a, Table[sol[[1]][[k]][[2]], {k, 1, numbrOfVariables - 1}]}] /. a -> 1;
     (*vector with coefficients  $k_i$ *)
     coeff1 = Table[If[Mod[k, 2] == 0, coeff1[[k]] = 0, coeff1[[k]] = vec[[ (k + 1) / 2]], {k, 1, r}];
    ],
    (numbrOfVariables := (p + 1) / 2;
     var = Table[variables[[i]], {i, 1, numbrOfVariables}];
     (*matrix on left side*)
     matrix = Table[Table[D[LegendreP[i, x], {x, j}] /. x -> 1, {i, 2, r, 2}], {j, 1,  $\mu$ , 2}];
     sol := Solve[matrix.var == 0, var];
     (*vector with the obtained  $c_2, c_4, \dots, c_r$ *)
     vec := Flatten[{a, Table[sol[[1]][[k]][[2]], {k, 1, numbrOfVariables - 1}]}] /. a -> 1;
     (*vector with coefficients  $k_i$ *)
     coeff1 = Table[If[Mod[k, 2] == 0, coeff1[[k]] = vec[[k / 2]], coeff1[[k]] = 0], {k, 1, r}];
  ];
  (*error function on the interval [-1,1]*)
  er[ $\eta$ _] := Sum[coeff1[[i]] * LegendreP[i,  $\eta$ ], {i, 1, r}];
  (* $e_R''(\eta) = 0$  for  $\eta \in [-1, 1]$ *)
  solution := NSolve[D[er[v], {v, 2}] == 0 && v >= -1 && v <= 1, v];
  (*SCP on interval [-1,1]*)
  points := Table[solution[[i]][[1]][[2]], {i, 1, Length[solution]};
  (*reparametrization of superconvergent points on interval [0,1]*)
  n = Length[ $\Xi$ ] - p - 1;
  (*"elements" of  $\Xi$ *)
  elts = { $\Xi$ [[1]]};
  Table[If[ $\Xi$ [[i - 1]] !=  $\Xi$ [[i]],  $\xi$  =  $\Xi$ [[i]]; AppendTo[elts,  $\xi$ ], {i, 2, Length[ $\Xi$ ]}];
  (*element length of the unifrom and open knot vector  $\Xi$ *)
  lengthelts =  $\Xi$ [[p + 2]] -  $\Xi$ [[p + 1]];
  (*parametrization of the interval [-1,1], to the interval [ $\xi_{p-1}, \xi_{p+2}$ ])*)
  param[u_] := (lengthelts / 2) u + (lengthelts / 2);
  (*superconvergent points on [ $\xi_{p-1}, \xi_{p+2}$ ])*)
  paramPoints := param[points] // N;

  (*superconvergent Points*)
  scp = Flatten[Table[elts[[i]] + paramPoints, {i, 1, Length[elts] - 1}]];

  Return[scp]
]
```

# APPENDIX E LS-SP

(\* INPUT: open knot vector ( $\Xi$ ), polynomial degree of the B-splines basis functions ( $p$ ),  
superconvergent collocation points ( $scp$ ), exact solution of Poisson's PDE ( $U$ )  
OUTPUT: numerical approximate solution ( $app$ ) \*)

```
LSSP[ $\Xi$ _,  $p$ _,  $scp$ _,  $U$ _] := Module[{f, g, n, B, rhs, row, fun, lhs, coeff, app},
```

```
  (*f(x) and g(x) of the Poisson's PDE*)
```

```
  f[x_] = D[U[x], {x, 2}];
```

```
  g[x_] := U[x];
```

```
  n = Length[ $\Xi$ ] -  $p$  - 1; (*number of B-spline basis functions*)
```

```
  B[i_,  $\xi$ _] := BSplineBasis[{ $p$ ,  $\Xi$ }, i,  $\xi$ ]; (*B-spline basis functions  $N_{i,p}(\xi)$ *)
```

```
  (* matrix on left side *)
```

```
  lhs = {};
```

```
  Table[
```

```
    row = Flatten[
```

```
      Table[
```

```
        (*local support property of  $N_{i,p}(\xi)$ *)
```

```
        If[ $scp[[k]] \geq \Xi[[i+1]] \ \&\& \ scp[[k]] \leq \Xi[[i+p+2]]$ ,
```

```
          If[( $k == 1 \ || \ k == n$ ),
```

```
            B[i,  $\xi$ ] /.  $\xi \rightarrow scp[[k]]$ ,
```

```
            (D[B[i,  $\xi$ ], { $\xi$ , 2}]) /.  $\xi \rightarrow scp[[k]]$ 
```

```
          ]
```

```
        ], {i, 0, n-1}]
```

```
    ];
```

```
  AppendTo[lhs, row];
```

```
  , {k, 1, Length[ $scp$ ]}];
```

```
  lhs = ReplaceAll[Null  $\rightarrow$  0][lhs];
```

```
  (* vector on right side *)
```

```
  rhs = {};
```

```
  Table[
```

```
    If[ $k == 1 \ || \ k == n$ ,
```

```
      fun = g[x] /.  $x \rightarrow scp[[k]]$ ,
```

```
      fun = f[x] /.  $x \rightarrow scp[[k]]$ 
```

```
    ]  $\times$ 
```

```
    AppendTo[rhs, fun], {k, 1, Length[ $scp$ ]}];
```

```
  (*coefficients  $c_i$ *)
```

```
  coeff = LeastSquares[N[lhs], N[rhs]];
```

```
  (* approximated solution *)
```

```
  app[u_] := Sum[(coeff[[i+1]]  $\times$  B[i, u]), {i, 0, (n-1)}];
```

```
  Return[app]
```

```
]
```

# APPENDIX F ASCP

```
(* INPUT: open knot vector ( $\Xi$ ), polynomial degree of the B-splines basis functions (p),
          superconvergent collocation points (scp)
   OUTPUT: alternating superconvergent collocation points (ascp) *)
ASCP[ $\Xi$ _, p_, scp_] := Module[{n, ascp},

  n = Length[ $\Xi$ ] - p - 1; (*number of B-spline basis functions*)

  If[Mod[p, 2]  $\neq$  0,
    If[Mod[n, 2]  $\neq$  0,
      ascp = Delete[scp, Join[Table[{i}, {i, 2, (Length[scp]/2), 2}], Table[{i}, {i, Length[scp] - 1, (Length[scp]/2 + 3), -2}]],
        ascp = Delete[scp, Join[Table[{i}, {i, 2, (Length[scp]/2), 2}], Table[{i}, {i, Length[scp] - 1, (Length[scp]/2 + 1), -2}]]];
      (*additional points*)
      Table[ascp = Union[ascp, {scp[[q]]}, {scp[[Length[scp] - q + 1]}], {q, 2, p - 3, 2}];
      (*imposing Dirichlet boundary conditions*)
      ascp = Flatten[Join[{0}, ascp, {1}]];
    ,
    (*Greville collocation points*)
    ascp = Parallelize[Table[Sum[ $\Xi$ [[m]], {m, i + 1, i + p}]/p, {i, 1, n}]];
  ];

  Return[ascp]
]
```

# APPENDIX G CSCP

```
(* INPUT: open knot vector ( $\Xi$ ), polynomial degree of the B-splines basis functions ( $p$ ),
          superconvergent collocation points ( $\tau$ )
   OUTPUT: clustered superconvergent collocation points (cscp) *)
CSCP[ $\Xi$ _,  $p$ _,  $scp$ _] := Module[{ $n$ ,  $q$ , rightPoint, leftPoint, cscp},

   $n$  = Length[ $\Xi$ ] -  $p$  - 1; (*number of B-spline basis functions*)

   $q$  = 0;
  rightPoint = {};
  leftPoint = {};
  If[Mod[ $p$ , 2]  $\neq$  0,
    cscp = Delete[ $scp$ , Join[Table[{ $i$ }, { $i$ , 3, Length[ $scp$ ] - 1, 4}], Table[{ $i$ }, { $i$ , 4, Length[ $scp$ ], 4}]]];
    (*additional points*)
  While[Length[cscp]  $\neq$   $n$  - 2,
    rightPoint = Length[ $scp$ ] -  $q$ ;
    cscp = Union[cscp, { $scp$ [[rightPoint]]}];
    leftPoint =  $q$  + 1;
    cscp = Union[cscp, { $scp$ [[leftPoint]]}];
     $q$  =  $q$  + 1;
    (*imposing Dirichlet boundary conditions*)
    cscp = Flatten[Join[{0}, cscp, {1}]];
  ,
    (*Greville collocation points*)
    cscp = Parallelize[Table[Sum[ $\Xi$ [[ $m$ ]], { $m$ ,  $i$  + 1,  $i$  +  $p$ }] /  $p$ , { $i$ , 1,  $n$ }}];
  ];
  Return[cscp]
]
```

# APPENDIX H Relative errors in 1D

```
(* INPUT: exact solution of Poisson's PDE (U), numerical approximate solution (app)
OUTPUT: relative error in L2 norm (error) *)
L2in1D[U_, app_] := Module[{error},

  error = Sqrt[NIntegrate[Abs[U[u] - app[u]]^2, {u, 0, 1}] / NIntegrate[Abs[U[u]]^2, {u, 0, 1}]];

  Return[error]
]

(* INPUT: exact solution of Poisson's PDE (U), numerical approximate solution (app)
OUTPUT: relative error in H1 norm (error) *)
H1in1D[U_, app_] := Module[{error},

  error = Sqrt[NIntegrate[Abs[U[u] - app[u]]^2 + Abs[D[U[u] - app[u], {u, 1}]]^2, {u, 0, 1}] /
    NIntegrate[Abs[U[u]]^2 + Abs[D[U[u], {u, 1}]]^2, {u, 0, 1}]];

  Return[error]
]

(* INPUT: exact solution of Poisson's PDE (U), numerical approximate solution (app)
OUTPUT: relative error in H2 norm (error) *)
H2in1D[U_, app_] := Module[{error},

  error = Sqrt[NIntegrate[Abs[U[u] - app[u]]^2 + Abs[D[U[u] - app[u], {u, 1}]]^2 +
    Abs[D[U[u] - app[u], {u, 2}]]^2, {u, 0, 1}] /
    NIntegrate[Abs[U[u]]^2 + Abs[D[U[u], {u, 1}]]^2 + Abs[D[U[u], {u, 2}]]^2, {u, 0, 1}]];

  Return[error]
]
```

# APPENDIX I Relative errors in 2D

(\* INPUT: exact solution of Poisson's PDE (U), numerical approximate solution (app)

OUTPUT: relative error in  $L^2$  norm (error) \*)

L2in2D[U\_, app\_, G\_] := Module[{error},

```
error = Sqrt[NIntegrate[Abs[U[G[u, v][[1]], G[u, v][[2]]] - app[u, v]]^2, {u, 0, 1}, {v, 0, 1}] /  
NIntegrate[Abs[U[G[u, v][[1]], G[u, v][[2]]]]^2, {u, 0, 1}, {v, 0, 1}];
```

```
Return[error]
```

```
]
```

(\* INPUT: exact solution of Poisson's PDE (U), numerical approximate solution (app)

OUTPUT: relative error in  $H_1$  norm (error) \*)

H1in2D[U\_, app\_, G\_] := Module[{error},

```
error = Sqrt[NIntegrate[Abs[U[G[u, v][[1]], G[u, v][[2]]] - app[u, v]]^2 +  
Abs[D[U[G[u, v][[1]], G[u, v][[2]]] - app[u, v], {u, 1}, {v, 1}]]^2, {u, 0, 1}, {v, 0, 1}] /  
NIntegrate[Abs[U[G[u, v][[1]], G[u, v][[2]]]]^2 +  
Abs[D[U[G[u, v][[1]], G[u, v][[2]]], {u, 1}, {v, 1}]]^2, {u, 0, 1}, {v, 0, 1}];
```

```
Return[error]
```

```
]
```

(\* INPUT: exact solution of Poisson's PDE (U), numerical approximate solution (app)

OUTPUT: relative error in  $H_2$  norm (error) \*)

H2in2D[U\_, app\_, G\_] := Module[{error},

```
error = Sqrt[NIntegrate[Abs[U[G[u, v][[1]], G[u, v][[2]]] - app[u, v]]^2 +  
Abs[D[U[G[u, v][[1]], G[u, v][[2]]] - app[u, v], {u, 1}, {v, 1}]]^2 +  
Abs[D[U[G[u, v][[1]], G[u, v][[2]]] - app[u, v], {u, 2}, {v, 2}]]^2, {u, 0, 1}, {v, 0, 1}] /  
NIntegrate[Abs[U[G[u, v][[1]], G[u, v][[2]]]]^2 + Abs[D[U[G[u, v][[1]], G[u, v][[2]]], {u, 1}, {v, 1}]]^2 +  
Abs[D[U[G[u, v][[1]], G[u, v][[2]]], {u, 2}, {v, 2}]]^2, {u, 0, 1}, {v, 0, 1}];
```

```
Return[error]
```

```
]
```