Abstract

Isogeometric analysis with full quadrature yields optimal convergence rates but require higher computational cost than necessary for splines of maximal continuity. In this thesis two such methods, the weak variational method and the weighted residual method, are presented. These method are compared with three isogeometric collocation method, one collocated at Greville points and the others at different sets of superconvergent points. Isogeometric collocation at superconvergent point may yield one order suboptimal continuity in L^2-norm for even polynomial orders but otherwise provide the same accuracy as the isogeometric analysis methods, with just one evaluation point per degree of freedom. Correct selection of superconvergent points are vital to obtain these rates.

Sammendrag

Isogeometrisk analyse med full integrasjon gir optimale konvergensrater men krever høyere beregningkostnader enn det som er nødvendig med spliner av maksimal kontinuitet. I denne masteroppgaven er to slike metoder presentert, en svak variasjonsmetode og en vektet residualmetode. Disse metodene er sammenlignet med tre isogeometriske kollokasjonsmetoder, der en er kollokert ved Greville punkter og de andre ved forsjellige sett av superkonvergente punkter. Isogeometrisk kollokasjon ved superkonvergente punkter krever kun ett evalueringspunkt per frihetsgrad og kan gi konvergensrate som er suboptimal med én orden i L^2-normen ved jevn polynomisk orden, men er ellers like optimal som de isogeometriske analysemetodene. For å oppnå disse konvergensratene må de superkonvergente punktene velges med omhu.

Contents

Preface

The thesis presented before you is submitted as a part of the master's degree programme in applied mathematics at the Norwegian University of Science and Technology (NTNU). The thesis was conducted at Department of Mathematical Sciences in 2016. The target group of the thesis is readers experienced in finite element analysis with preferably some understanding of splines and the isogeometric analysis.

With this thesis I wish to give a throughout description of isogeometric analysis and isogeometric collocation based on Bézier extraction, with special emphasis on superconvergent points. The thesis is written with focus on parts I found most challenging about this topics and provide the reader with mathematical derivations and explanatory illustrations. The methods are implemented in MATLAB.

This thesis is greatly influenced by the article *The variational collocation method* by Gomez et al. and *Optimal-order isogeometric collocation at Galerkin superconvergent points* by Montardini et al. I hope this thesis will enrich the research environment and serve as an understanding supplement to students and researchers interested in this topic.

I would like to express my gratitude to my supervisor Trond Kvamsdal for introducing me to the topic and for his useful comments, remarks and guidance throughout the project. Furthermore I would like to thank Kjetil André Johannessen for always answering my questions with fruitful explanations. I would like to thank my family and friends for their continuous encouragement that always keep me on track and to Jan Christian Eliassen for his loving support. If I ever lost interest, you motivated me.

Silje Irene Hansen

Oslo, 2016

CHAPTER 1

Introduction

Isogeometric analysis (IGA) was initialized by Hughes et al. in [3] as an attempt to unify finite element analysis (FEA) with computer-aided design (CAD). This is done by using traditional CAD-bases in FEA methods. By using the same bases less analysis time and cost is spend on converting the basis and geometry to analysis-suitable spaces and geometries. With bases that exactly represent the CAD-space the geometry can be utilized exactly as designers constructed without approximation errors. In addition this change of bases provide higher accuracy in a range of applications, i.e structural vibrations, fluid-structure, shells, contact, fracture and phase-field problems. The most utilized basis functions are B-splines and NURBS, but also more advanced spline functions such as T-splines ([4, 5, 6]), hierarchical B-splines and locally refined B-splines are becoming more common in literature.

IGA methods include multiple integrals that must be solved. In traditional FEA these are typically approximated using Gaussian quadrature with yields optimal convergence orders. These methods are however inefficient when it comes to the number of evaluation points per degree of freedom. Full Gaussian quadrature does not account for the additional continuity over element boundaries provided by smooth basis functions used in IGA. This idea was first issued in [7] and further tackled in [8] and in [9]

Isogeometric collocation (IGC) aim at optimizing the computational efficiency with one evaluation point per degree of freedom. It was first introduced in [10] and further developet in among others [11, 12, 13]. Few evaluation points leads to a efficient method but for splines of maximum continuity the method is reported to give suboptimal convergence rates, when collocated at traditional collocation points, such as Greville or Demko abscissas.

Recently research have focused on isogeometric collocation at superconvergent points. This was first investigated in [14] in the form of least-squares. Then in [1] and in [2] further investigated these points and developed collocation methods that provide nearly-optimal convergence rates using just one evaluation point per degree of freedom. In [1] it was proposed to numerically integrate IGA methods with superconvergent points as quadrature points.

In this thesis Bernstein, B-spline and NURBS basis functions are briefly described in Chapter 2. These functions are used to develop two isogeometric analysis methods in Chapter 3, the weak variational method and the weighted residual method. In Chapter 4 isogeometric collocation is presented and further developed to isogeometric collocation at superconvergent points in Chapter 5. Chapter 6 briefly presents a weighted residual method with superconvergent quadrature points before numerical results are presented in Chapter 7 and Chapter 8. Chapter 9 conclude this thesis and Chapter 10 hold some thoughts on future work.

CHAPTER 2

Basis functions

This chapter presents basis functions that are utilized in the isogeometric analysis based on Bézier extraction. The most common basis functions applied to the isogeometric analysis are B-splines and non-uniform rational B-splines (NURBS). These are presented in Section 2.2 and 2.3, respectively. The Bézier extraction uses Bernstein polynomials, also called Bézier basis functions. These are theoretically simpler basis functions than spline basis functions and are presented in Section 2.1.

2.1 Bernstein polynomials

Bernstein polynomials are essential for the Bézier extraction approach which is used in Chapter 3. A Bézier curve generated by Bernstein polynomials is a special case of a spline curve, which is presented in the next section.

This section presents evaluation, derivation, geometric representation and some properties of Bernstein polynomials. Content in this section can be found in among others [15, ch. 1] and [16, ch. 4-5].

2.1.1 Bernstein polynomial evaluation and derivation

Bernstein polynomials B_i^p of order p are evaluated at parametric values $t \in [0,1]$ by

$$B_i^p(t) = \binom{p}{i} t^i (1-t)^{p-i}, \qquad \text{for } i = 0, \ldots, p. \tag{2.1}$$

$B_i^p(t)$ is set to zero or not defined for $t \notin [0,1]$. All Bernstein polynomials are nonnegative and the basis $\{B_i^p(t)\}_{i=0}^p$ constitutes a partition of unity $\sum_{i=0}^p B_i^p(t) = 1$.

The procedure for differentiating Bernstein polynomials $\frac{\mathrm{d}B_i^p}{\mathrm{d}t}$ is presented in among others [16, ch. 5.3] and results in

$$\frac{\mathrm{d}B_i^p}{\mathrm{d}t} = p(B_{i-1}^{p-1} - B_i^{p-1}), \tag{2.2}$$

where $B_{-1}^{p-1} \equiv B_p^{p-1} \equiv 0$. Further are the second derivative evaluated by

$$\frac{\mathrm{d}^2 B_i^p}{\mathrm{d}t^2} = p\left[\frac{\mathrm{d}B_{i-1}^{p-1}}{\mathrm{d}t} - \frac{\mathrm{d}B_i^{p-1}}{\mathrm{d}t}\right] = p(p-1)\left[B_{i-2}^{p-2} - 2B_{i-1}^{p-2} + B_i^{p-2}\right], \tag{2.3}$$

and so forth.

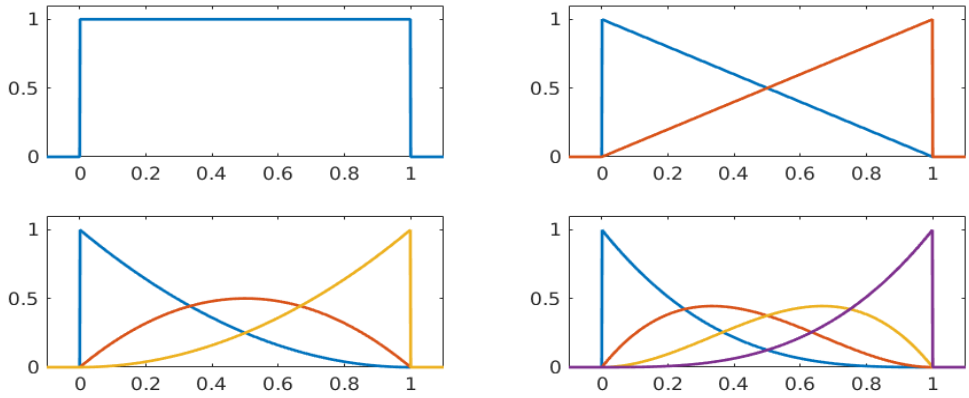


Figure 2.1: Bernstein basis functions of order $p = 0, 1, 2, 3, 4, 5$. The $p + 1$ basis functions are always positive in $(0, 1)$ and zero elsewheere.

A set of Bernstein basis functions are shown in Figure 2.1, for $p = 0, 1, 2, 3$. Their derivatives are illustrated in Figure 2.2. Observe that the polynomial order of the bases are reduced by one after differentiation and that Bernstein derivatives can be negative, even though Bernstein polynomials are nonnegative.

The set of Bernstein polynomials $\{B_i^p\}_{i=0}^p$ is a basis for the space of all polynomials of order p. Thus, a curve $\boldsymbol{q}^p(t)$ may be computed by linearly combining Bernstein polynomials $B_i^p(t)$ and $p + 1$ control points $\boldsymbol{c}$

$$\boldsymbol{q}^p(t) = \sum_{i=0}^{p} \boldsymbol{c}_i B_i^p(t). \tag{2.4}$$

The curve $\boldsymbol{q}^p(t)$ is called a *Bézier curve* and one such curve of order $p = 3$ is presented in Figure 2.3. Every Bézier curve lies in the convex hull of its control polygon and will in general interpolate its control points only at the ends.

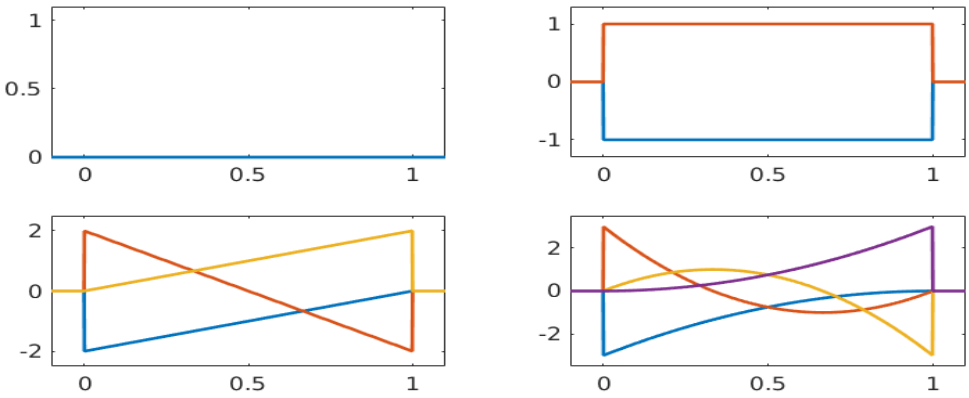


Figure 2.2: Derivation of the Bernstein bases in Figure 2.1 with $p = 0, 1, 2, 3$.

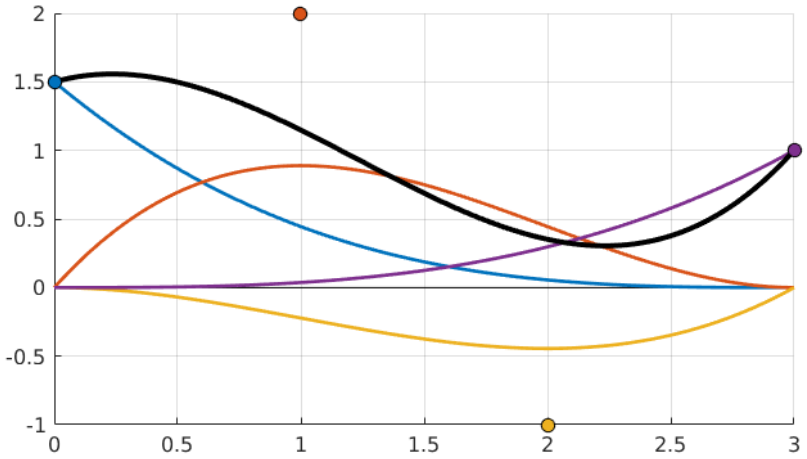


Figure 2.3: A Bézier curve generated by a set of Bernstein polynomials of order $p = 3$ and four control points. The curve interpolate its control points only at the ends.

A Bézier curve may also be formed recursively by *de Casteljau algorithm*: Let $p + 1$ control points $\{\boldsymbol{c}_i\}_{i=0}^{p}$ be given where p is a positive integer. The Bézier curve $\boldsymbol{q}^p(t) = \boldsymbol{q}_0^p(t)$ of degree p is determined by the recursion

$$\boldsymbol{q}_i^j(t) = (1 - t)\, \boldsymbol{q}_i^{j-1}(t) + t\, \boldsymbol{q}_{i-1}^{j-1}(t),$$

for $j = 1, \ldots, p$ and $i = j, \ldots, p$, where $\boldsymbol{q}_i^0(t) = \boldsymbol{c}_i$, for $i = 0, \ldots, p$.

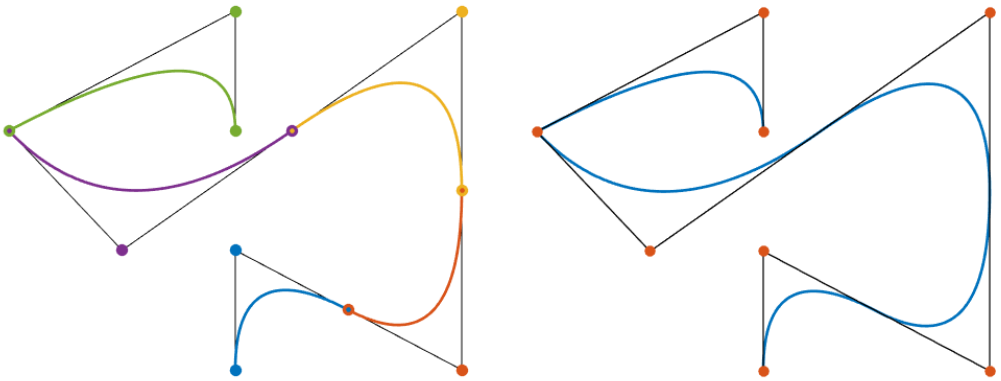
De Casteljau algorithm computes the same polynomial as Equation (2.4), without calculating any basis functions. A curve generated by this method is said to

be in Bézier form and is more numerical stable than a curve based on Bernstein polynomials. Since the curve generated by the two approaches are equal Bernstein polynomials are sometimes referred to as Bézier basis functions.

2.1.2 Continuity across Multiple Elements

Bézier curves require polynomials of high order to satisfy a large number of constraints. Computations of polynomials of high orders are inefficient and unstable [15]. Therefore, complex curves are divided into segments, also called *elements*, that are piecewise polynomials. Figure 2.4 **a)** presents a composite Bézier curve, where each colored segment correspond an element.

To ensure continuity across element boundaries correct placement of control points must be calculated. This complicates the design and manipulation of geometries and results in redundant control points. The curve in Figure 2.4 **a)** use 15 control points but can be uniquely defined with only eight. A set of basis functions that can generate exactly the same curve without redundant control points are B-splines. Such a curve is shown in Figure 2.4 **b)** and is detailed in the following section.



a) Bézier curve made up of five Bézier elements of different colors. The continuity is enforced by correct placement of 15 control points.

b) Spline curve made up of five B-spline elements determined by only eight control points.

Figure 2.4: A C^1-continuous curve defined over five elements generated by Bernstein polynomials in **a)** and B-splines in **b)**. At the leftmost knot where the curve is C^0-continuous.

2.2 B-splines

Previous section presents Bernstein polynomials and Bézier curves. To avoid high polynomial orders when dealing with a large number of constraints these curves are partitioned into segments. This leads to redundantly stored control points and complicates geometric modification. This section presents B-splines that are used to produce smooth, complex curves without redundant control points, see Figure 2.4. B-splines are simple to construct and modify geometrically, which makes them popular in CAD-engineering. This section briefly explains topics that are needed for the isogeometric analysis, such as the B-spline parameter space, B-spline evaluation and derivation, generation of objects, knot insertion and curve approximation. Content in this chapter are found in among others [15], [16] and [17].

2.2.1 Defining the parameter space

Every Bézier element Ω_e are mapped from the same parameter element by a *local map* Ψ_e. B-splines however, maps a *patch* of multiple elements between parameter and physical space with a single *global mapping* Φ. This difference is illustrated in Figure 2.5. A patch is a subdomain, typically used when the domain is too complex, repetitive, consist of different materials or when the geometry is processed in parallel.

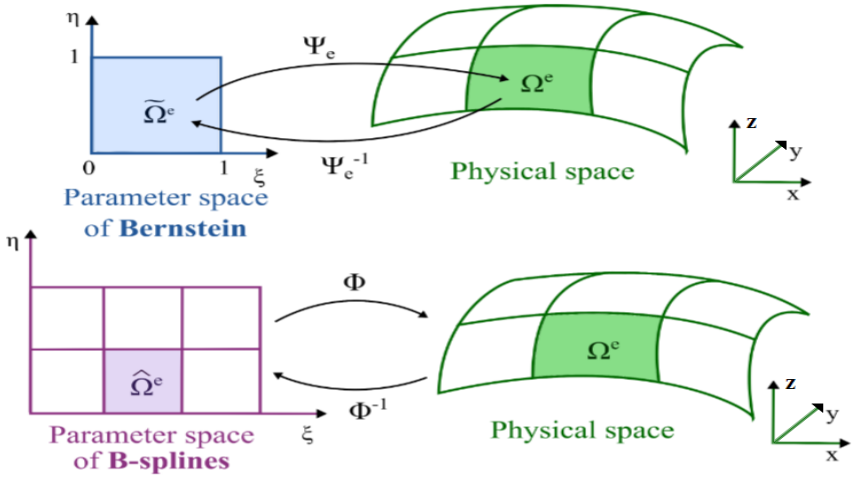


Figure 2.5: Bézier elements are mapped from one parameter element with local mappings Ψ_e. A patch of multiple spline elements are mapped from parameter to physical space with a single global mapping Φ.

A patch is partitioned into elements by *knots* $\xi_i \in \mathbb{R}$. A non-decreasing set of knots is called a *knot vector* $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, and defines a parametric dimension with n basis functions of polynomial order p. A knot vector is *uniform* when knots are equally spaced. If a knot value is repeated m times it is said to have *multiplicity* m.

A knot vector where the first and last knot have multiplicity $p+1$ is called *open*, and the resulting curve will interpolate its control point at the ends. When two preceding knots are distinct their knot span $[\xi_j, \xi_{j+1}]$ defines an element. If a knot span have zero measure it will still affect B-spline properties.

2.2.2 B-spline basis functions

B-spline basis functions $\{N_i\}_{i=1}^{n}$ may be computed using *Cox-de Boor recursion formula*. Given a polynomial order p and a knot vector Ξ the formulae is

$$N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi), \tag{2.5}$$

where basis functions of order $p = 0$ are defined as piecewise constant functions

$$N_i^0(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi \leq \xi_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

Here the convention '0/0 = 0' is required. Computation of the three first B-spline basis functions for a uniform knot vector $\Xi = \{1, 2, 3, \dots\}$ of order $p = 0, 1, 2, 3$ are presented in Figure 2.6.

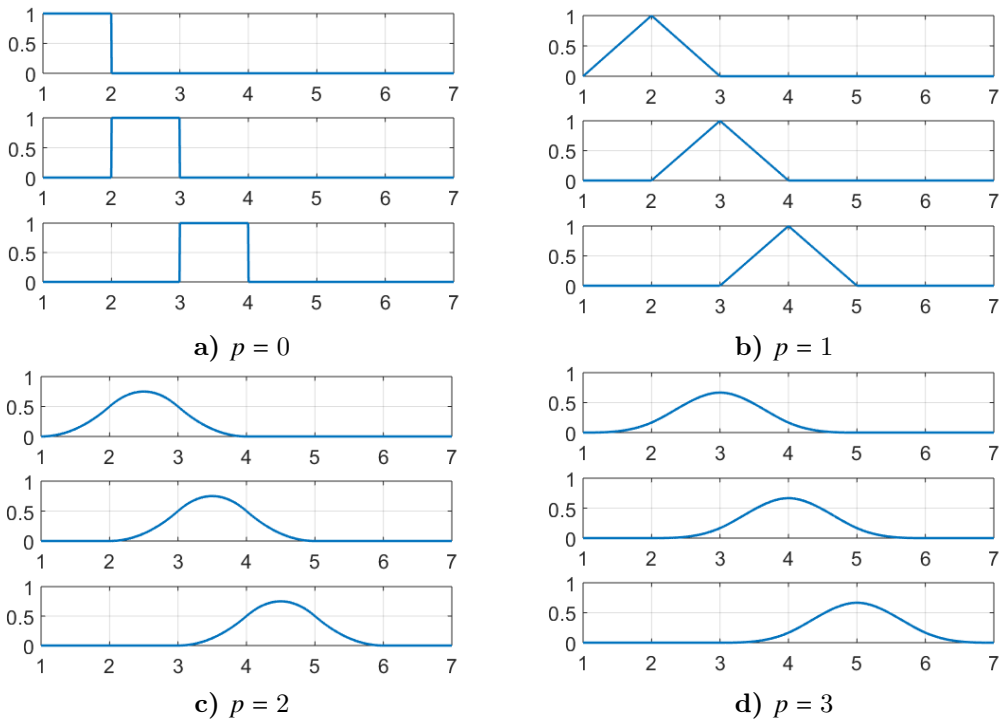


Figure 2.6: The three first B-spline basis functions generated on a uniform knot vector $\Xi = \{1, 2, 3, \cdots\}$ with polynomial order $p = 0, 1, 2, 3$.

A B-spline N_i^p has local support on $[\xi_i, \xi_{i+p+1}]$ and is zero everywhere, except in the open domain (ξ_i, ξ_{i+p+1}) where it is always positive. Thus, each knot span $[\xi_j, \xi_{j+1}]$ where $j = (p+1), \ldots, (n-p-1)$, supports only $p+1$ basis functions $\{N_i^p\}_{i=j-p}^{j}$. These properties are evident in Figure 2.6. B-spline bases satisfy the partition of unity

$$\sum_{i=0}^{p} N_i^p(\xi) = 1.$$

The continuity across element boundaries are determined by the knot vector Ξ. The basis is C^{p-m}-continuous across element boundaries with m repeated knot values. Within each element all basis functions are C^∞-continuous. In Figure 2.7 basis functions defined by an open knot vector with knots of different multiplicities are shown. The knot vector is $\Xi = \{0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4\}$ and the basis has polynomial order $p = 3$. As expected, the continuity decreases when the multiplicity increases. Also, all basis functions tend to "lean toward" knot values of higher multiplicity, because the a larger part of the local support is located here.

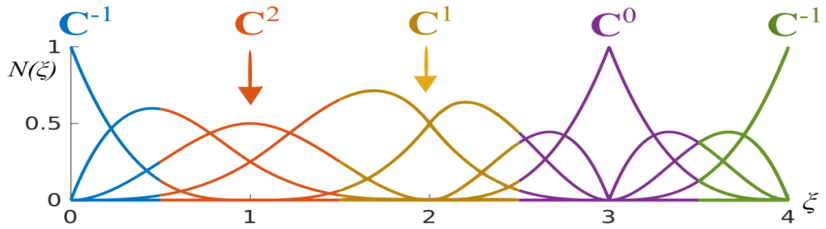


Figure 2.7: B-spline basis functions defined by an open knot vector $\Xi = \{0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4\}$, with $p = 3$. Basis functions are C^{p-m}-continuous at knots of multiplicity m.

2.2.3 Derivation of B-splines

Higher order derivatives of B-splines [15, pp. 49-61] are given by the formulae

$$\frac{\mathrm{d}^k}{\mathrm{d}\xi^k} N_i^p(\xi) = \frac{p!}{(p-k)!} \sum_{j=0}^{k} \alpha_{k,j} N_{i+j}^{p-k}(\xi), \tag{2.7}$$

where k is the order of derivation and

$$\begin{aligned}
\alpha_{0,0} &= 1, \\
\alpha_{k,0} &= \frac{\alpha_{k-1,0}}{\xi_{i+p+1-k} - \xi_i} \\
\alpha_{k,j} &= \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{i+j+p+1-k} - \xi_{i+j}}, \qquad j = 1, \ldots, k-1, \\
\alpha_{k,k} &= \frac{-\alpha_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}}.
\end{aligned} \tag{2.8}$$

The first, second and third order derivative of the basis from Figure 2.7 are presented in Figure 2.8. Notice that B-spline derivatives may be negative and that the continuity of the basis is reduced when differentiated.

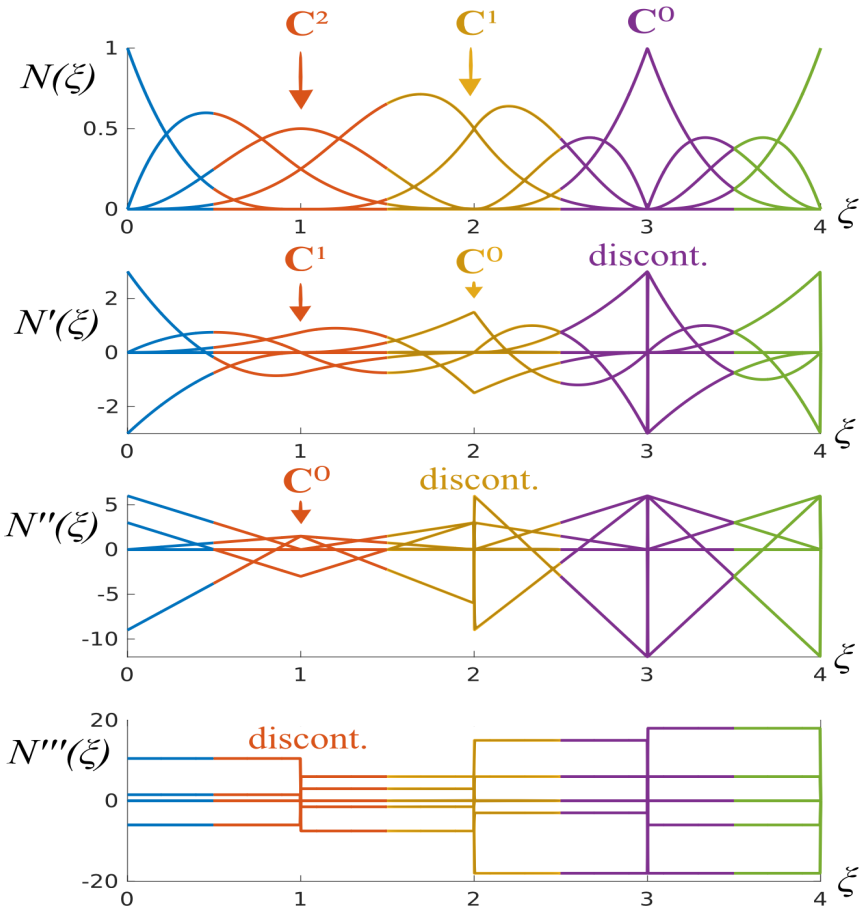


Figure 2.8: B-spline derivatives of the basis from Figure 2.7. The continuity is reduced by one for each differentiation.

2.2.4 Spline geometries

A spline geometry $\boldsymbol{G}(\boldsymbol{\xi})$ is generated by linearly combining a B-spline basis $\{N_i(\boldsymbol{\xi})\}_{i=1}^{n}$ with a set of control points $\mathcal{P} = \{\boldsymbol{P}_i\}_{i=1}^{n}$

$$\boldsymbol{G}(\boldsymbol{\xi}) = \sum_{i=1}^{N} N_i(\boldsymbol{\xi})\boldsymbol{P}_i. \tag{2.9}$$

Let three B-spline bases $\{\tilde{N}_i^{p_\xi}(\xi)\}_{i=1}^{n_\xi}$, $\{\tilde{M}_j^{p_\eta}(\eta)\}_{j=1}^{n_\eta}$ and $\{\tilde{L}_k^{p_\zeta}(\zeta)\}_{k=1}^{n_\zeta}$ of order p_ξ, p_η and p_ζ be defined respectively by knot vector

$$\begin{aligned}\Xi &= \{\xi_1, \xi_2, \ldots, \xi_{n_\xi+p_\xi+1}\},\\ \mathcal{H} &= \{\eta_1, \eta_2, \ldots, \eta_{n_\eta+p_\eta+1}\},\\ \mathcal{Z} &= \{\zeta_1, \zeta_2, \ldots, \zeta_{n_\zeta+p_\zeta+1}\}.\end{aligned}$$

Then for curves, $N_i(\boldsymbol{\xi}) = \tilde{N}_i(\xi)$ is the univariate B-spline. For surfaces $N_i(\boldsymbol{\xi})$ is the bivariate B-spline generated as a tensor product of two univariate B-splines, such that

$$\sum_{i=1}^{N} N_i(\boldsymbol{\xi}) = \sum_{i=1}^{n_\xi}\sum_{j=1}^{n_\eta} \tilde{N}_i^{p_\xi}(\xi)\tilde{M}_j^{p_\eta}(\eta).$$

And for solids, $N_i(\boldsymbol{\xi})$ is the trivariate B-spline, thus

$$\sum_{i=1}^{N} N_i(\boldsymbol{\xi}) = \sum_{i=1}^{n_\xi}\sum_{j=1}^{n_\eta}\sum_{k=1}^{n_\zeta} \tilde{N}_i^{p_\xi}(\xi)\tilde{M}_j^{p_\eta}(\eta)\tilde{L}_k^{p_\zeta}(\zeta).$$

The dimension of $\boldsymbol{P}_i$ determines the physical space of the object, e.g. a curve may lie in a three-dimensional space. Spline geometries inherits properties from their B-splines, such as polynomial order, continuity and compact support. Exact definition of these properties may be found in [15] and [17].

Generation of a spline geometry with Equation (2.9) can be seen as a mapping from parameter space to physical space, thus

$$\boldsymbol{x} = \Phi(\boldsymbol{\xi}) = \sum_{i=1}^{N} N_i(\boldsymbol{\xi})\boldsymbol{P}_i, \tag{2.10}$$

where Φ is the global mapping illustrated in Figure 2.5.

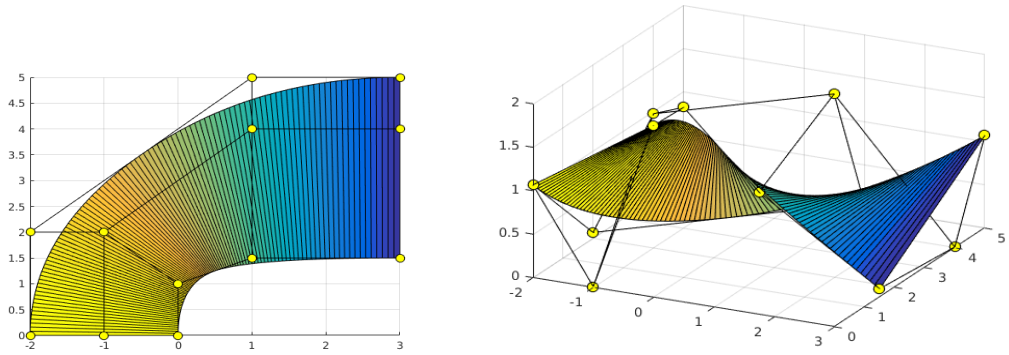
A spline curve is presented in Figure 2.4 **b)**. The curve is generated by basis functions of order $p = 2$ with knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$ and control points

$$\mathcal{P} = \{(1, 0), (1, 1), (2, 0), (2, 3), (1/2, 1), (0, 2), (1, 3), (1, 2)\}.$$

Figure 2.9 presents a spline surface generated by $\Xi = \{0, 0, 0, 1/2, 1, 1, 1\}$ and $\mathcal{H} = \{0, 0, 0, 2, 2, 2\}$ with $p_\xi = p_\eta = 2$ and control points

$$\mathcal{P}(x, y) = \begin{Bmatrix} (0,0) & (-1,0) & (-2,0) \\ (0,1) & (-1,2) & (-2,2) \\ (1,3/2) & (1,4) & (1,5) \\ (3,3/2) & (3,4) & (3,5) \end{Bmatrix}.$$

with z-components $z(x, y) = 1 - \sin(\frac{(3-x)^2+y^2}{2})$.



a) 2D-view **b)** 3D-view

Figure 2.9: B-spline surface generated by knot vectors of order $p_\xi, p_\eta = 2$ and defined by the yellow control points. The control polygon is shown in black. The surface is presented in two different views.

2.2.5 Knot Insertion

Knot insertion is a technique that adds new knots in a parametric dimension without altering the object geometrically or parametrically. Insertion of knots between existing ones forms new elements, while insertion of a knot on an existing knot value increases the multiplicity and thus reduces the continuity at this value. To preserve the geometric and parametric properties of the object, control points are placed stricly, e.g. by following [18, pp. 37-40] or [19, p. 17]. In this section the latter is presented due to its relevance to the Bézier extraction operator used to generate shape functions in Section 3.5 and Section 4.3.1.

Insertion of a new knot $\bar{\xi} \in [\xi_k, \xi_{k+1})$ to the knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$ results in the extended knot vector

$$\bar{\Xi} = \{\xi_1, \xi_2, \ldots, \xi_k, \bar{\xi}, \xi_{k+1}, \ldots, \xi_{m+p+1}\},$$

where $p < k < n$ and $\Xi \subset \bar{\Xi}$. The $m = n + 1$ new basis functions are generated by Equation (2.5) and (2.6) applied to $\bar{\Xi}$.

New control points $\bar{\mathcal{P}} = \{\bar{\boldsymbol{P}}_i\}_{i=1}^{m}$ are formed from original control points $\mathcal{P} = \{\boldsymbol{P}_i\}_{i=1}^{n}$, by

$$\bar{\boldsymbol{P}}_i = \begin{cases} \boldsymbol{P}_1, & i = 1, \\ \alpha_i \boldsymbol{P}_i + (1 - \alpha_i)\boldsymbol{P}_{i-1}, & 1 < i < m, \\ \boldsymbol{P}_n, & i = m, \end{cases} \tag{2.11}$$

where

$$\alpha_i = \begin{cases} 1, & 1 \le i \le k - p, \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \le i \le k, \\ 0, & i \ge k + 1. \end{cases} \tag{2.12}$$

2.2.6 Spline approximation

The non-interpolatory property of splines complicates approximations of splines to given functions or data points. This section presents *spline interpolation*, which is a spline approximation method that hold for B-splines of arbitrary polynomial order.

A spline $G(\boldsymbol{\xi})$ generated by a basis $\{N_j\}_{j=1}^{m}$ interpolates given data points (x_i, y_i) if

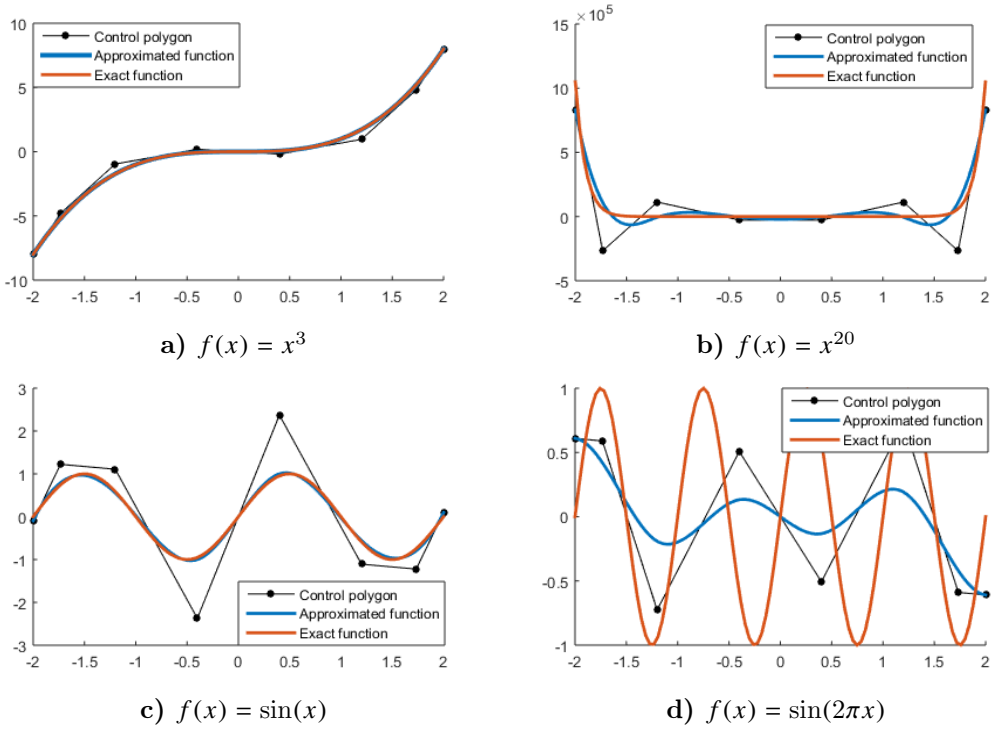
$$G(x_i) = \sum_{j=1}^{m} N_j(x_i) d_j = y_i \qquad \text{for } i = 1, \ldots, m.$$

The problem of spline interpolation is to find d_j such that $G(\xi_i)$ is as close to y_i as possible for $i = 1, \ldots, m$. This can be found by solving the square system

$$\begin{bmatrix} N_1(x_1) & \cdots & N_m(x_1) \\ \vdots & \ddots & \vdots \\ N_1(x_m) & \cdots & N_m(x_m) \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ d_m \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}.$$

Spline interpolation is exact when the function to be interpolated is in the spline space. Thus, all polynomials of degree up to p are interpolated exactly. This can be seen in Figure 2.10 **a)**, where $f(x) = x^3$ is exactly represented by a basis of polynomial order $p = 3$. Functions that are not represented in the spline space may still be reasonably accurate, e.g $f(x) = sin(x)$ in Figure 2.10 **c)**. But for some functions the approximation tend to oscillate or are smeared out, as observed by $f(x) = x^{20}$ in Figure 2.10 **b)** and by $f(x) = \sin(2\pi x)$ in Figure 2.10 **d)** respectively.

If the approximate solution is too oscillatory or too smeared an error minimizing method such as *least squares methods* may give better results.



a) $f(x) = x^3$ **b)** $f(x) = x^{20}$

c) $f(x) = \sin(x)$ **d)** $f(x) = \sin(2\pi x)$

Figure 2.10: Spline interpolation with a B-spline basis of polynomial order $p = 3$. The function of a) is exactly represented, while the approximate solution of b) are oscillating more that the exact solution. In c) the approximate solution is close to exact even though the function is not in the solution space. The approximate solution of d) is smeared out compared to the exact solution.

2.3 NURBS

B-splines enable exact generation of geometries that are represented by piecewise polynomials. Unfortionately, not all geometries are represented in this way and thus, are only approximated by B-splines, e.g. conic sections, like circles, ellipses and hyperbolas. With the introduction of non-uniform rational B-splines (NURBS) many of these geometries can be exactly represented. In this section some theory and formulas for NURBS are briefly stated.

NURBS have the same definition of knot vector as B-splines, but their control points are enlarged by one dimension. NURBS geometries in $\mathbb{R}^d$ are obtained as projective transformations of B-spline geometries in $\mathbb{R}^{d+1}$. Let $\mathcal{P}^w = \{\boldsymbol{P}_i^{\boldsymbol{w}}\}_{i=1}^n$ be the set of projective control points that define the projective B-spline geometry in $\mathbb{R}^{d+1}$. The last dimension of these points are referred to as the weight of the point

$$w_i = (\boldsymbol{P}_i^{\boldsymbol{w}})_{d+1}.$$

NURBS control points $\mathcal{P} = \{\boldsymbol{P}_i\}_{i=1}^n$ are obtained by dividing the remaining components of the projective control points with its weight

$$(\boldsymbol{P}_i)_j) = \frac{(\boldsymbol{P}_i^{\boldsymbol{w}})_j}{w_i}, \qquad j = 1, \ldots, d, \tag{2.13}$$

where $(\boldsymbol{P}_i^{\boldsymbol{w}})_j$ is the jth component of the ith control point. From the weights a *weighting function* $W(\xi)$ is defined as linear combinations with B-splines $N_i^p(\xi)$

$$W(\xi) = \sum_{i=1}^{n} N_i^p(\xi) w_i.$$

This is used to compute NURBS geometries. A NURBS curve is defined as

$$(C(\xi))_j = \frac{(C^w(\xi))_j}{W(\xi)}, \qquad j = 1, \ldots, d.$$

The projective curve $C^w(\xi) = \sum_{i=1}^n N_i^p(\xi)\boldsymbol{P}_i^{\boldsymbol{w}}$ is a standard B-spline curve. Both $C^w(\xi)$ and $W(\xi)$ are piecewise polynomial functions, which reveal the rational property of NURBS. This and the definition of control points in (2.13) yields

$$\begin{aligned}(C(\xi))_j &= \frac{(C^w(\xi))_j}{W(\xi)} = \frac{\sum_{i=1}^n N_i^p(\xi)(\boldsymbol{P}_i^{\boldsymbol{w}})_j}{W(\xi)} = \frac{\sum_{i=1}^n N_i^p(\xi) w_i (\boldsymbol{P}_i)_j}{W(\xi)} \\ &= \sum_{i=1}^{n} R_i^p(\xi)(\boldsymbol{P}_i)_j, \qquad j = 1, \ldots, d,\end{aligned}$$

where the rational function

$$R_i^p(\xi) = \frac{N_i^p(\xi) w_i}{W(\xi)}$$

is a NURBS basis function in one parametric direction. NURBS basis functions for higher dimensions are found using the same approach. In two and three parametric dimensions these are

$$\begin{aligned}R_{i,j}^{p_\xi,p_\eta}(\xi,\eta) &= \frac{N_i^{p_\xi}(\xi) M_j^{p_\eta}(\eta) w_{i,j}}{W(\xi,\eta)} &&= \frac{N_i^{p_\xi}(\xi) M_j^{p_\eta}(\eta) w_{i,j}}{\sum_{i=1}^{n_\xi}\sum_{j=1}^{n_\eta} N_i^{p_\xi}(\xi) M_j^{p_\eta}(\eta) w_{i,j}} \\ R_{i,j,k}^{p_\xi,p_\eta,p_\zeta}(\xi,\eta,\zeta) &= \frac{N_i^{p_\xi}(\xi) M_j^{p_\eta}(\eta) L_k^{p_\zeta}(\zeta) w_{i,j,k}}{W(\xi,\eta,\zeta)} &&= \frac{N_i^{p_\xi}(\xi) M_j^{p_\eta}(\eta) L_k^{p_\zeta}(\zeta) w_{i,j,k}}{\sum_{i=1}^{n_\xi}\sum_{j=1}^{n_\eta}\sum_{k=1}^{n_\zeta} N_i^{p_\xi}(\xi) M_j^{p_\eta}(\eta) L_k^{p_\zeta}(\zeta) w_{i,j,k}}\end{aligned}$$

NURBS have the same support and continuity as B-splines, they are pointwise non-negative and constituted a partition of unity property. NURBS with weights $w_i = 1, \ \forall\, i = 1, \ldots, n$ are just standard B-splines.

CHAPTER 3

Isogeomtric analysis using Galerkin method

Isogeometric analysis (IGA) attempts to unify computer-aided design (CAD) with finite element analysis (FEA), by incorporating the same smooth basis in FEA as is used in CAD. Recall that FEA numerically approximates solutions of physical problems, governed by differential equations. The method partitions the geometry into *elements*, evaluates an approximate solution within each element and combines these to one global approximate solution for the whole structure. This results in a system of linear equations which is easily solved using linear algebra on a computer. For more information on FEA the reader is referred to the practical books [20, 21, 22, 23] or the more mathematically technical books [24, 25].

This chapter presents IGA with B-splines as CAD-basis functions for the Poisson equation. Most essential mathematical techniques are captured by this example, but details may vary for other differential equations. Following the Bubnov-Galerkin method, a weak formulation is derived in Section 3.1 and the weighted residual method is derived in 3.2. These methods are further discretized to a Galerkin formulation in Section 3.3. The main implementation structure are presented in Section 3.4, where details concerning the smooth basis function and how these may be obtain through Bézier extraction are explained in 3.5. Section 3.7 presents error estimates. The content in this chapter are presented with greater details in [26].

3.1 Weak variational method

This section presents the weak variational method through a Poisson problem. This problem capture all essential mathematical techniques for the isogeometric framework.

For details concerning other differential equations the reader is referred to traditional FEA books on the topic.

The Poisson equation is a harmonic equation used to model vibrations on strings and membranes [27]. One Poisson problem is to find the solution $u : \Omega \to \mathbb{R}$ to the system

$$\begin{cases} \Delta u = -f, & \text{in } \Omega, \\ u = g, & \text{on } \Gamma_D, \\ \frac{\mathrm{d}u}{\mathrm{d}n} = h, & \text{on } \Gamma_N, \end{cases} \tag{3.1}$$

where $\boldsymbol{n}$ is the outward normal vector of the domain Ω. The boundary condition is composed of Dirichlet boundary condition on Γ_D and Neumann boundary condition Γ_N. The full boundary Γ have the properties

$$\Gamma_D \bigcup \Gamma_N = \Gamma,$$
$$\Gamma_D \bigcap \Gamma_N = \emptyset,$$

and functions $f : \Omega \to \mathbb{R}$, $g : \Gamma_D \to \mathbb{R}$ and $h : \Gamma_N \to \mathbb{R}$ are given.

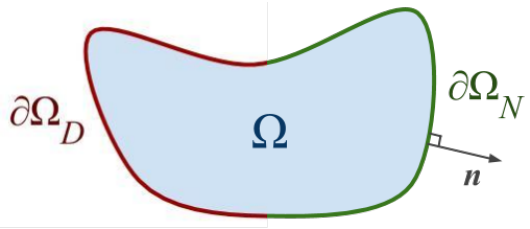


Figure 3.1: Example domain Ω with Dirichlet boundary Γ_D in red and Neumann boundary Γ_N in green.

The Poisson problem in (3.1) is in *strong form*. In traditional FEA, basis functions are not sufficiently smooth to represent all derivatives of this form without increasing complexity. Thus, a weak variational form is utilized to allow for functions with low order of continuity. A variational form is obtained by applying a technique from calculus of variations onto the problem. For the Poisson problem, multiply (3.1) with an arbitrary test function $v \in \mathcal{V}_S$ and integrate over the domain Ω

$$-\int_\Omega v \Delta u \, \mathrm{d}\Omega = \int_\Omega v f \, \mathrm{d}\Omega. \tag{3.2}$$

To find a solution $u \in \mathcal{U}_S$ such that (3.2) and the imposed boundary conditions hold for all $v \in \mathcal{V}_S$ is called a variational or *weighted residuals* formulation. The weak variational formulation is attained through integration by parts on the left side of Equation (3.2), which results in

$$\int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega - \int_\Gamma v \frac{\mathrm{d}u}{\mathrm{d}n} \, \mathrm{d}\Gamma = \int_\Omega v f \, \mathrm{d}\Omega. \tag{3.3}$$

For a solution to solve this problem it must be square integrable in its first derivative, so $u \in \mathcal{U}_W \subseteq H^1(\Omega)$. See Appendix A for information on Sobolev and Lebesgue spaces. From the boundary term it is evident that Neumann boundary conditions appear naturally and do not have to be enforced. Dirichlet boundary conditions must however be strongly enforced to the trial space, thus

$$u \in \mathcal{U}_W = \{u \in H^1(\Omega) \,:\, u|_{\Gamma_D} = g\}.$$

The test function space reflect the trial space, but are defined to be homogeneous on the Dirichlet boundary, so

$$v \in \mathcal{V}_W = \{u \in H^1(\Omega) \,:\, u|_{\Gamma_D} = 0\}.$$

Application of the natural Neumann boundary condition and the definition of $\mathcal{V}_W$ to Equation (3.3) results in

$$\int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega = \int_\Omega v f \, \mathrm{d}\Omega + \int_{\Gamma_N} v h \, \mathrm{d}\Gamma. \tag{3.4}$$

Here, all terms dependent on u are placed to the left and all independent terms are placed to the right. Thus, the resulting weak variational formulation is written:

Given functions f, g, h, find $u \in \mathcal{U}_W$ such that

$$a(u, v) = F(v), \qquad \forall\, v \in \mathcal{V}_W, \tag{3.5}$$

where

$$a(u, v) = \int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega, \tag{3.6}$$

$$F(v) = \int_\Omega v f \, \mathrm{d}\Omega + \int_{\Gamma_N} v h \, \mathrm{d}\Gamma. \tag{3.7}$$

3.2 Weighted residual method

The strong variational form, more commonly known as the weighted residual method, can be found directly from (3.2). However, as Neumann boundary conditions does not appear naturally in this form they are either strongly enforced into the trial space $\mathcal{U}_S$ or imposed by modifying the weak variational formulation. This may be done by reversely integrating the weak variational form (3.4) by parts. Again let the Dirichlet

boundary term disappear due to the definition that all test functions $v \in \mathcal{V}_C$ are zero on the Dirichlet boundary. This yields

$$\int_\Omega [\Delta u + f] v \, \mathrm{d}\Omega = \int_{\Gamma_N} [\nabla u \cdot \boldsymbol{n} - h] v \, \mathrm{d}\Gamma. \tag{3.8}$$

The trial, or candidate space $\mathcal{U}_S$ for this problem is the space of all functions that have square integrable first and second derivatives, so

$$\mathcal{U}_S = \{u \in H^2(\Omega) \, : \, u|_{\Gamma_D} = g \text{ and } \frac{\mathrm{d}u}{\mathrm{d}n}\Big|_{\Gamma_N} = h\},$$

where $H^2(\Omega)$ is a Sobolev space, as presented in Appendix A. Again, the test function space $\mathcal{V}_S$ reflects the trial function space, but is homogeneous on the Dirichlet boundary

$$\mathcal{V}_S = \{u \in H^2(\Omega) \, : \, u|_{\Gamma_D} = 0\}.$$

The resulting weighted residual method can be written as:

Given functions f, g, h, find $u \in \mathcal{U}_S$ such that

$$a(u, v) = F(v), \qquad \forall \, v \in \mathcal{V}_S, \tag{3.9}$$

where

$$a(u, v) = \int_\Omega v \Delta u \, \mathrm{d}\Omega, \tag{3.10}$$

$$F(v) = \int_{\Gamma_N} [\nabla u \cdot \boldsymbol{n} - h] v \, \mathrm{d}\Gamma - \int_\Omega v f \, \mathrm{d}\Omega. \tag{3.11}$$

Under appropriate regularity assumptions it can be shown that the weak formulation and the strong formulation of (3.9) are equivalent. For this proof the reader is referred to [20].

The weighted residual method is not used as often as the weak variational method. The weak form are preferred in traditional FEA because it requires a solution space which is not as smooth as that required for the weighted residual method. Because of this most well-established theory for FEA are applicable to the weak formulation but not necessarily to the weighted residual method. Even though the basis functions of IGA satisfies the smoothness requirements of the weighted residual, the weak formulation are often favored because of this, and that it provides a symmetric mass matrix and naturally imposes Neumann boundary conditions.

3.3 Galerkin formulation

Before applying numerical methods to the methods presented above they must be discretized to a finite set of equations. Following the Galerkin method this is done by approximating the candidate and test function spaces $\mathcal{S}$ and $\mathcal{V}$ with finite-dimensional subsets, respectively

$$\mathcal{S}^h \subset \mathcal{S} \quad \text{and} \quad \mathcal{V}^h \subset \mathcal{V}.$$

By substituting u and v in the weak formulation of Equation (3.5) with the discretized candidate function $\hat{u}^h \in \mathcal{S}^h$ and the discretized test function $v^h \in \mathcal{V}^h$ respectively, the weak formulation becomes

$$a(\hat{u}_h, v_h) = F(v_h). \tag{3.12}$$

Since every finite-dimensional vector space may be spanned by a finite dimensional basis [28], that is

$$\mathcal{S}^h = \text{span}(\{\phi_i\}_{i=1}^M) \quad \text{and} \quad \mathcal{V}^h = \text{span}(\{\varphi_j\}_{j=1}^N),$$

all elements of a vector space can be uniquely expressed as a linear combination of the basis [28]. Here, ϕ_i and φ_j are basis functions of the M- and N-dimensional vector spaces. For unique $\boldsymbol{b} \in \mathbb{R}^M$ and $\boldsymbol{d} \in \mathbb{R}^N$ the discretized candidate and discretized test functions are written

$$\hat{u}^h = \sum_{i=1}^{M} \phi_i b_i \in \mathcal{S}^h \quad \text{and} \quad v^h = \sum_{J=1}^{N} \varphi_j d_j \in \mathcal{V}^h. \tag{3.13}$$

Note that since $\hat{u}^h$ and v^h are in different spaces they are a combination of different bases. Implementation of the method requires generation of two different sets of basis functions. If the bases comprised the same space, just one set of basis functions are required, which would reduce complexity of the method. This is achieved by recognizing that every $\hat{u}^h \in \mathcal{S}^h$ is composed of a unique function $u^h \in \mathcal{V}^h$ and given data g^h

$$\hat{u}^h = u^h + g^h. \tag{3.14}$$

Substituting $\hat{u}_h$ in (3.12) with (3.14) yields

$$a(v^h, u^h) = F(v^h) - a(v^h, g^h), \tag{3.15}$$

where u^h, $v^h \in \mathcal{V}^h$. Define $\hat{u}^h$, u^h, v^h and g^h as weighted combinations with respect to basis functions as in (3.13). The approximate solution is calculated as the weighted sum

$$\hat{u}_h = \sum_{i=1}^{n} \varphi_i d_i + \sum_{j=1}^{n} \varphi_j g_j, \tag{3.16}$$

where n is the global number of degrees of freedom. This method is named the Bubnov-Galerkin method.

Not all basis functions are required in both defining the *lifting function* $g^h = \sum_{j=1}^{n} N_j g_j$ and the term $u^h = \sum_{i=1}^{n_p} N_i d_i$. In fact, they are determined by distinct subsets of functions. Let γ be the set of all global shape function numbers comprised by the subsets $\mathcal{B}$ and $\mathcal{D}$, such that

$$\mathcal{B} \bigcup \mathcal{D} = \gamma,$$
$$\mathcal{B} \bigcap \mathcal{D} = \emptyset.$$

Define $\mathcal{B}$ to be the set of all shape functions that are zero on the Dirichlet boundary

$$N_i|_{\Gamma_D} = 0 \qquad \forall \quad i \in \mathcal{B},$$

Note that this includes shape functions that are nonzero on the Neumann boundary. Further, define $\mathcal{D}$ to be the set of all shape functions that contribute to the Dirichlet boundary. Recall that $u^h|_{\Gamma_D} = \sum_{i \in \mathcal{D}} N_i d_i = 0$. Due to the linearly independence of B-splines this causes $d_i = 0$ for $i \in \mathcal{D}$. Thus, u^h can be written as a sum over only the elements in $\mathcal{B}$

$$u^h = \sum_{i \in \mathcal{B}} N_i d_i.$$

The lifting function $g^h = \sum_{j \in \gamma} N_j g_j$ is chosen so that it only gives contribution when included in the boundary, thus $g_i = 0$ for $i \in \gamma_\Omega$. So g^h can be written as a sum over only the elements of $\mathcal{D}$

$$g^h = \sum_{i \in \mathcal{D}} N_i g_i. \tag{3.17}$$

With this notation, the approximated solution $\hat{u}^h$ in (3.16) can be written

$$\hat{u}^h = \sum_{i \in \mathcal{B}} N_i d_i + \sum_{i \in \mathcal{D}} N_j g_j. \tag{3.18}$$

This numbering is useful when implementing the method and is presented in among others [18]. The notation is adopted from [1]. This results in the Galerkin formulation:

Find $\boldsymbol{d} \in \mathbb{R}^n$ such that

$$A\,\boldsymbol{d} = \boldsymbol{b}, \tag{3.19}$$

where

$$A_{ij} = a(\varphi_j, \varphi_i),$$
$$b_i = F(\varphi_i) - \sum_{k=1}^{n} g_k\, a(\varphi_i, \varphi_k), \qquad i, j = 1, \ldots, n,$$

with $a(\cdot,\cdot)$ and $F(\cdot)$ as defined in (3.6)-(3.7).

The same principle is used to discretize the weighted residual method of section 3.2. Thus, for both methods the problem is reduced to a system of linear equations which are easily solved with numerical linear algebra on a computer. The matrix $\boldsymbol{A}$ is often called a *stiffness matrix* and vector $\boldsymbol{b}$ is often called a *load vector*. Many properties and analysis of the method is determined by the basis functions chosen.

3.4 Implementation details

The main procedural structure of all finite element based methods are similar. Figure 3.2 illustrates the flow of such an implementation. Red boxes illustrates where IGA and IGC differ. The chart is inspired by [18]. This section briefly describe the general implementation, where some steps are more thoroughly explained in subsequent sections.

First, necessary problem data are read into the function. This input data are often preprocessed into data processing arrays that hold information about elements, global and elemental shape function numbers, and equation numbers of the system in an organized way. For more information about these arrays, see [26] and [18]. Further, are important variables for the method initialized, such as the global stiffness matrix A and load vector $\boldsymbol{b}$, before they are assembled in the succeeding element loop.

The element loop iterates all elements in the patch. For each element shape functions are computed for the required evaluation points. That is typically quadrature points for integral-based methods and collocation points for collocation-based methods. The shape function routine also differs between methods in the number of derivative that are required. The weak variational method require basis functions and their first-order derivative while method such as collocation and the weighted residual method also require the double derivative for the Poisson equation. The implementation of this routine are described in 3.5. In this thesis all shape functions are generated through Bézier extraction of B-splines.

With these shape functions, local contributions A^e and $\boldsymbol{b}^e$ to the global stiffness matrix and load vector are computed. In this computation the impact of boundary conditions are also regarded. After these computations, the local contributions are assembled in the global variables $\boldsymbol{A}$ and $\boldsymbol{b}$. When this is executed for all elements in the patch the system $\boldsymbol{Ad} = \boldsymbol{b}$ is solved. The approximate solution $\hat{u}^h$ is formed as a linear combination of its basis functions due to the non-interpolatory property of splines. At last, $\hat{u}^h$ is returned from the method.

The implementation described here is relevant for single-path objects. For multiple patches an outer loop iterating each patch is required. For efficiency some other changes may also be favourable, see [18, pp. 96-97].

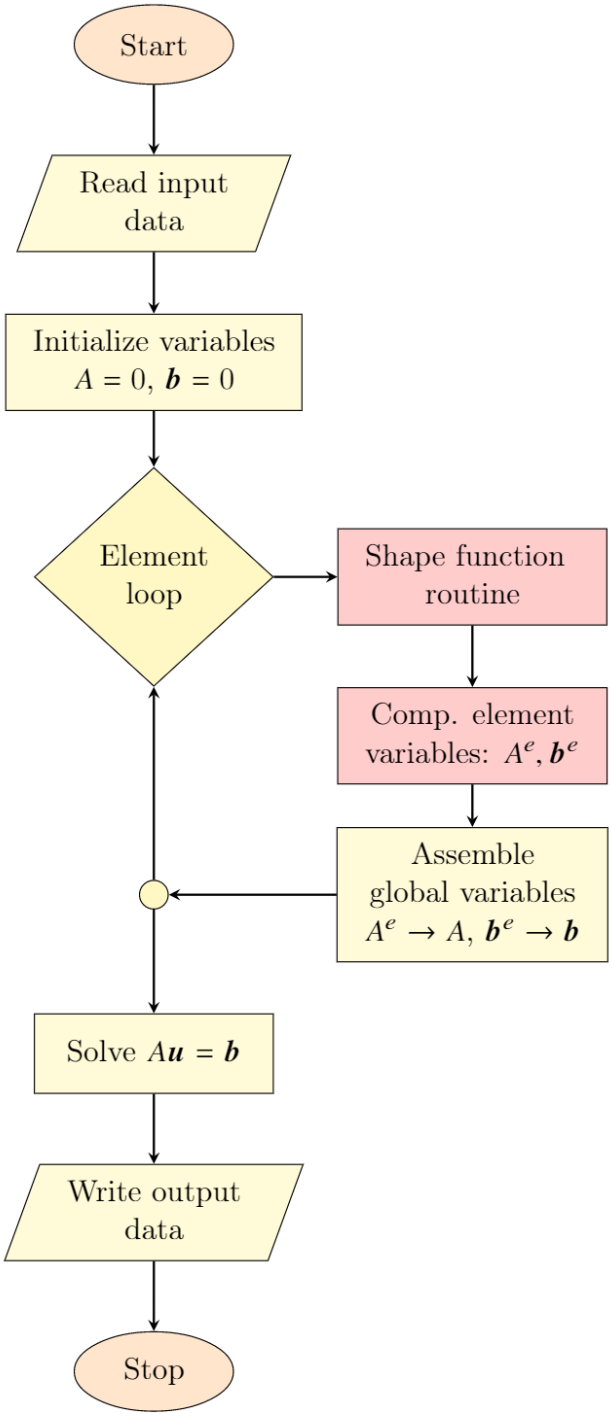


Figure 3.2: Flowchart for the finite element based methods. Red boxes represent routines where IGA and IGC differ. The chart is inspired by [18].

3.5 Shape functions

For each element, shape functions are evaluated at necessary point values. This section examine shape function routine for the weak variational method, which require basis functions and their first order derivatives with respect to physical space evaluated at quadrature points. For the weighted residual method or the collocation method, the double derivatives are required in addition, which is further described in Section 4.3.1. Here, the notations $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3) = (\xi, \eta, \zeta)$ are treated as equivalent to provide more compact equations. The same holds for $\boldsymbol{x} = (x_1, x_2, x_3) = (x, y, z)$. The length of these vectors may be changed to relate the theory to a d-dimensional problem. Uppercase letters in bold represent bases in matrix form.

Univariate B-splines may be calculated using Equations (2.5) and (2.6), and their first derivative with Equation (2.7). However, in this thesis all shape functions are generated through Bézier extraction. That is, for each element a set of Bézier basis functions are evaluated at necessary points, then a Bézier extraction operator is calculated and used to define B-splines. Calculation of the Bézier extraction operator and the theory behind this method is described in Appendix B. This approach provides an element structure to IGA that can be incorporated into existing FEA implementations without many changes to the main structure of the code.

The required Bézier basis functions and first order derivatives are evaluated over the parent domain $\widetilde{\Omega} = [0, 1]$ using Equation (2.1) and (2.2). Remark that a Bézier basis $B(\widetilde{\xi}_i)$ defined over the domain $[0, 1]$ is in general not identical to a Bézier basis $B(\xi_i)$ defined over an element in parametric space. Both bases have the same shape, but their values are spread over different spaces which yield different slopes. To form derivatives with respect to the parameter space the following holds

$$\frac{\partial B(\xi_i)}{\partial \xi_i} = \frac{\partial B(\widetilde{\xi}_i)}{\partial \xi_i} = \frac{\partial B(\widetilde{\xi}_i)}{\partial \widetilde{\xi}_i} \frac{\partial \widetilde{\xi}_i}{\partial \xi_i}. \tag{3.20}$$

If all elements locally contains the same evaluation points, then the Bézier basis for all elements are similar. Thus, one such basis of polynomial order p may be computed just once for the whole patch, and the computation may be placed outside the element loop, to increase efficiency.

Multivariate shape functions are computed by the Kronecker tensor product

$$\boldsymbol{B}(\xi, \eta) = \boldsymbol{B}(\boldsymbol{\xi}) = \boldsymbol{B}(\eta) \otimes \boldsymbol{B}(\xi),$$

where the Kronecker tensor product is defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \cdots & a_{mn}\boldsymbol{B} \end{bmatrix}.$$

The Bézier extraction method produces elementwise Bézier extraction operators for each dimension C_ξ^e and C_η^e, such that

$$\begin{aligned} \boldsymbol{N}^e(\xi) &= \boldsymbol{C}_\xi^e \boldsymbol{B}^e(\xi), \\ \boldsymbol{N}^e(\eta) &= \boldsymbol{C}_\eta^e \boldsymbol{B}^e(\eta), \end{aligned}$$

and the multivariate B-spline becomes

$$\begin{aligned} \boldsymbol{N}^e(\xi,\eta) &= \boldsymbol{C}_\xi^e \boldsymbol{B}^e(\xi) \otimes \boldsymbol{C}_\eta^e \boldsymbol{B}^e(\eta) \\ &= \left(\boldsymbol{C}_\xi^e \otimes \boldsymbol{C}_\eta^e\right) \left(\boldsymbol{B}^e(\xi) \otimes \boldsymbol{B}^e(\eta)\right) \\ &= \boldsymbol{C}^e \boldsymbol{B}^e(\xi,\eta). \end{aligned}$$

The same procedure also holds for derivatives, thus

$$\frac{\partial \boldsymbol{N}^e(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \boldsymbol{C}^e \frac{\partial \boldsymbol{B}^e(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}. \tag{3.21}$$

Note that $\frac{\partial \boldsymbol{N}^e(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}$ is the derivative of Bernstein polynomials with respect to parameter space. To obtain derivatives with respect to the physical space $\frac{\partial N^e(\boldsymbol{\xi})}{\partial x_i}$ the chain rule is applied

$$\frac{\partial N^e(\boldsymbol{\xi})}{\partial x_i} = \sum_{j=1}^{d} \frac{\partial N(\boldsymbol{\xi})}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_i}, \tag{3.22}$$

where $\frac{\partial \xi_j}{\partial x_i}$ is a component in the inverse of the gradient of the mapping between physical and parametric space $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} = \nabla\Phi(\boldsymbol{\xi})$. The mapping is presented in Equation (2.10) and its gradient has the form

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} = \nabla\Phi(\boldsymbol{\xi}) = \left(\frac{\partial \boldsymbol{x}}{\partial \xi_1}, \frac{\partial \boldsymbol{x}}{\partial \xi_2}, \frac{\partial \boldsymbol{x}}{\partial \xi_3}\right), \tag{3.23}$$

where

$$\frac{\partial \boldsymbol{x}}{\partial \xi_j} = \sum_{i=1}^{n} \frac{\partial N_i^p(\boldsymbol{\xi})}{\partial \xi_j} \boldsymbol{P}_i.$$

In addition to evaluating shape functions the shape function routine also include computation of the Jacobian determinant J of the mapping between parent space and physical space. This mapping is needed in numerical integration. The Jacobian determinant has the form

$$J = \left| \frac{\partial \boldsymbol{x}}{\partial \widetilde{\boldsymbol{\xi}}} \right| = \left| \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{\xi}}} \right|,$$

where $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}$ is calculated in Equation (3.23) and $\frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{\xi}}}$ is the gradient of the mapping between parent and parameter space. Both the parent space and the parameter space are square in 2D problems, therefore the $\frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{\xi}}}$ is calculated simply as element area in parameter space $\hat{A}$, divided by the area of the parent domain $\widetilde{A}$

$$\frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{\xi}}} = \frac{\hat{A}}{\widetilde{A}}.$$

Since $\widetilde{\Omega}$ is the unit square $\widetilde{A} = 1$, so the mapping between parent domain and parameter space is reduced to $\frac{\partial \boldsymbol{\xi}}{\partial \widetilde{\boldsymbol{\xi}}} = \hat{A}$.

3.6 Implementation of Boundary Conditions

Neumann boundary condition
The weak formulation naturally imposes the Neumann boundary conditions so implementation of Neumann boundary conditions are straight forward when the value of h is given at the mapped evaluation points. The contribution is calculated by approximating line integrals over the boundary.

Dirichlet boundary conditions
Implementation of Dirichlet boundary conditions are more demanding for IGA than it is in traditional FEA. That is because splines are not interpolatory and thus the components g_i of the lifting function from Equation (3.17) are not given directly from the problem description.

Homogeneous Dirichlet boundary conditions $g = 0$ are trivial and applied by simply setting $g_i = 0$ for all $i \in \mathcal{D}$. Recall that $g_i = 0$ for all $i \in \mathcal{B}$. When g is constant the partition of unity ensures that $g_i = g$ yields $g^h|_{\Gamma_D} = g$:

$$g^h|_{\Gamma_D} = \sum_{i \in \mathcal{D}} g_i \, N_i|_{\Gamma_D} + \sum_{j \in \mathcal{B}} 0 \, N_j|_{\Gamma_D} = g \sum_{i \in \mathcal{D}} N_i|_{\Gamma_D} = g.$$

(Recall that $N_i|_{\Gamma_D} = 0$, $\forall i \in \mathcal{B}$ and thus $\sum_{i \in \mathcal{D}} N_i|_{\Gamma_D} = 1$.)

When g is a variable function or a set of data points the lifting must be approximated. One such approximation technique is *spline interpolation* presented in Section 2.2.6. If the function exist in the spline space the lifting is exact, if not it yields approximation $g^h|_{\Gamma_D} \approx g$. Spline interpolation might lead to slightly smeared or oscillatory g^h. If this is not acceptable a better lifting function may be found by a curve fitting algorithm such as least-squares. If g is given as set of more data values than there are degrees-of-freedom the least-square approximation often give better results than selecting some points for spline interpolation.

3.7 Error estimates

A priori error estimates yields an upper bound for the error $e^h = u - u^h$ before the approximate solution $u^h \in \mathcal{S}^h$ is computed. Content in this section is highly influenced by the work in [26]. For FEA with Galerkin method on elliptic problems the following a priori error estimates are well known.

Theorem 3.7.1 *Let $u \in \mathcal{S}$ be the exact solution to the elliptical problem (3.1) with approximate solution $u^h \in \mathcal{S}^h$ found by traditional finite element method of degree r. If $u \in H^{r+1}(\Omega)$ then*

$$\left\|u - u^h\right\|_{H^k(\Omega)} \leq C h^{r+1-k} \, |u|_{H^{r+1}(\Omega)}, \qquad k = 0, 1, \tag{3.24}$$

where h is a mesh parameter and C is a constant independent of h and u.

The statement and proof of Theorem 3.7.1 are found in [24, ch. 4]. Equation (3.24) can be written as two estimates

$$\begin{aligned} \|u - u_h\|_{H^0(\Omega)} &\leq C h^{r+1} \, |u|_{H^{r+1}(\Omega)}, \qquad (k = 0), \\ \|u - u_h\|_{H^1(\Omega)} &\leq C h^{r} \quad |u|_{H^{r+1}(\Omega)}, \qquad (k = 1). \end{aligned}$$

The spaces $H^1(\Omega)$ and $H^0(\Omega) = L^2(\Omega)$ are defined in Equation (A.2) and (A.1) in the Appendix, with the norm and semi-norm defined respectively as (A.3) and (A.4).

The derivation of a priori error estimate for IGA with splines analogous to (3.24) is fairly technical and is found in [29]. The basic idea is explained in among others [30] and [18, app. 3.B]. These error estimates are harder to obtain because the support of each basis function comprises more than one element, and because the continuity of the solution may change from one element boundary to the next. In addition for NURBS, the rationality make the approximation properties harder to determine.

In [29] and [18] two important properties are presented that reduce the complexity. First, the geometrical mapping from parameter space to physical space $\Phi : \hat{\Omega} \to \Omega$ does not change with h-refinement. Second, the weighting function $W(\xi)$ for NURBS also remain unchanged during mesh refinement. The discussions and calculations of [29] results in the following error estimate for the approximation of elliptical problems:

Theorem 3.7.2 *Let k and l be integer indices such that $0 \leq k \leq l \leq p+1$ and let $u \in H^l(\Omega)$, then*

$$\sum_{e=1}^{n_{el}} |u - \Pi_k u|^2_{H^k(\Omega^e)} \leq C_{shape} \sum_{e=1}^{n_{el}} h_e^{2(l-k)} \sum_{i=0}^{l} \|\nabla \Phi\|^{2(i-l)}_{L^\infty(\Phi^{-1}(\Omega^e))} \, |u|^2_{H^i(\Omega^e)}, \tag{3.25}$$

where C_{shape} is a constant that depends on p, the shape of the domain Ω and the regularity of the mesh.

The mesh parameter h can be defined in many ways but is frequently defined to be the diameter of the smallest circle (in two dimensions) or sphere (in three dimensions) that can circumscribe any element in the mesh [18, ch. 3].

Since the geometrical mapping Φ remain unchanged during mesh refinement, the error is independent on $\|\nabla\Phi\|^{2(i-l)}_{L^\infty(\Phi^{-1}(\Omega^e))}$ when the mesh is h-refined. Thus, (3.25) yield the same results as (3.24) which in turn lead to the result that IGA using splines have the same optimal convergence rate as traditional FEA. That is, for smooth problems the convergence rate is $p+1$ with respect to h in the $L^2(\Omega)$-norm, p in the $H^1(\Omega)$-seminorm and $p-1$ in the $H^2(\Omega)$-seminorm. For non-smooth problems, e.g. a domain with singularities, the error changes with the factor h^α, where $\alpha = \min(p, \gamma)$ and γ is a real number resembling the strength of the singularity. Thus, with small γ the convergence is limited by the strength of the singularity and not by the polynomial order p [30].

This results states that IGA is a stronger method, since it require fewer degrees of freedom to generate solutions of the same high order as traditional FEA. The convergence rate is independent of the continuity of the mesh. Further study on this topic is found in [31], where the error estimates dependent on the order p and the global regularity k in addition to the mesh parameter h are considered. For a posteriori error estimates the reader is referred to [32].

CHAPTER 4

Isogeometric collocation

As opposed to the variational method presented in Chapter 3, collocation methods use the strong form of the problem enforced at a set of sites called collocation points. Since the strong form is used directly collocation methods demand additional smoothness to that needed for the weak form to be well posed. Isogeometric collocation (IGC) is collocation with spline basis functions, which satisfies the necessary smoothness requirements.

IGC methods require just one point evaluation per degree of freedom and has evident advantages in applications where efficiency is restrained by the cost of quadratures. The collocation method may have better accuracy than the variational method per unit of computation time [33], although its accuracy per degree of freedom and convergence rate is significantly lower for the collocation methods enforced traditional collocation points, such as Greville or Demko abscissas. This is especially evident for for odd degree p where the collocation is suboptimal by two orders in the L^2-norm and by one order in the H^1-seminorm. For collocation whit even p the method is only suboptimal by one order in the L^2-norm. This is evident in the numerical results presented in Chapter 7. In Chapter 5 collocation at superconvergent points are discussed with yield higher convergence rates.

In this chapter the isogeometric collocation is presented. In Section 4.1 the method is derived before it is discretized to a finite system of equations in Section 4.2. In Section 4.3 some implementation details are presented, with special emphasis on the shape function routine in Subsection 4.3.1.

4.1 Isogeometric collocation method

Isogeometric collocation (IGC) was initialized by Auricchio et al. in [10]. The following content may be found in among others this and in [11, 13]. A collocation method entails finding a solution $u \in \mathcal{U}_C$ to a problem in strong form enforced at a set of collocation points. This results in a set of collocation equations applied to points τ_i either at the interior Ω or on the boundary Γ.

At the Dirichlet boundary the conditions are strongly enforced a priori in the solution space $\mathcal{U}_C$, that is $u(\tau_i) = g(\tau_i)$ for $\tau_i \in \Gamma_D$. Thus, for a problem imposed by only Dirichlet boundary conditions the collocation equations are stated directly from the strong form (3.1) and the collocation method is to find $u \in \mathcal{U}_C$ such that

$$\Delta u(\tau_i) + f(\tau_i) = 0 \qquad \forall\ \tau_i \in \Omega. \tag{4.1}$$

Neumann boundary conditions however, does not appear naturally in strong form. Therefore, collocation equations for a Neumann boundary value problem are often obtained from its strong variational method derived in 3.2 as

$$\int_\Omega [\Delta u + f] v \, \mathrm{d}\Omega = \int_{\Gamma_N} [\nabla u \cdot \boldsymbol{n} - h] v \, \mathrm{d}\Gamma. \tag{4.2}$$

For the considered problem, select as test functions v of (4.2) the smooth Dirac delta function δ_i^ϵ that converge to the Dirac delta distribution located at τ_i when its compact support $\epsilon \to 0$. This distribution satisfies the sifting property, thus

$$\begin{aligned} \lim_{\epsilon \to 0} \int_\Omega f(x) \delta_i^\epsilon (\tau_i - x) &= f(\tau_i) \quad \forall\, \tau_i \in \Omega, \\ \lim_{\epsilon \to 0} \int_\Omega f(x) \delta_i^\epsilon (\tau_i - x) &= 0 \qquad \forall\, \tau_i \notin \Omega, \end{aligned}$$

for all $f \in [L^\infty(\Omega)]^2$ that are continuous around each τ_i [11]. Let $\epsilon \to 0$ results in three sets of equations depending on a collocation point's location

$$\begin{cases} [\Delta u + f](\tau_i) = 0 & \forall\, \tau_i \in \Omega, \\ [\nabla \cdot \boldsymbol{n} - h](\tau_i) = 0 & \forall\, \tau_i \in \Gamma_N \text{ on edge}, \\ [\nabla \cdot \boldsymbol{n}_{e_1} - h_{e_1}](\tau_i) + [\nabla u \cdot \boldsymbol{n}_{e_2} - h_{e_2}](\tau_i) = 0 & \forall\, \tau_i \in \Gamma_N \text{ on corner}, \end{cases} \tag{4.3}$$

where $\boldsymbol{n}_{e_j}$ and h_{e_j} belong to edge e_j. When $\tau_i \in \Gamma_N$ is a corner point, two edges e_1 and e_2 meet with possibly different normal vectors $\boldsymbol{n}_{e_j}$ and h_{e_j}, for $j = 1, 2$.

When a geometry consist of multiple patches, even more equations must be considered to account for inter-patch boundaries. For more details on boundary value conditions and patched geometries the interested reader is referred to [11].

4.2 Discretization

Collocating methods may be discretized using the same idea as presented for the weak variational method in 3.3. By introducing a discrete solution space

$$\mathcal{S}^h = \text{span}(\{\phi_i\}_{i=1}^M),$$

and substituting the solution u of the system (4.3) with the discretized candidate solution

$$\hat{u}^h = \sum_{i=1}^{M} N_i u_i \ \in \mathcal{S}^h, \tag{4.4}$$

the problem becomes a finite dimensional problem. When the problem is composed of non-homogeneous Dirichlet boundary conditions the Bubnov-Galerkin method can be applied, i.e. forcing the solution to be zero at the Dirichlet boundary by introducing

$$\mathcal{S}^h \ni \hat{u}^h = u^h + g^h.$$

Here g^h is the prescribed value at the Dirichlet boundary where $u^h \in \mathcal{V}^h$ is defined to be zero. This leads to a system of equations

$$\sum_{i \in \gamma_\Omega} u_i \Delta N_i(\boldsymbol{\tau}_j) = -f(\boldsymbol{\tau}_j) - \sum_{i \in \gamma_\Gamma} g_i \Delta N_i(\boldsymbol{\tau}_j) \qquad \forall\ j \in \gamma_\Omega.$$

which result in the problem of finding $\boldsymbol{u} \in \mathbb{R}^n$ such that

$$A\boldsymbol{u} = \boldsymbol{f}.$$

For one-dimensional problems the stiffness matrix A the components of the approximate solution $\boldsymbol{u}$ and the load vector $\boldsymbol{f}$ becomes

$$A = \begin{bmatrix} N_1''(\tau_1) & N_2''(\tau_1) & \cdots & N_n''(\tau_1) \\ N_1''(\tau_2) & N_2''(\tau_2) & \cdots & N_n''(\tau_2) \\ \vdots & \vdots & & \vdots \\ N_1''(\tau_n) & N_2''(\tau_n) & \cdots & N_n''(\tau_n) \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, \quad \boldsymbol{f} = - \begin{bmatrix} f(\tau_1) + \sum_{i \in \gamma_\Gamma} g_i N_i''(\tau_1) \\ f(\tau_2) + \sum_{i \in \gamma_\Gamma} g_i N_i''(\tau_2) \\ \vdots \\ f(\tau_n) + \sum_{i \in \gamma_\Gamma} g_i N_i''(\tau_n) \end{bmatrix}.$$

From which the approximate solution $\hat{u}^h$ is formed by using the components from vector $\boldsymbol{u}$ as weights in the linear combination of Equation (4.4).

4.3 Implementation details

Isogeometric collocation (IGC) is a finite element based method with the general implementation structure as described in 3.4. The flow of the implementation are presented in 3.2, boxes in red are routines where the implementation of IGA and ICS method are diverse. This section presents these routines for IGC methods.

4.3.1 Shape function routine

The shape function routine of the isogeometric collocation and the weighted residual method extends that of the weak variational method, which is presented in 3.5. In addition to basis functions, their derivatives and the Jacobian matrix, also the double derivatives are required. The mapping of double derivatives from parameter to physical space are constructed from a more intricate system of equations. This will be examined in the following.

In this section the following notation for variables in the physical space used interchangeably $\boldsymbol{x} = (x, y) = (x_1, x_2)$. This also holds for variables in the parameter space $\boldsymbol{\xi} = (\xi, \eta) = (\xi_1, \xi_2)$. Uppercase letters in bold represents a vectors of shape functions, i.e.:

$$\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \end{bmatrix}, \qquad \frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{x}^2} = \begin{bmatrix} \frac{\partial^2 N}{\partial x^2} \\ \frac{\partial^2 N}{\partial x \partial y} \\ \frac{\partial^2 N}{\partial y^2} \end{bmatrix}. \tag{4.5}$$

For compactness is the notation $\boldsymbol{N} = \boldsymbol{N}(\boldsymbol{x})$ used. To find the double derivative of a B-spline derivated with respect to physical space $\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{x}^2}$, are the chain rule and product rule applied to the derivative with respect to parameter space $\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{\xi}^2}$. This yields

$$\begin{aligned} \frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{\xi}^2} &= \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \right) = \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}} \right) \\ &= \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}^T \right) \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}^T \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}} \right) \\ &= \frac{\partial^2 \boldsymbol{x}}{\partial \boldsymbol{\xi}^2}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}} + \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}^T \right)^2 \frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{x}^2}, \end{aligned}$$

where $\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{x}^2}$, $\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{\xi}^2}$ and $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}}$ has the form presented in Equation (4.5), and

$$\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} \right)^2 = \begin{bmatrix} \left(\frac{\partial x}{\partial \xi} \right)^2 & \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} & \left(\frac{\partial x}{\partial \eta} \right)^2 \\ 2 \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} & 2 \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} \\ \left(\frac{\partial x}{\partial \xi} \right)^2 & \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} & \left(\frac{\partial x}{\partial \eta} \right)^2 \end{bmatrix},$$

and

$$\frac{\partial^2 \boldsymbol{x}}{\partial \boldsymbol{\xi}^2} = \begin{bmatrix} \frac{\partial^2 x}{\partial \xi^2} & \frac{\partial^2 x}{\partial \xi \partial \eta} & \frac{\partial^2 x}{\partial \eta^2} \\ \frac{\partial^2 y}{\partial \xi^2} & \frac{\partial^2 y}{\partial \xi \partial \eta} & \frac{\partial^2 y}{\partial \eta^2} \end{bmatrix}.$$

Solving this with respect to $\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{x}^2}$ yields

$$\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{x}^2} = \left[\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right)^2\right]^{-T} \left(\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{\xi}^2} - \frac{\partial^2 \boldsymbol{x}}{\partial \boldsymbol{\xi}^2}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}}\right),$$

For higher dimensional geometries these calculations may be extended by considering higher dimensional variables.

CHAPTER 5

Isogeometric collocation at superconvergent points

The work on isogeometric collocation at superconvergent points was initialized by Anitescu et al. in [14] where a least-squares method was utilized. Further, Gomez et al. presented some theory regarding Cauchy-Galerkin (CG) points in [1] and provided examples that just one such point per degree of freedom would produce the Galerkin solution. CG points are locations where the second derivative of the weak variational solution are exact. To be applicable these points must be estimated with superconvergent theory. In [2] Montardini et al. provide examples that the selection of superconvergent points affect the convergence rate of the method.

5.1 Existence of Cauchy-Galerkin points

This section mathematically proves that there exists a set of collocation points $\{\boldsymbol{\tau}_i\}_{i\in\mathcal{B}}$ such that collocation at these points produce the Galerkin solution exactly. It is further proven that for B-splines $\{N_i\}_{i=1}^n$, there exists at least n distinct collocation points, where each point $\boldsymbol{\tau}_i$ lies within the support of basis function N_i. These points are called Cauchy-Galerkin points. The proof in this section are influenced by [1].

First the strong form is multiplied by a test function $v \in \mathcal{V}$ and integrated over the domain Ω

$$\int_\Omega v(\boldsymbol{x})(\Delta u(\boldsymbol{x}) + f(\boldsymbol{x}))\,\mathrm{d}x = 0.$$

This equation is not integrated by parts but discretized in its strong form by choosing $u^h \in \mathcal{S}^h$ and $v^h \in \mathcal{V}^h$. This yields

$$\int_{\Omega} N_i(\boldsymbol{x})(\Delta u^h(\boldsymbol{x}) + f(\boldsymbol{x}))\,\mathrm{d}x = 0 \qquad \forall\ i \in \mathcal{B},$$

which is mathematically equivalent to the weak variational formulation. The spline basis function N_i is nonzero only in its support S_i, thus

$$\int_{S_i} N_i(\boldsymbol{x})(\Delta u^h(\boldsymbol{x}) + f(\boldsymbol{x}))\,\mathrm{d}x = 0 \qquad \forall\ i \in \mathcal{B}. \tag{5.1}$$

Further, the first mean value theorem of integral calculus is utilized as stated in the following theorem [34, 1]:

Theorem 5.1.1 *Let Ω be a measurable subset of $\mathbb{R}^d$. If $R : \Omega \to \mathbb{R}$ is continuous and $v : \Omega \to \mathbb{R}$ is a limited integrable function that never changes sign throughout Ω, then, there exists $\boldsymbol{\tau} \in \Omega$ such that*

$$\int_{\Omega} R(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}x = R(\boldsymbol{\tau}) \int_{\Omega} v(\boldsymbol{x})\,\mathrm{d}x.$$

Since $N_i \geq 0$ on S_i Theorem 5.1.1 can be applied to equation (5.1), which results in

$$(\Delta u^h(\boldsymbol{\tau}_i) + f(\boldsymbol{\tau}_i)) \int_{S_i} N_i(\boldsymbol{x})\,\mathrm{d}x = 0 \qquad \forall\ i \in \mathcal{B}. \tag{5.2}$$

Since the integral over all basis functions in a support are always positive this states

$$\Delta u^h(\boldsymbol{\tau}_i) + f(\boldsymbol{\tau}_i) = 0,$$

for a point $\boldsymbol{\tau_i} \in \mathcal{S}_i$. Thus, it is proven that for each support there exists one point that solves the weak variational form. In [1] it is further proved by the following Theorem that all such points in a problem are distinct.

Theorem 5.1.2 *Given a knot vector $\Xi = \{\xi_i\}_{i=1,\ldots,n+p+1}$ in the closed interval $[a, b]$ with knot multiplicity at most $p + 1$, i.e. $\xi_j < \xi_{j+p+1}$ for all $j = 1, \ldots, n$. If the integrable function $R : [a, b] \to \mathbb{R}$ is orthogonal to the spline space $\mathcal{S}_{p,\Xi}$ on $[a, b]$, that is*

$$\int_{a}^{b} s(x)R(x)\,\mathrm{d}x = 0 \qquad \forall\ s \in \mathcal{S}_{p,\Xi}$$

then, there exists $\boldsymbol{\Lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_{n+1}\}$ in $[a, b]$, strictly increasing with $\xi_i \leq \lambda_i \leq \xi_{i+p}$ (with equality only when $\xi_i = \xi_{i+p}$) so that R is also orthogonal to $\mathcal{S}_{0,\Lambda}$, that is

$$\int_{\lambda_i}^{\lambda_{i+1}} R(x)\,\mathrm{d}x = 0 \qquad \forall\ i = 1, \ldots, n. \tag{5.3}$$

The proof of Theorem 5.1.2 is explained in [1] and fully detailed in [35]. Applying the mean value theorem of integral calculus to equation (5.3) yields

Corollary 5.1.3 *If R is continuous, then there exists the set $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_n\}$, strictly increasing, with $\tau_i \in (\xi_i, \xi_{i+p+1})$ for all $i = 1, \ldots, n$, such that*

$$R(\tau_i) = 0, \qquad i = 1, \ldots, n.$$

From Theorem 5.1.2 and Corollary 5.1.3 it follows that there exists at least n distinct CG points for a space of dimension n.

5.2 Computation of CG points

This section presents a method for computing CG points numerically. The method is here given for a one-dimensional Poisson problem, but it can be shown that CG points of higher spatial dimensions are obtained as tensor products of one-dimensional points. The numerical method considered is Newton's method, which is a well known method for finding successively better approximations of roots of a real-valued function.

Using Newton's method a set of CG points are computed from a set of initial guesses $\{\tau_j^0\}_{j=1}^n$, by computing

$$\tau_j^{k+1} = \tau_j^k - \frac{\sum_{i=1}^n u_i N_i''(\tau_j^k) + f(\tau_j^k)}{\sum_{i=1}^n u_i N_i'''(\tau_j^k) + f'(\tau_j^k)}, \qquad k = 0, 1, \ldots$$

until convergence. Consider a one-dimensional Poisson problem with homogeneous Dirichlet boundary conditions of $n = 8$ degrees of freedom, with a Galerkin solution u^h corresponding to a C^2-continuous spline space $\mathcal{S}_{p,\Xi}^h$ of polynomial degree $p = 3$, defined by an uniform and open knot vector $\Xi = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. The CG points computed for this problem with seven arbitrarily chosen solutions $u(x)$ are presented in Table 5.1. Note that the difference in exact solution only affects the force $f(x)$ of the manufactured problem.

From Table 5.1 it is evident that the difference in force $f(x)$ slightly affect the position of CG-points. For the solutions examined, at most 10 CG points where found for $n = 8$, but the number of points depend on $f(x)$. As observed from Table 5.1 **c)**, 5.1 **e)** and 5.1 **f)** CG points of some problems falls in the middle of two CG points of other problems.

Table 5.1: CG points calculated by Newtons method for different solutions $u(x, y)$.

0.024899222255952	0.056335343027007	0.040187875591323	0.052726590690915
0.139188399826615	0.157044484235268	0.146234012306232	0.155809390585161
0.232484505353508	0.239988559963723	0.237883474395239	0.253644606705188
0.352110332731213	0.358457323431574	0.365781021695521	
0.437397882549075	0.444412320668358	0.500000000000000	0.427530132075481
0.552011712375973	0.558469916921334		0.572469867924518
0.639327771449647	0.643697303747818	0.634218978304479	
0.757896029809930	0.762191992599416	0.762116525604761	0.746355393294813
0.840542398191455	0.846126530890523	0.853765987693768	0.844190609414839
0.941966817586150	0.946322511751100	0.959812124408676	0.947273409309085
a) $u(x) = 5+3x^2+2x^5$	**b)** $u(x) = \cos(x)$	**c)** $u(x) = \sin(2\pi x)$	**d)** $u(x) = \sin(3\pi x)$

0.106788350482425	0.060883892868127	0.103375425740678	0.047006584886665
	0.164414539315037		0.152538165253663
0.248766571618999	0.252138606827844	0.274048731702566	0.244151054144083
0.359468584019918	0.405026750153493		0.358954315469556
0.412212444558746		0.500000000000000	0.430388194962915
0.527120726380412	0.542835605316807		0.550581565783809
0.628402727040357	0.638696717173056	0.725951268297434	0.636868699543071
0.747917364890810	0.760604021064598		0.756180221210721
0.833296952016743	0.845445524299568	0.896624574259322	0.843894667084483
0.937751496773618	0.945982760891420		0.946518186786675
e) $u(x) = \sin(x^3)$	**f)** $u(x) = \sin^2(x) + \cos^2(2x)$	**g)** $u(x) = \tan(x)$	**h) Average**

5.3 Superconvergent points

Computation of CG points as presented in previous section requires the knowledge of u_i and thus the Galerkin solution. This can obviously not be permitted when the aim is to use these points to compute the solution. In [1] and [14] an approach for obtaining estimated CG points are provided, which is based on the theory of superconvergence as presented in [36]. Thus, the name *superconvergent points* are adapted for these points. The superconvergent points for all even polynomail orders p and for the odd polynomial orders $p = 3, 5, 7$ are presented in Table 5.2.

Table 5.2: Superconvergent points (SCP) for B-splines of degree p on $[-1, 1]$.

Polynomial order	SCP
p even	-1, 0 1
$p = 3$	$\pm\frac{1}{\sqrt{3}}$
$p = 5$	$\pm\frac{\sqrt{225-30\sqrt{30}}}{15}$
$p = 7$	± 0.5049185675126533

5.4 Isogeometric collocation at Greville points (C-GP)

Isogeometric collocation at Greville points (C-GP) are the most common isogeometric collocation method in literature. Greville points are calculated as

$$\tau_i^{GP} = \frac{\xi_{i+1} + \cdots + \xi_{i+p}}{p}, \quad \forall\, i = 1, \ldots, n.$$

These points are illustrated by green circles in Figure 5.1 and Figure 5.2 for respectively odd and even polynomial orders p, and are compared with superconvergent points shown as transparent circles in the figures.

From Figure 5.2 it is observed that all Greville points for even p coincide with the superconvergent points. Thus, using Greville points is a good choice for IGC methods of even p. In the preceding sections three other choices for selecting superconvergent points for even p are presented. These four different methods yields no significant difference in accuracy for even p as is evident in the numerical results of Chapter 7.

For odd p on the other hand, the Greville points never coincide with superconvergent points as observed in Figure 5.1. In [1] and [2] it is observed that Greville points are positioned where the second derivative of the error is largest. And thus, there is reasons to believe that these points are the least optimal collocation points for solution spaces of odd p.

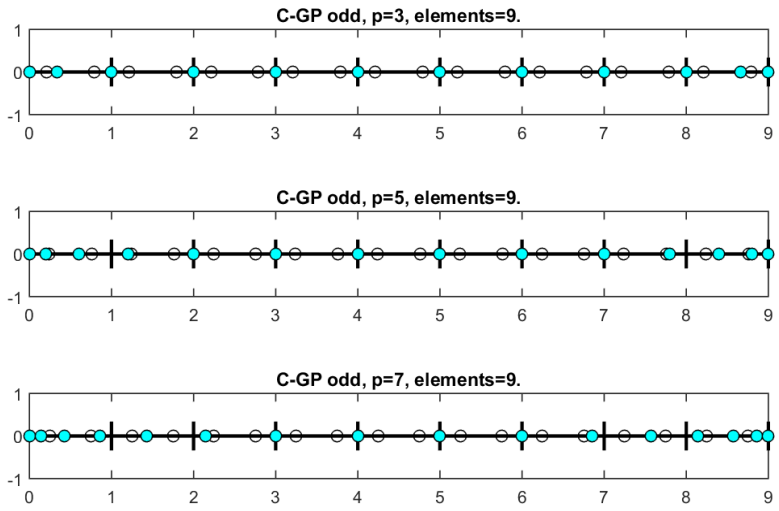


Figure 5.1: Greville points for odd p are illustrated in green. Isogeometric collocation at Greville points (C-GP) is very common in literature. Transparent circles show locations of superconvergent points. For odd p the Greville points never coincide with superconvergent points.

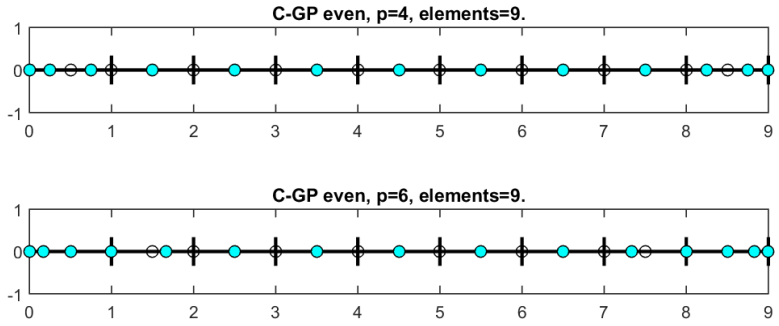


Figure 5.2: Greville points for even p are illustrated in green. Isogeometric collocation at Greville points (C-GP) is the most used collocation method. Transparent circles show locations of superconvergent points. For even p all Greville points coincide with superconvergent points.

5.5 Least-squares approximation at superconvergent points (C-LS)

In [14] Anitescu et al. proposed the first isogeometric collocation method that uses superconvergent points. This method utilizes all the superconvergent points. Since there are more superconvergent points that degrees of freedom, this method is a least-squares approximation of the overdetermined system. The method yield optimal convergence rate for odd p, but suboptimal convergence rate by one order in the L^2-norm for even p [14]. Thus, the method provides better convergence rate than collocation at Greville points for odd p, but with the disadvantage of requiring more evaluation points than degrees of freedom with increase the computational costs.

5.6 Isogeometric collocation at alternating superconvergent points (C-ASP)

In [1] Gomez et al. states that any subset of superconvergent points will produce the Galerking solution when the support of each basis function contain at least one point. A natural first choice is to select points spread on the whole domain. This can be done by choosing one point in each element and place any extra points near the middle or the boundary. This will in effect be to select every other superconvergent point and is therefor called IGC at alternating superconvergent points (C-ASP). Extra points are required when the number of freedom is higher than the number of elements. This method was first presented in [1] and later in [2].

Placement of points for a odd number of elements when p is odd is presented in Figure 5.3 and when p is even is presented Figure 5.4. For an even number of elements the selection becomes a little different but the main idea is the same. This is presented for odd p in Figure 5.5 and for even p in Figure 5.6. The green points in the figures are the selected superconvergent points while the remaining superconvergent points are shown as transparent.

This method yield one order suboptimal convergence rate for both odd and even p in L^2-norm but is otherwise optimal. This is evident in the numerical results presented in Chapter 7.

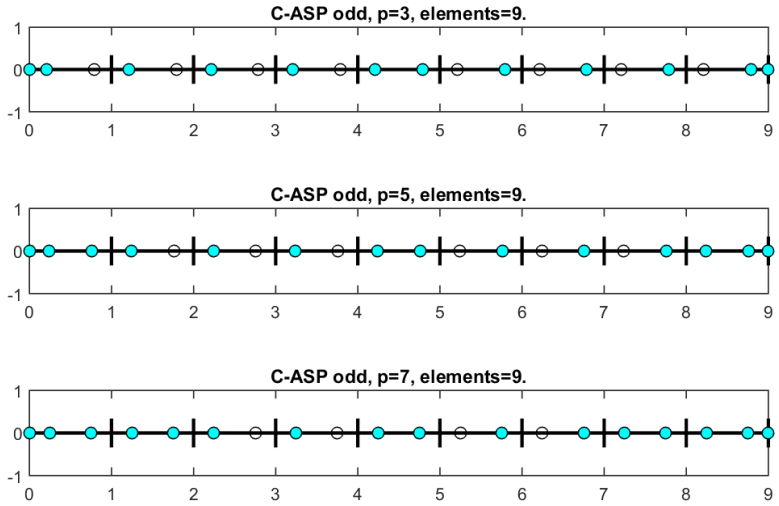


Figure 5.3: Alternating superconvergent points for odd p on a odd number of elements are shown in green. These points are selected for collocation at alternating superconvergent points method (C-ASP). Transparent circles illustrate locations of the remaining superconvergent points.

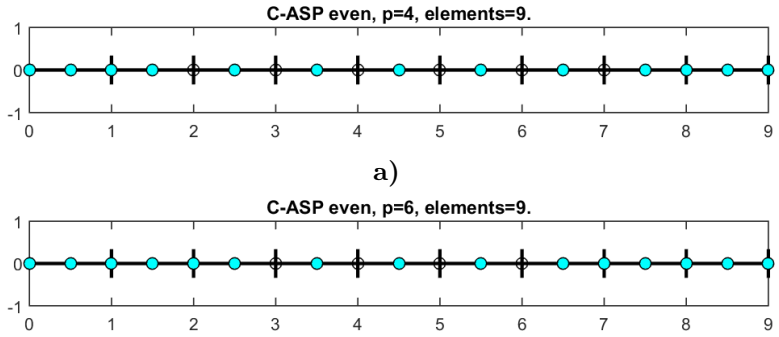


Figure 5.4: Alternating superconvergent points for even p on a odd number of elements are shown in green. These points are selected for collocation at alternating superconvergent points method (C-ASP). Transparent circles illustrate locations of the remaining superconvergent points.

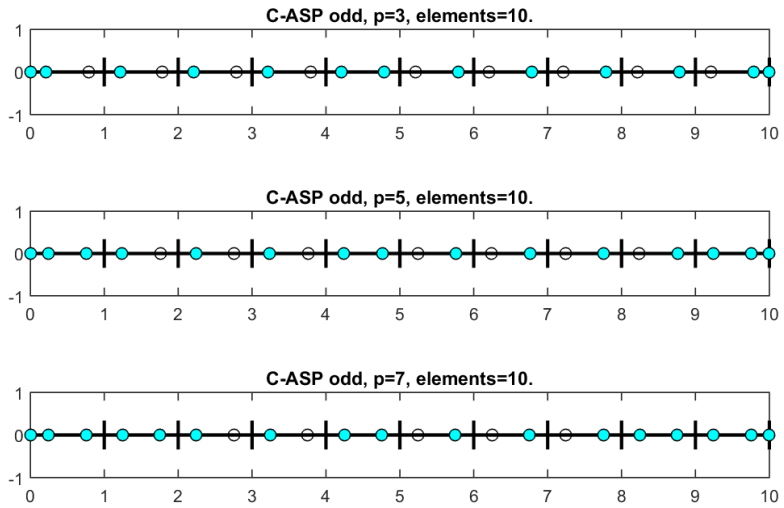


Figure 5.5: Alternating superconvergent points for odd p on a even number of elements are shown in green. These points are selected for collocation at alternating superconvergent points method (C-ASP). Transparent circles illustrate locations of the remaining superconvergent points.

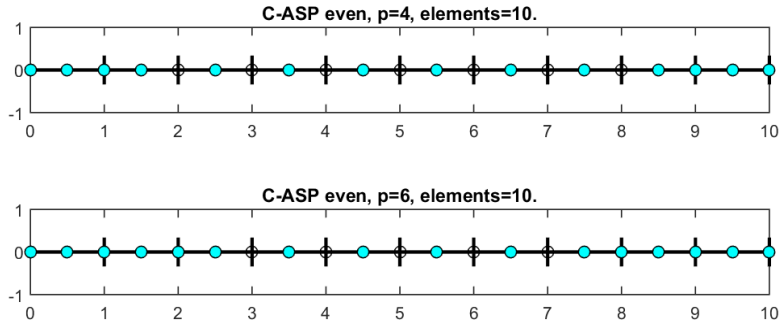


Figure 5.6: Alternating superconvergent points for even p on a even number of elements are shown in green. These points are selected for collocation at alternating superconvergent points method (C-ASP). Transparent circles illustrate locations of the remaining superconvergent points.

5.7 Isogeometric collocation at clustered superconvergent points (C-CSP)

Isogeometric collocation at clustered superconvergent points (C-CSP) selects in general the two superconvergent point of every second element. This approach provides a periodic distribution of collocation points and contain the symmetry of superconvergent points in each element.

Placement of points for a odd number of elements when p is odd is presented in Figure 5.7 and when p is even is presented Figure 5.8. For an even number of elements the selection becomes a little different but the main idea is the same. This is presented for odd p in Figure 5.9 and for even p in Figure 5.10. The green points in the figures are the selected superconvergent points while the remaining superconvergent points are shown as transparent.

This method yield optimal order when p is odd for all considered norms. Thus, for odd p this method achieves the same orders as IGA methods with full quadrature but require just one evaluation point per degree of freedom. For even p however the method is still suboptimal by one order in the L^2-norm. This is also evident in the numerical results presented in Chapter 7.

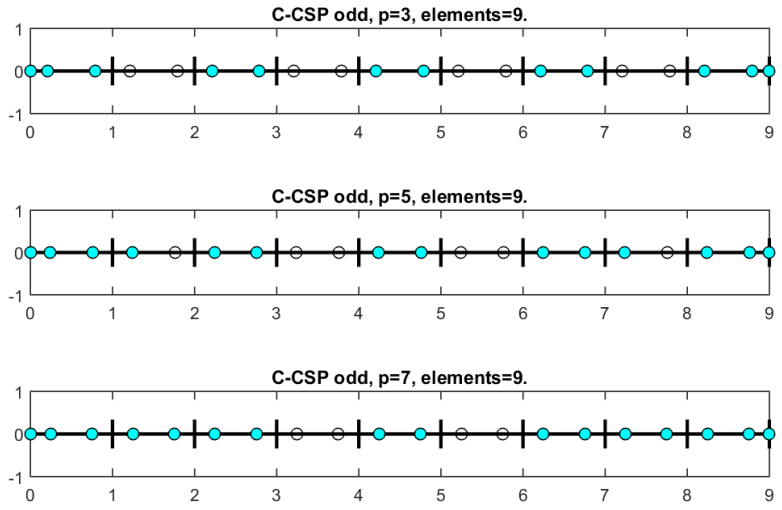


Figure 5.7: Clustered superconvergent points for odd p on a odd number of elements are shown in green. These points are selected for collocation at clustered superconvergent points method (C-ASP). Transparent circles illustrate locations of the remaining superconvergent points.

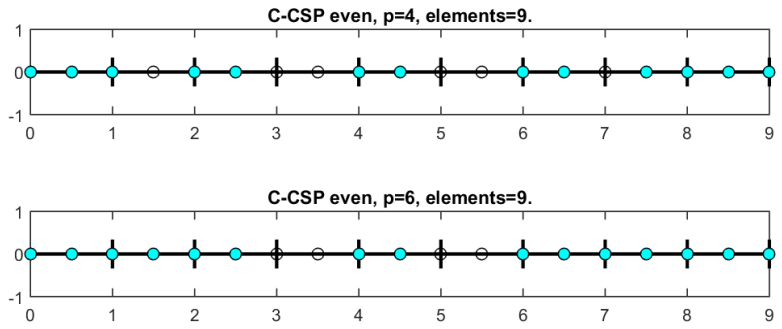


Figure 5.8: Clustered superconvergent points for even p on a odd number of elements are shown in green. These points are selected for collocation at clustered superconvergent points method (C-ASP). Transparent circles illustrate locations of the remaining superconvergent points.

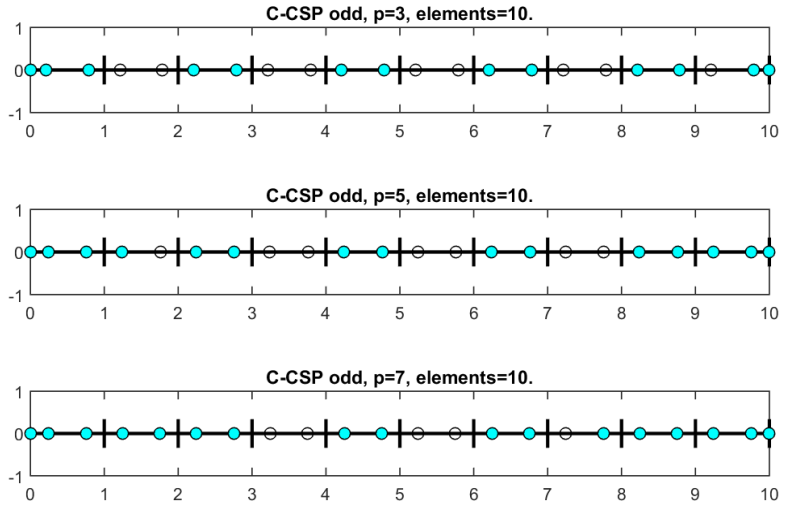


Figure 5.9: Clustered superconvergent points for odd p on a even number of elements are shown in green. These points are selected for collocation at clustered superconvergent points method (C-ASP). Transparent circles illustrate locations of the remaining superconvergent points.

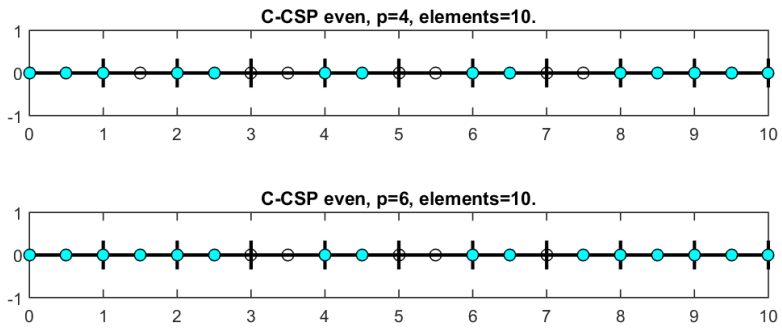


Figure 5.10: Clustered superconvergent points for even p on a even number of elements are shown in green. These points are selected for collocation at clustered superconvergent points method (C-ASP). Transparent circles illustrate locations of the remaining superconvergent points.

CHAPTER 6

Weighted residual at superconvergent points

In [1] Gomez et al. proposes using superconvergent points as quadrature points. This is implemented for the weighted residual method and results are presented in 8. In the method all superconvergent points are used as quadrature points, so there are in general two evaluation points for each element. The weigts used are the same weights used in Gaussian quadrature.

CHAPTER 7

Numerical results

This chapter presents numerical results of the three isogeometric collocation (IGC) methods presented in this thesis, and compare them with the well-established isogeometric analysis (IGA) of weak variational form and IGA of strong variational form. The results are shown for five geometrical shapes that all persist an uniform knot vector after h-refinement in physical space. The three isogeometric collocation methods are IGC with the traditional Greville points (C-GP), with alternating superconvergent points (C-ASP) and with clustered superconvergent points (C-CSP). As presented in 5 these methods differ only in the placement of their collocation points.

The problems are generated from predefined manufactured solutions, and the error and convergence rates are measured in the L^2-norm, H^1-seminorm and the H^2-seminorm. All problems considered in this thesis consistently results in the convergence rates presented in Table 7.1, for all five methods.

Table 7.1: Convergence rates for five methods.

Method	L^2	H^1	H^2
Weak GP	$p+1$	p	$p-1$
Strong GP	$p+1$	p	$p-1$
C-CSP odd	$\mathbf{p+1}$	p	$p-1$
C-CSP even	p	p	$p-1$
C-ASP odd	p	p	$p-1$
C-ASP even	p	p	$p-1$
C-GP odd	$\mathbf{p-1}$	$p-1$	$p-1$
C-GP even	p	p	$p-1$

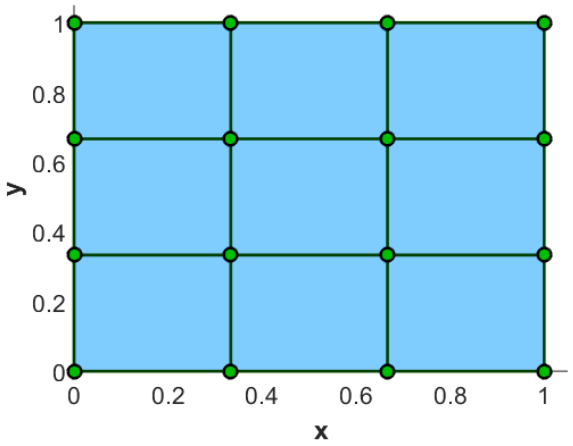
From Table 7.1 it is observed that all methods have optimal convergence rate of $p-1$ in the H^2-seminorm and that all have optimal convergence rate of p in the H^1-seminorm, except of C-GP for odd p. It is further shown that only odd C-CSP achieves the optimal convergence rate of $p+1$ in the L^2-norm, while the other IGC methods are one order lower, except for C-GP which are two orders under the optimal. In the following sections each problem is presented with its geometry and numerical results for all five methods with polynomial order p from 3 up to 8.

7.1 Rectangle

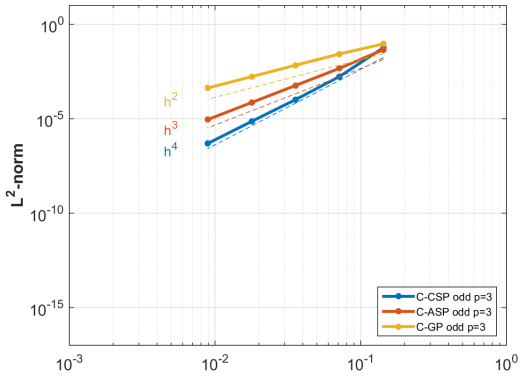
This first problem to approximate a smooth manufactured solution on the simplest two-dimensional geometry, that is a rectangle where the coarsest possible mesh of one element is shown in Figure 7.1 **a)** for polynomial order $p = 3$. The green points illustrates control points that define the green control polygon. The smooth solution to the manufactured problem is

$$u(x, y) = \sin(2\pi(x^2 + y^2)).$$

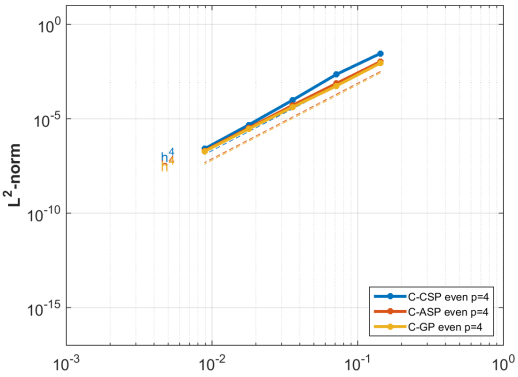
This problem demonstrate the convergence rate obtained for a smooth problem alone without any consideration of the geometrical structure. Figure 7.1 **b)** and 7.1 **c)** compares the three isogeometric collocation (IGC) methods for polynomial order $p = 3$ (odd) and $p = 4$ (even) in the L^2-norm. The three methods are IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP), where the only differences are the placement of collocation points as explained in 5.



a) Rectangle: Blue geometry illustrates the coarsest mesh comprising one element with green control points.



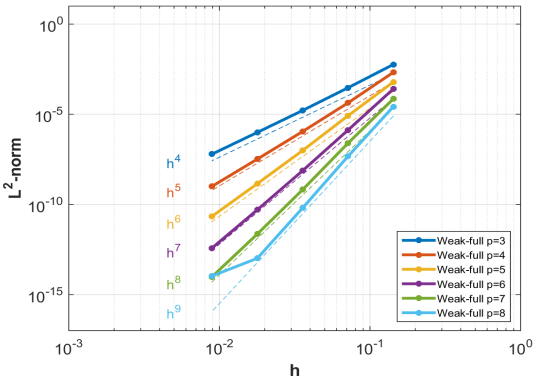
b) Rect.: Comparison for odd p=3.



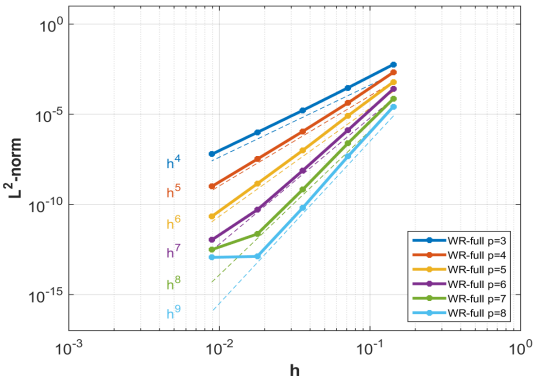
c) Rect.: Comparison for even p=4.

Figure 7.1: Rectangle: Geometry of coarsest mesh are presented in **a)**. In **b)** isogeometric collocation (IGC) at Greville points(C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP) are compared for odd p in the L^2-norm for a smooth manufactured solution. In **c)** the same are plotted for even p.

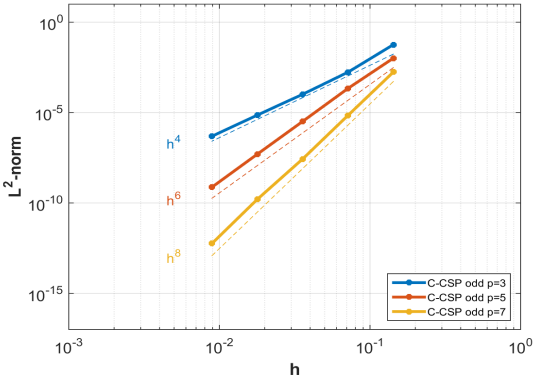
The following three pages shows numerical results from all five methods for p from 3 up to 8. Results for the three IGC methods are split in odd and even p because of the difference in convergence rate. Figure 7.2 presents results in L^2-norm, 7.3 in H^1-seminorm and 7.4 in H^2-seminorm. Under each subfigure are the observed convergence rate for each method stated. All observed convergence rates coincide Table 7.1.



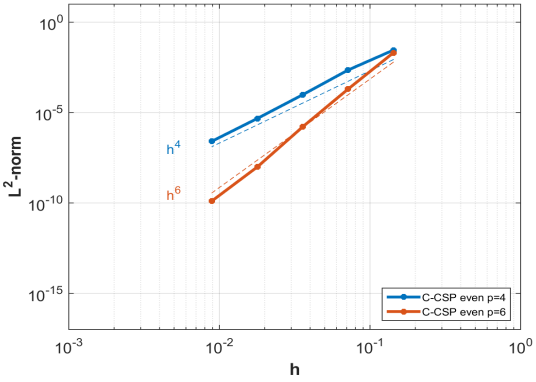
a) Weak-full: rate $p+1$.



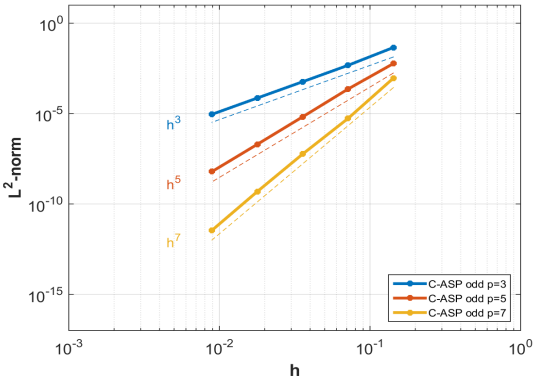
b) WR-full: rate $p+1$.



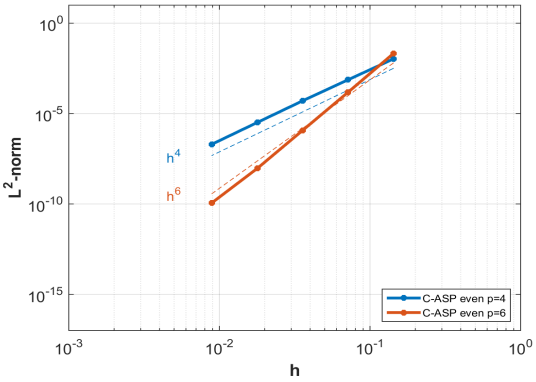
c) C-CSP odd: rate $p+1$.



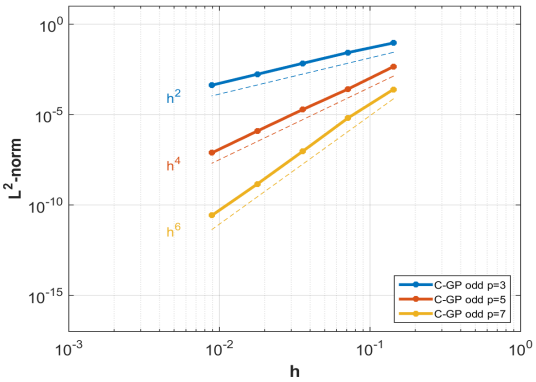
d) C-CSP even: rate p.



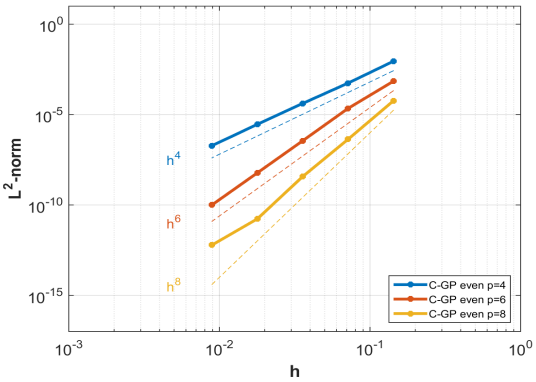
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

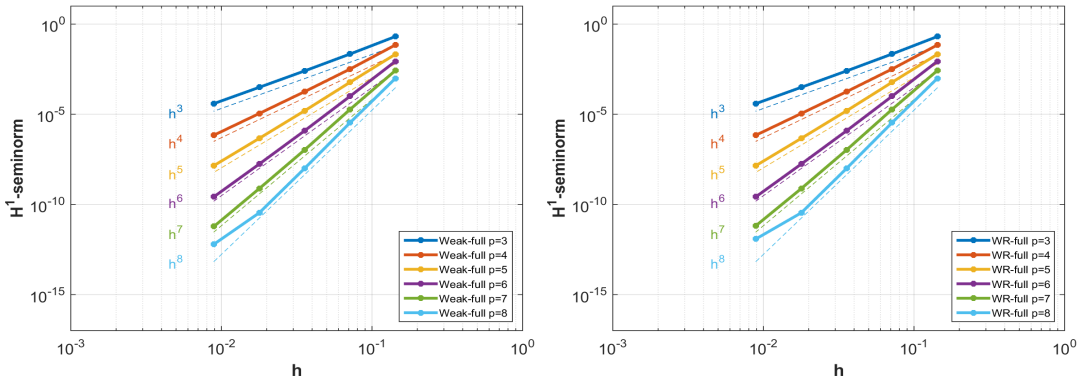


g) C-GP odd: rate $p-1$.



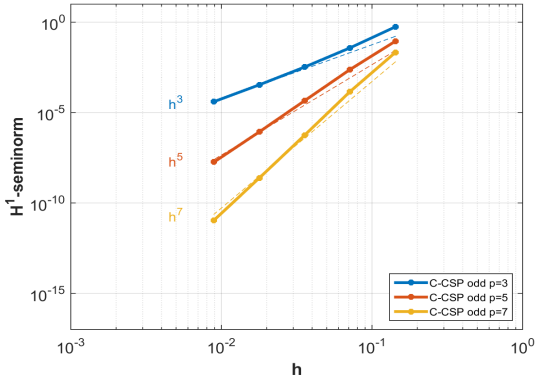
h) C-GP even: rate p.

Figure 7.2: Rectangle, L^2-norm: Convergence plot of L^2 norm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.

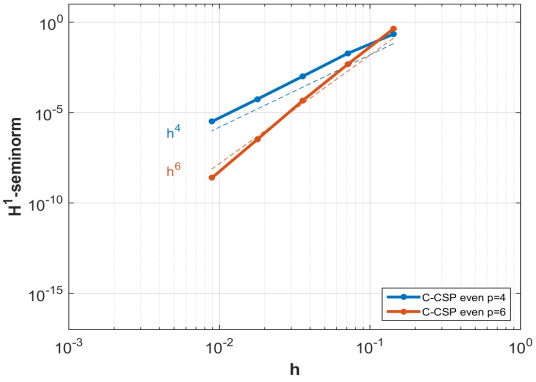


a) Weak-full: rate p.

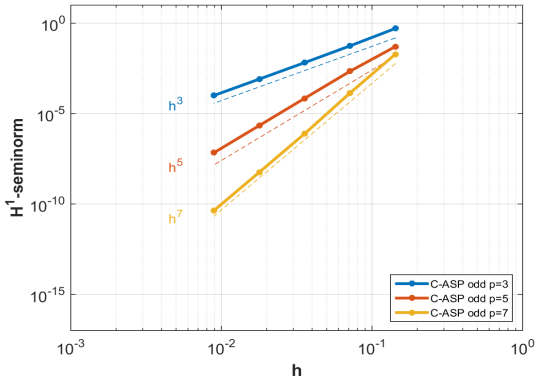
b) WR-full: rate p.



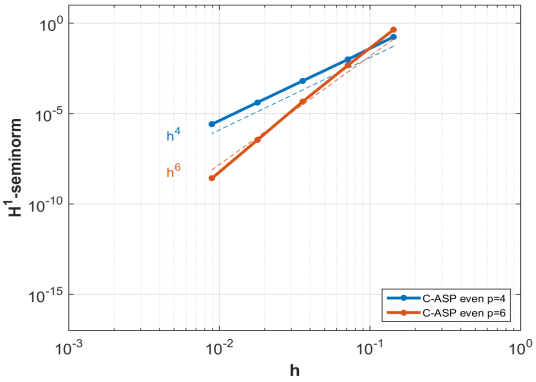
c) C-CSP odd: rate p.



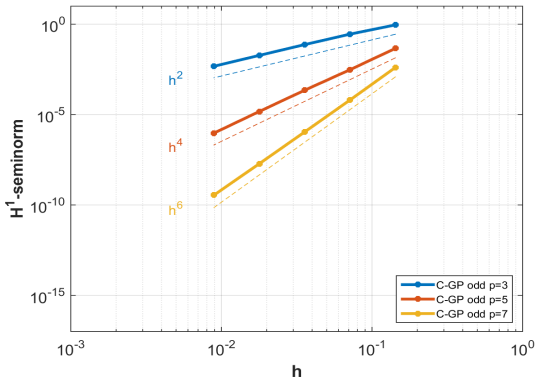
d) C-CSP even: rate p.



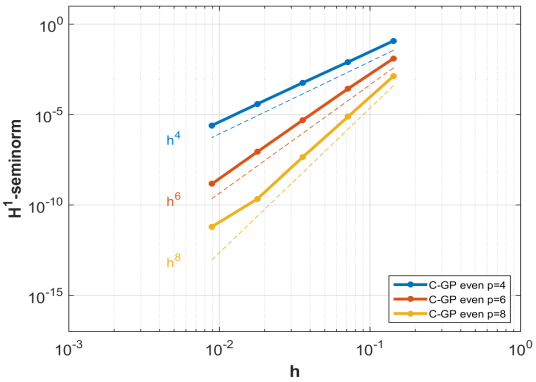
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

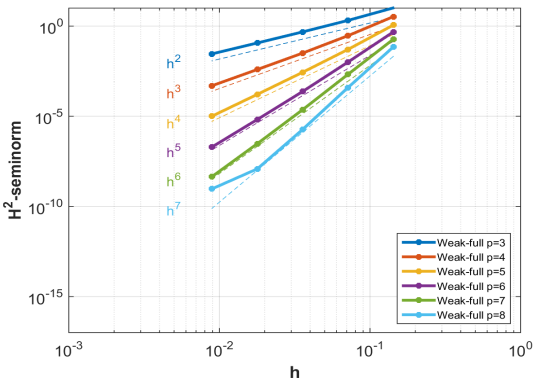


g) C-GP odd: rate $p-1$.

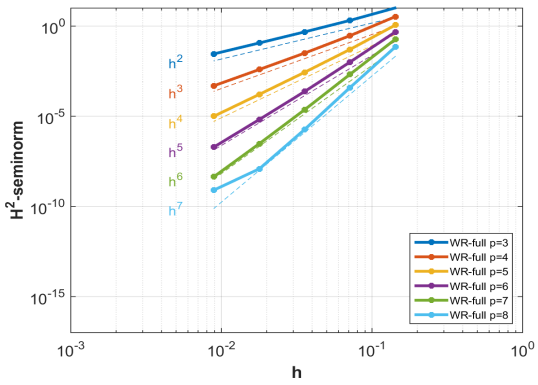


h) C-GP even: rate p.

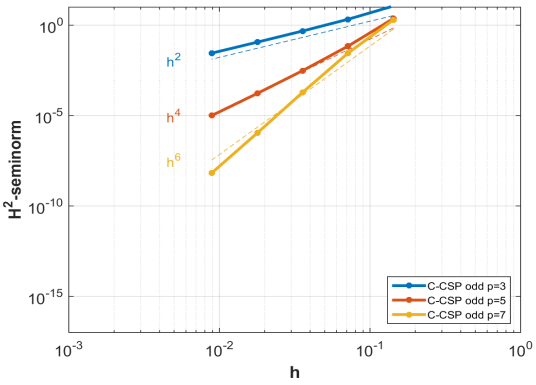
Figure 7.3: Rectangle, H^1-seminorm: Convergence plot of H^1-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.



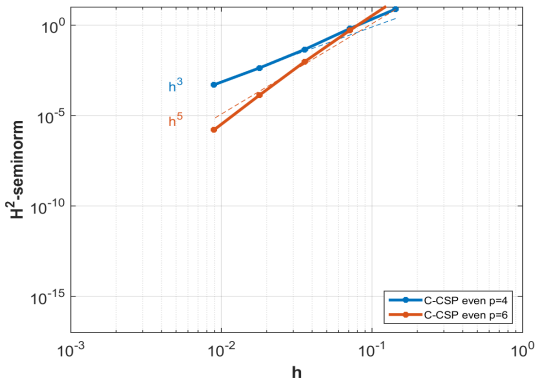
a) Weak-full: rate $p-1$.



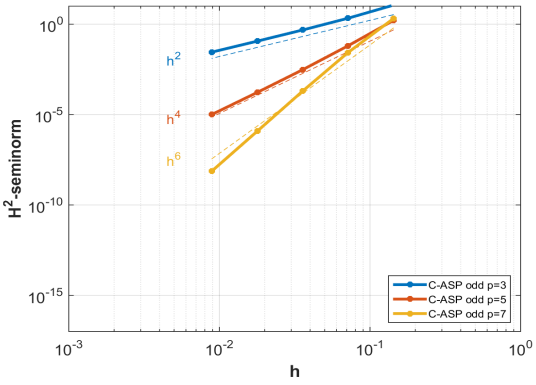
b) WR-full: rate $p-1$.



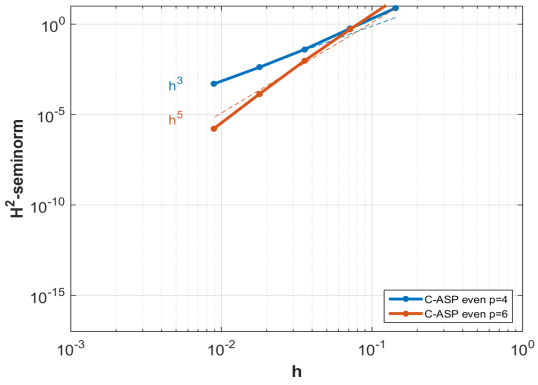
c) C-CSP odd: rate $p-1$.



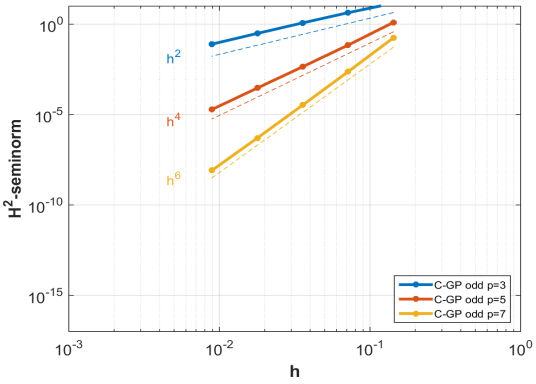
d) C-CSP even: rate $p-1$.



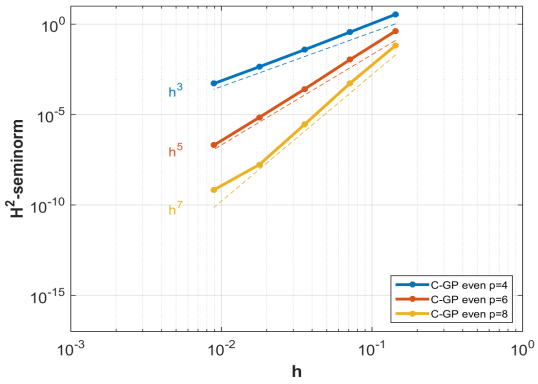
e) C-ASP odd: rate $p-1$.



f) C-ASP even: rate $p-1$.



g) C-GP odd: rate $p-1$.



h) C-GP even: rate $p-1$.

Figure 7.4: Rectangle, H^2-seminorm: Convergence plot of H^2-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.

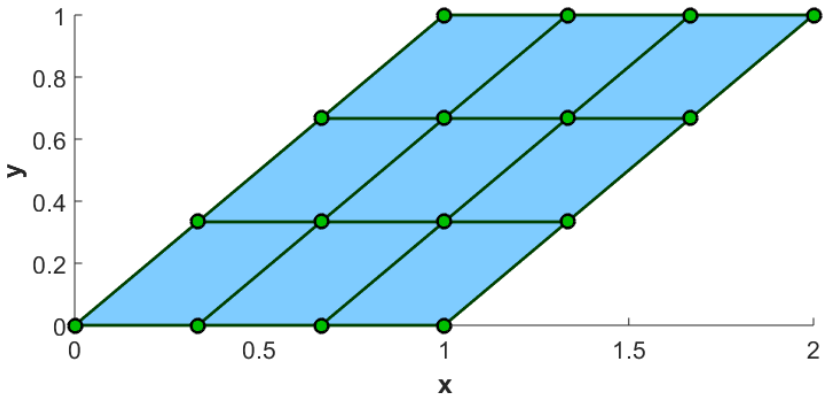
7.2 Parallelogram

This second problem is a polynomial manufactured solution on a geometry with affine mapping. The geometry is a parallelogram where the coarsest possible mesh of one element is shown in Figure 7.5 **a)** for polynomial order $p = 3$. The green points illustrates control points which define the green control polygon. The affine mapping should not affect the resulting convergence rate much. The chosen manufactured solution are used in among others [2] and has the form

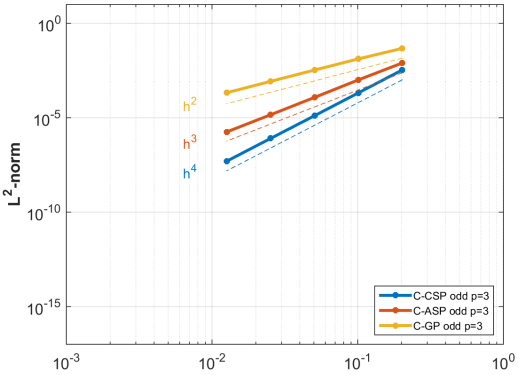
$$u(x, y) = -(x^2 + y^2 - 1)(x^2 + y^2 - 4)xy^2.$$

This problem is chosen to prove the exactness of the methods. When the solution space of a method contains the exact solution the error should consist of only machine error. All solution spaces of the methods considered in this thesis are polynomial and thus all methods with a polynomial order $p \leq 7$ should result in the exact solution. It is evident from all plots in this section that this is the case.

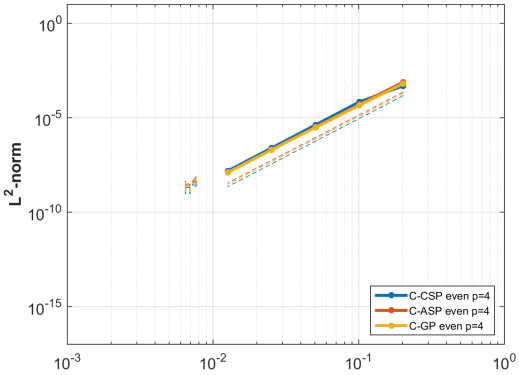
Figure 7.5 **b)** and 7.5 **c)** compares the three isogeometric collocation (IGC) methods for polynomial order $p = 3$ (odd) and $p = 4$ (even) in the L^2-norm. The three methods are IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP), where the only differences are the placement of collocation points as explained in 5.



a) Parallelogram: Blue geometry illustrates the coarsest mesh comprising one element with green control points..



b) Parall.: Comparison for odd p=3.

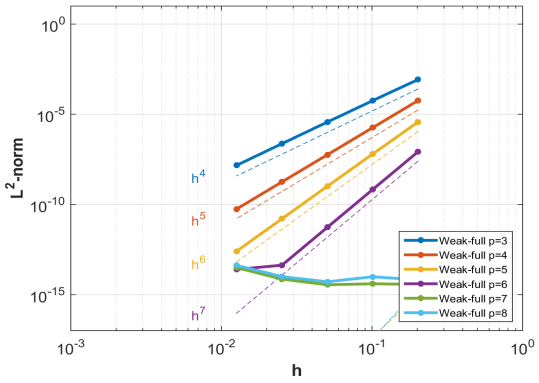


c) Parall.: Comparison for even p=4.

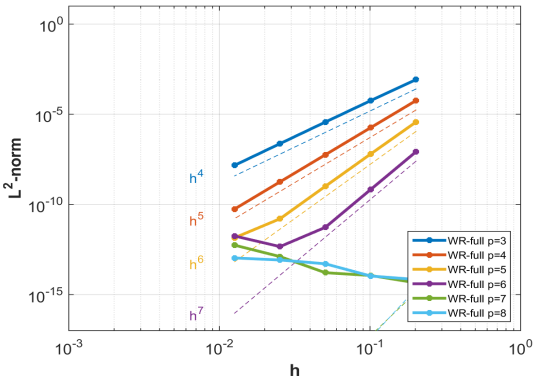
Figure 7.5: Parallelogram: Geometry of coarsest mesh are presented in **a)**. In **b)** isogeometric collocation (IGC) at Greville points(C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP) are compared for odd p in the L^2-norm with a polynomial manufactured solution. In **c)** the same are plotted for even p.

The following three pages shows numerical results from all five methods for p from 3 up to 8. The three IGC methods are split in odd and even p due to the difference in convergence rate. Figure 7.6 presents results in L^2-norm, 7.7 in H^1-seminorm and 7.8 in H^2-seminorm. Under each subfigure are the observed convergence rate for each method stated. All observed convergence rates coincide Table 7.1 except for methods of polynomial order $p \leq 7$ which are exact in the order of machine error.

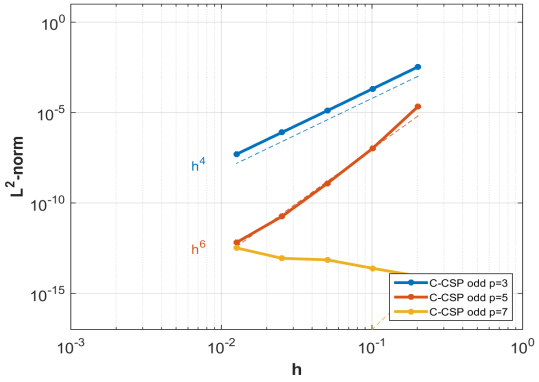
When the approximate solution is exact it might be observed that the error increases with refinement. This is because When the mesh is refined higher error occurs due to rounding error in an increased number of floating point operations.



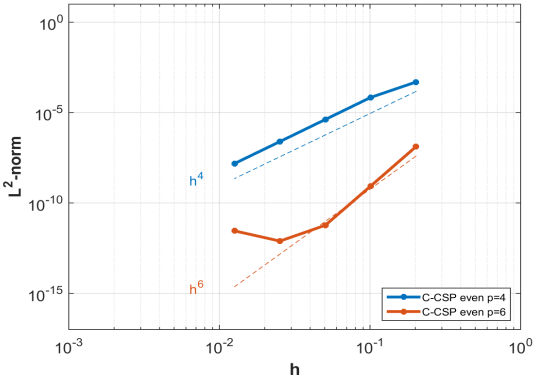
a) Weak-full: rate $p+1$.



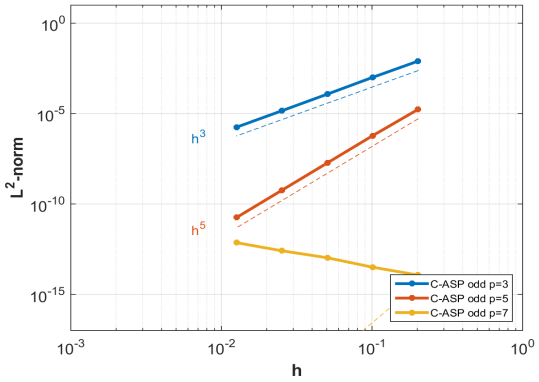
b) WR-full: rate $p+1$.



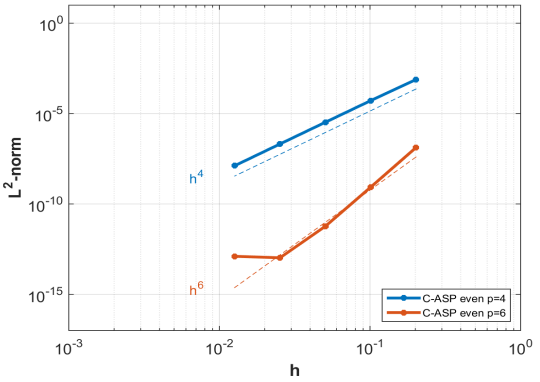
c) C-CSP odd: rate $p+1$.



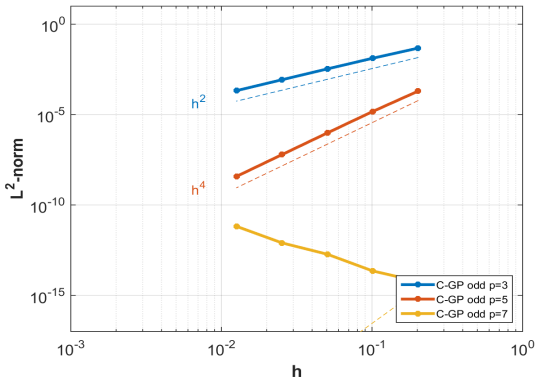
d) C-CSP even: rate p.



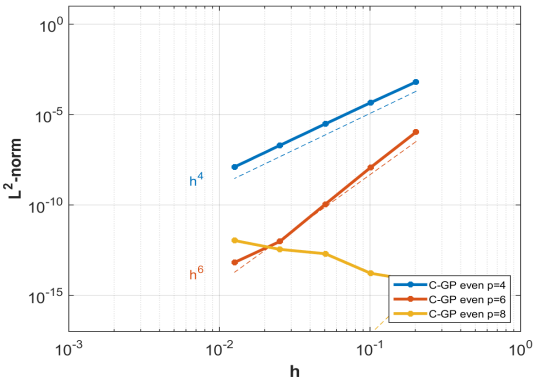
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

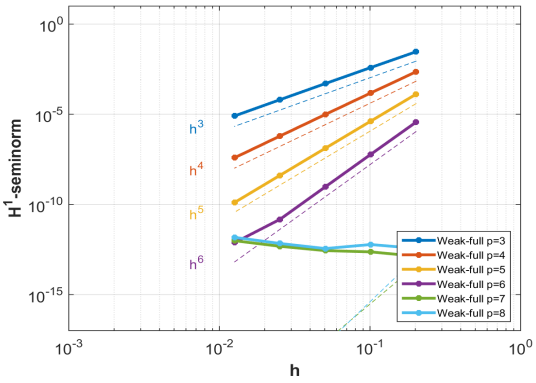


g) C-GP odd: rate $p-1$.

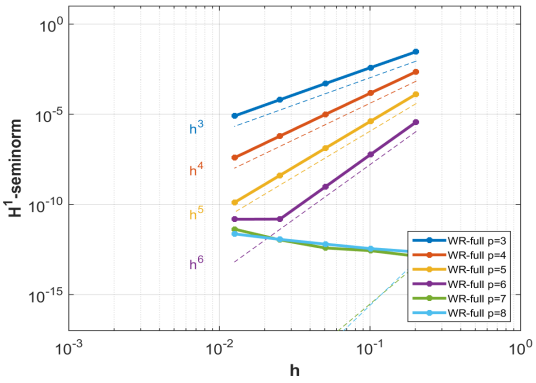


h) C-GP even: rate p.

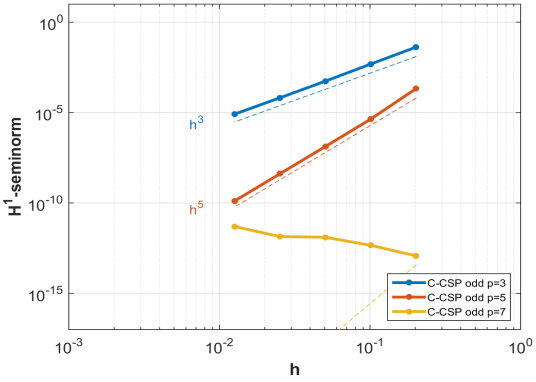
Figure 7.6: Parallelogram, L^2-norm: Convergence plot of L^2 norm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and polynomial solution. Convergence rates are written under respective subfigures.



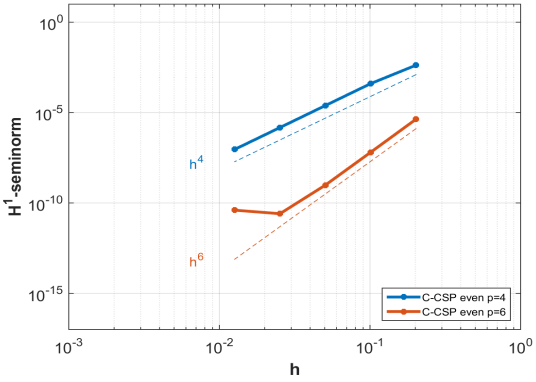
a) Weak-full: rate p.



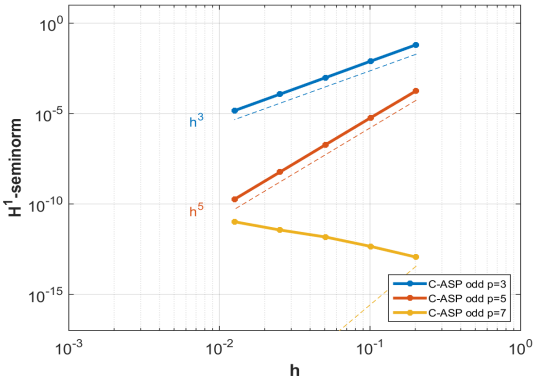
b) WR-full: rate p.



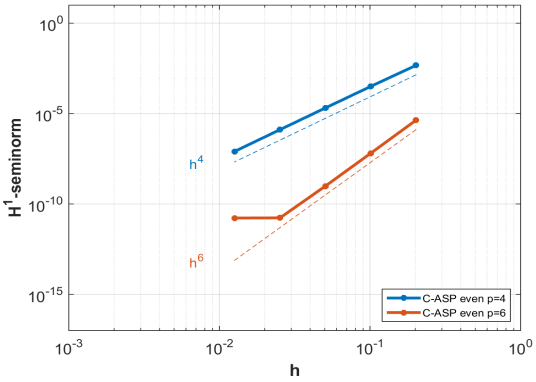
c) C-CSP odd: rate p.



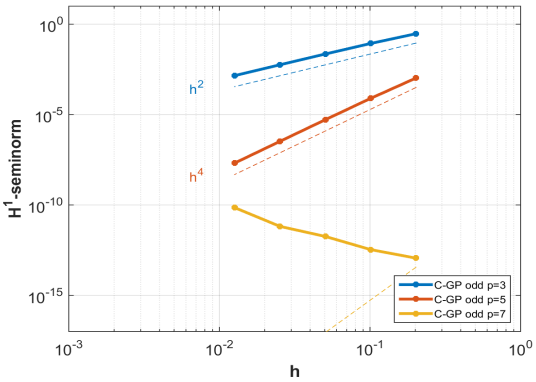
d) C-CSP even: rate p.



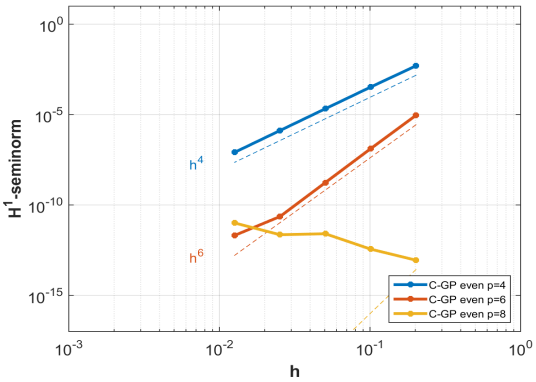
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

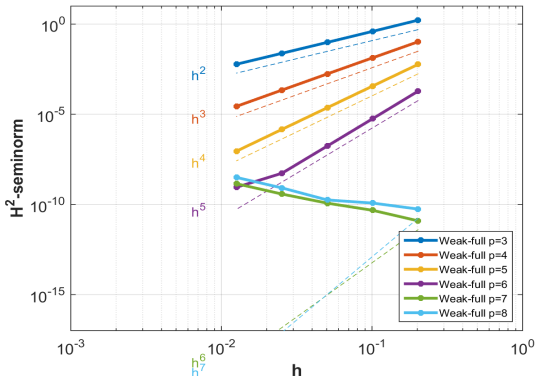


g) C-GP odd: rate $p-1$.

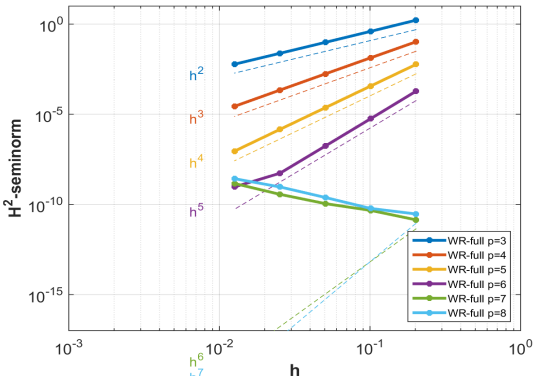


h) C-GP even: rate p.

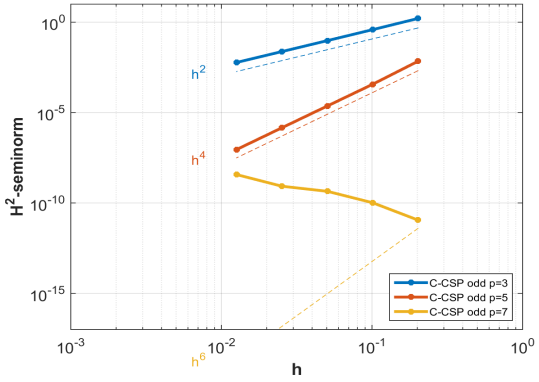
Figure 7.7: Parallelogram, H^1-seminorm: Convergence plot of H^1-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and polynomial solution. Convergence rates are written under respective subfigures.



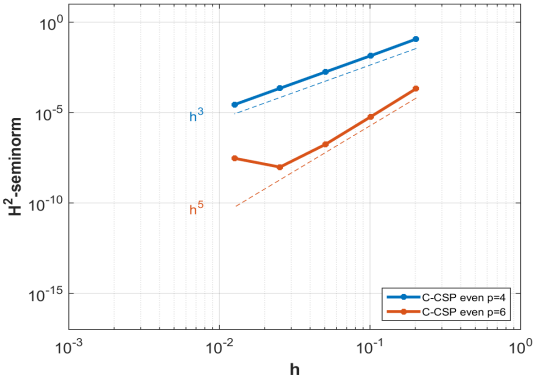
a) Weak-full: rate $p-1$.



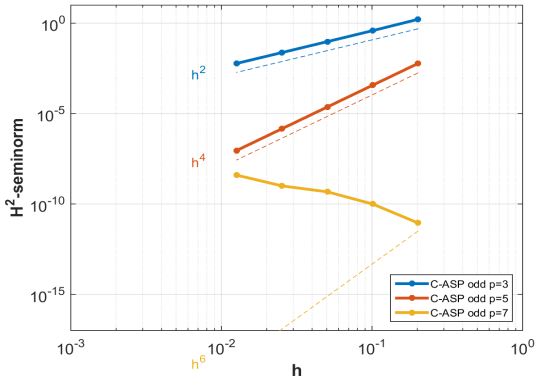
b) WR-full: rate $p-1$.



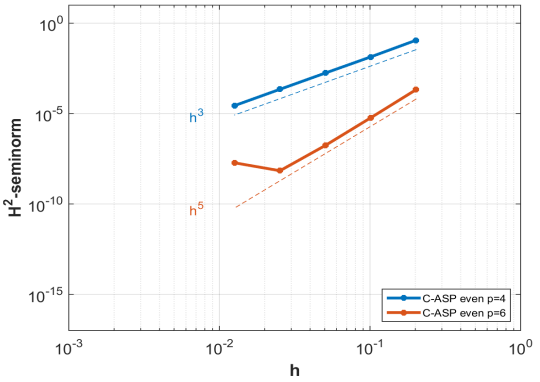
c) C-CSP odd: rate $p-1$.



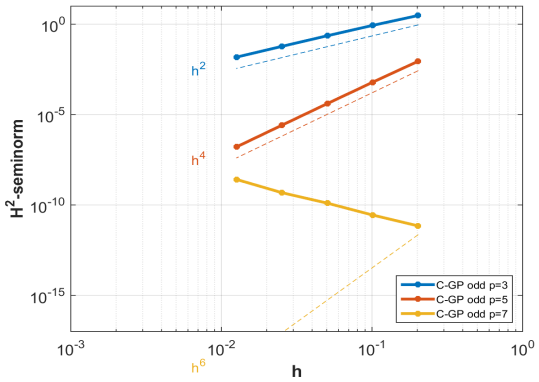
d) C-CSP even: rate $p-1$.



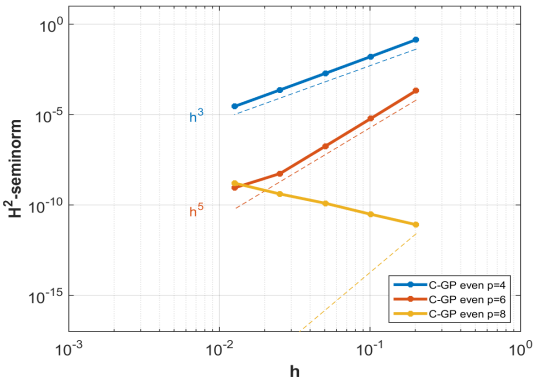
e) C-ASP odd: rate $p-1$.



f) C-ASP even: rate $p-1$.



g) C-GP odd: rate $p-1$.



h) C-GP even: rate $p-1$.

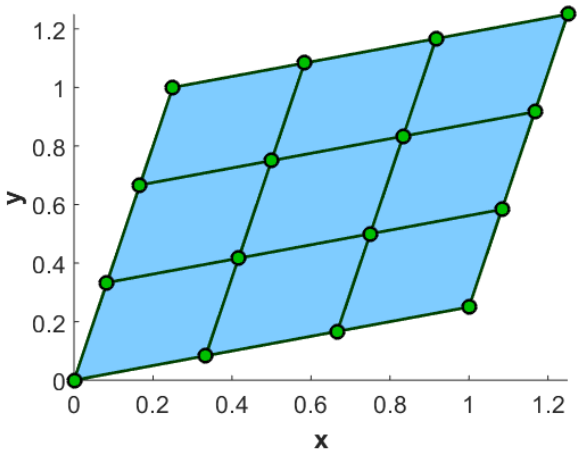
Figure 7.8: Parallelogram, H^2-seminorm: Convergence plot of H^2-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and polynomial solution. Convergence rates are written under respective subfigures.

7.3 Rhombus

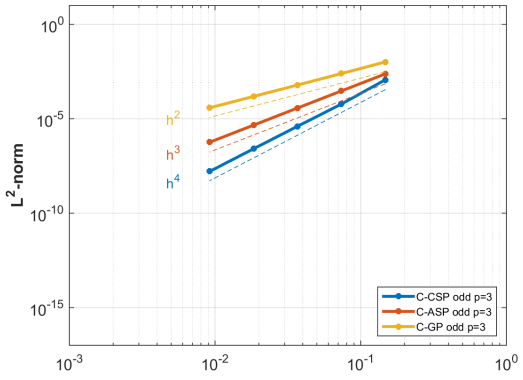
This third problem is a smooth manufactured solution on another affine mesh, that is, a rhombus with corners in (0,0), (1,0.25), (0.25,1) and (1.25,1.25). The coarsest possible mesh of one element is shown in Figure 7.9 **a)** for polynomial order $p = 3$. The green points illustrates control points that define the green control polygon. The smooth manufactured solution chosen are used in among others [2] an has the form

$$u(x, y) = \sin(\frac{4}{15}\pi(y - 4x)) \sin(\frac{16}{15}\pi(\frac{x}{4} - y))(x^3 + y^3).$$

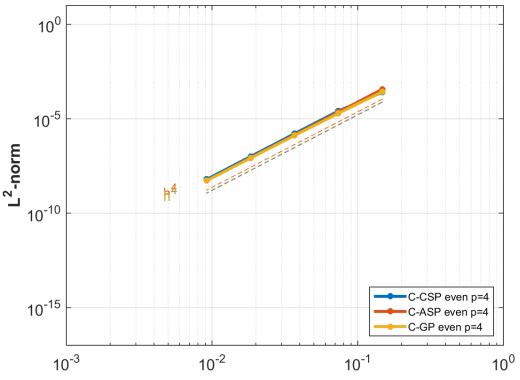
Figure 7.9 **b)** and 7.9 **c)** compares the three isogeometric collocation (IGC) methods for polynomial order $p = 3$ (odd) and $p = 4$ (even) in the L^2-norm. The three methods are IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP), where the only differences are the placement of collocation points as explained in 5.



a) Rhombus: Blue geometry illustrates the coarsest mesh comprising one element with green control points.



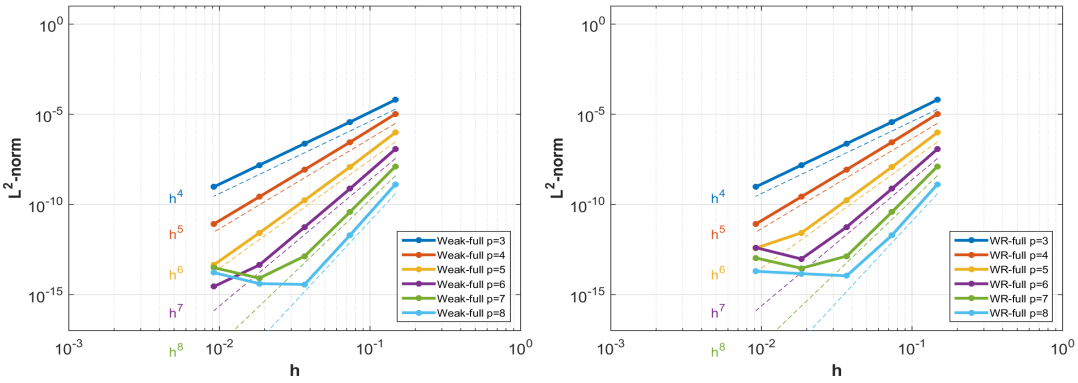
b) Rhombus: Comparison for odd p=3.



c) Rhombus: Comparison for even p=4.

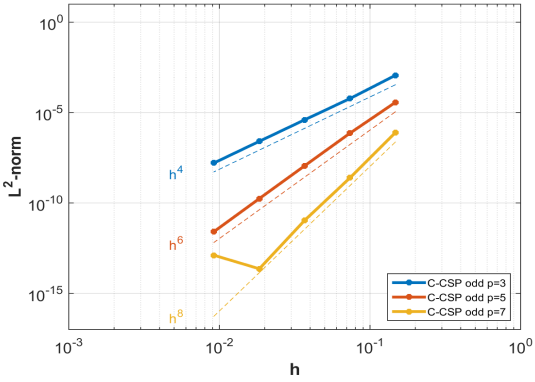
Figure 7.9: Rhombus: Geometry of coarsest mesh are presented in **a)**. In **b)** isogeometric collocation (IGC) at Greville points(C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP) are compared for odd p in the L^2-norm with smooth manufactured solution. In **c)** the same are plotted for even p.

The following three pages shows numerical results from all five methods for p from 3 up to 8. Results for the three IGC methods are split in odd and even p because of the difference in convergence rate. Figure 7.10 presents results in L^2-norm, 7.11 in H^1-seminorm and 7.12 in H^2-seminorm. Under each subfigure are the observed convergence rate for each method stated. All observed convergence rates coincide Table 7.1.

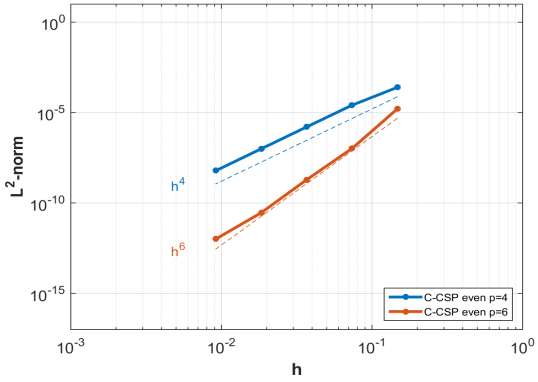


a) Weak-full: rate $p+1$.

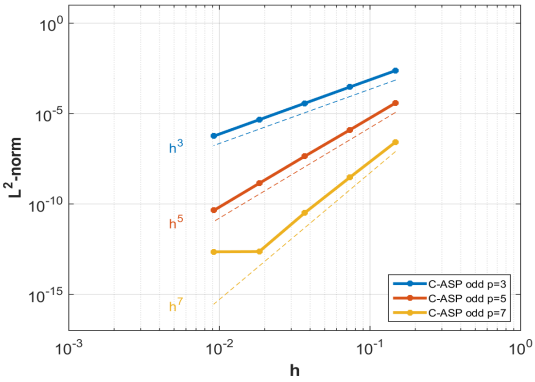
b) WR-full: rate $p+1$.



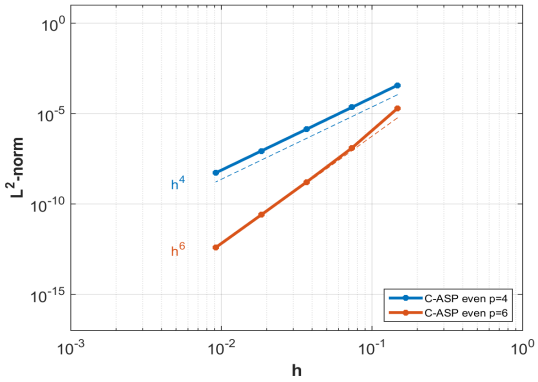
c) C-CSP odd: rate $p+1$.



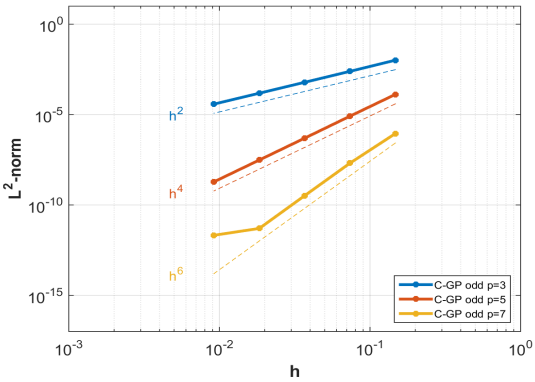
d) C-CSP even: rate p.



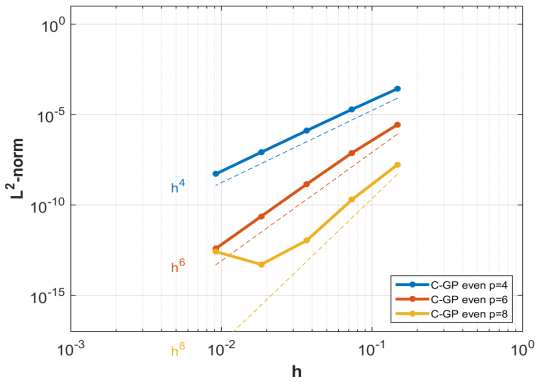
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

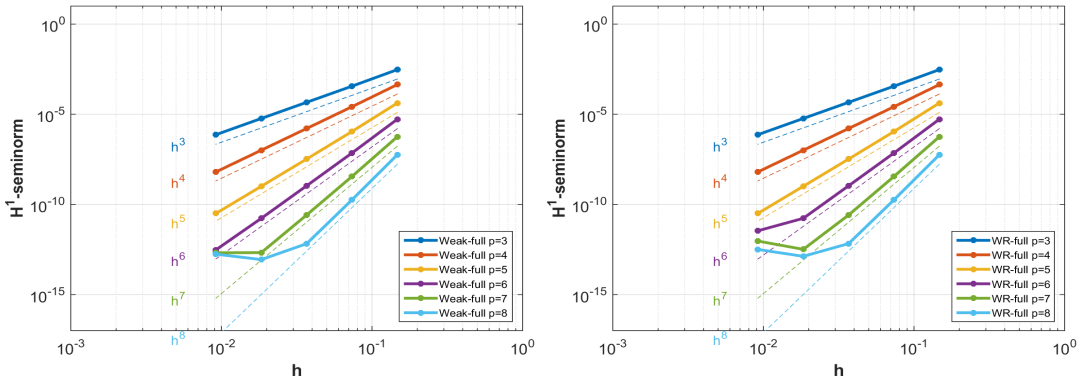


g) C-GP odd: rate $p-1$.



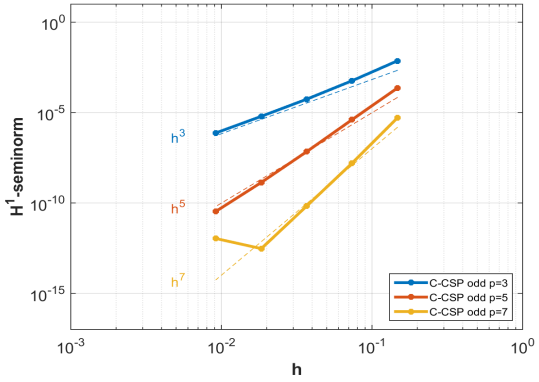
h) C-GP even: rate p.

Figure 7.10: Rhombus, L^2-norm: Convergence plot of L^2 norm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.

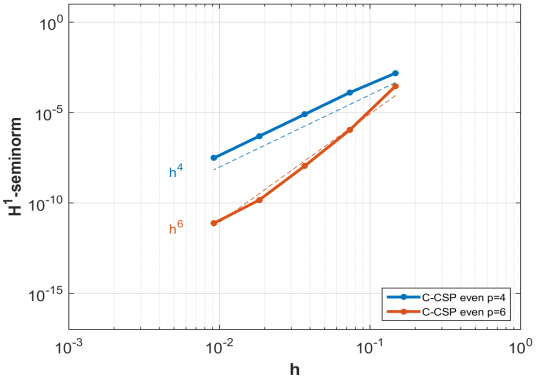


a) Weak-full: rate p.

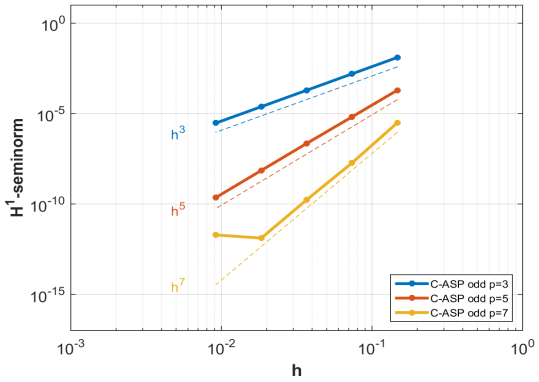
b) WR-full: rate p.



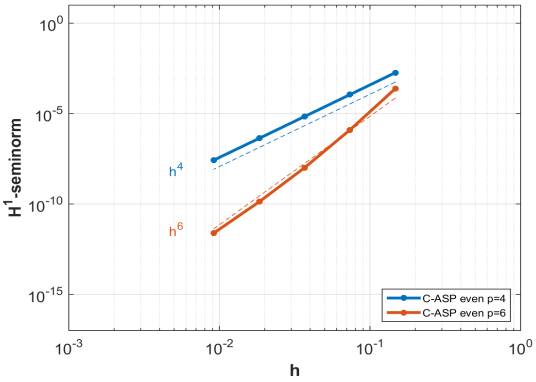
c) C-CSP odd: rate p.



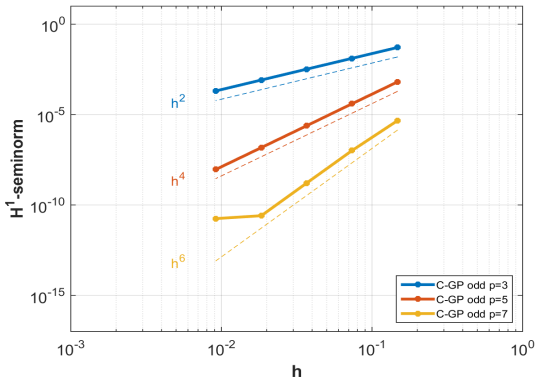
d) C-CSP even: rate p.



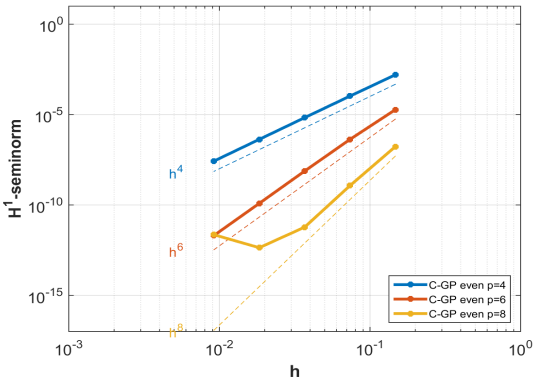
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

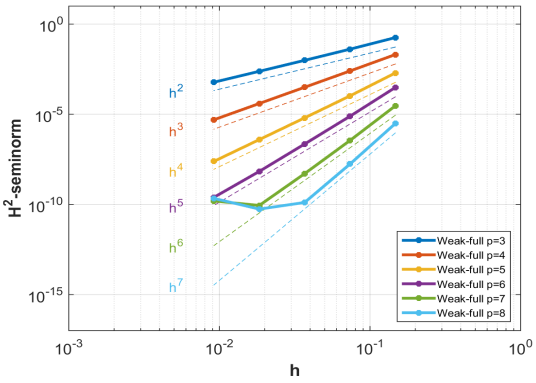


g) C-GP odd: rate $p-1$.

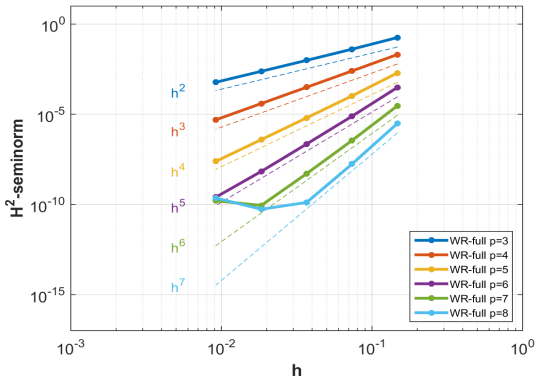


h) C-GP even: rate p.

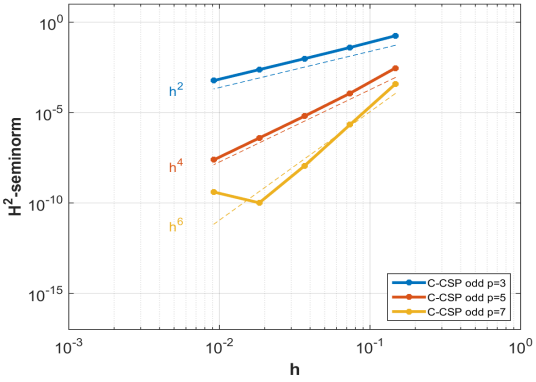
Figure 7.11: Rhombus, H^1-seminorm: Convergence plot of H^1-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.



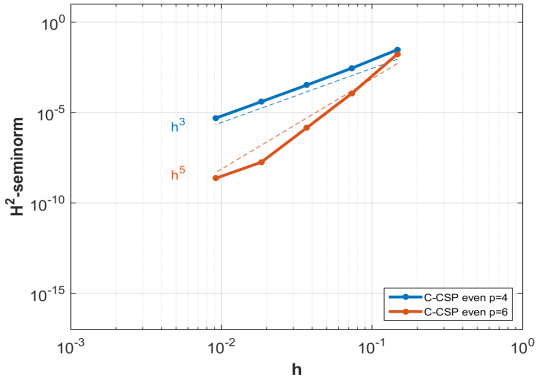
a) Weak-full: rate $p-1$.



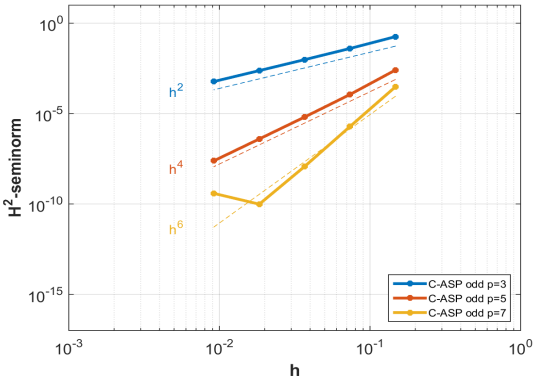
b) WR-full: rate $p-1$.



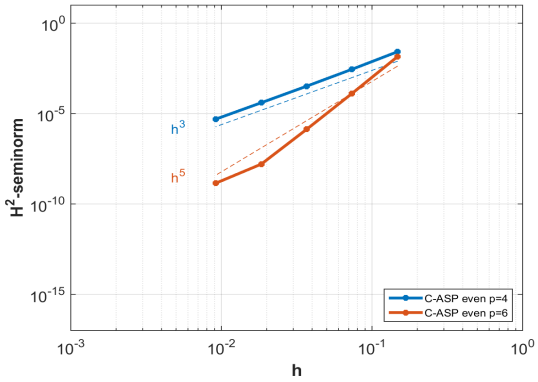
c) C-CSP odd: rate $p-1$.



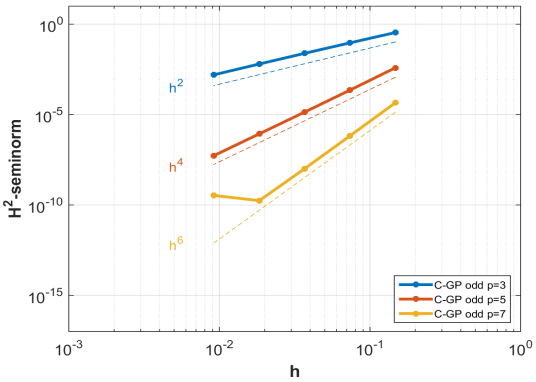
d) C-CSP even: rate $p-1$.



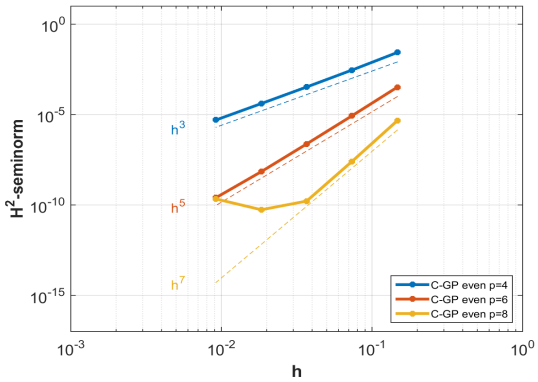
e) C-ASP odd: rate $p-1$.



f) C-ASP even: rate $p-1$.



g) C-GP odd: rate $p-1$.



h) C-GP even: rate $p-1$.

Figure 7.12: Rhombus, H^2-seminorm: Convergence plot of H^2-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.

7.4 Moderate trapezoid

The third problem considered is the trapezoid where the coarsest possible mesh of one element is shown in Figure 7.13 **a)** for polynomial order $p = 3$. The green points illustrates control points that define the green control polygon. The force f is chosen to conform the manufactures solution

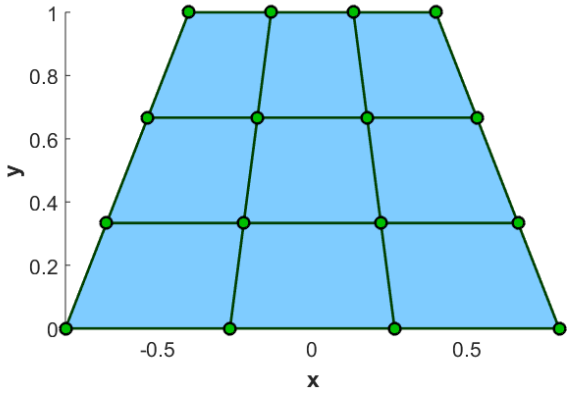
$$u(x, y) = (x^2 + y^2 - 1)(x^2 + y^2 - 16)\sin(x)\sin(y).$$

This is a smooth manufactured solution that are used in among others [10]. Affine transformations preserve lines, points, planes and parallel sides. Thus, the trapezoid is not an affine mapping since the left and right side are no longer parallel. This leads to a variable Jacobian which yield a rational function. This might result in displacement of the superconvergent points, but it is not known how much and what the affect of this is to the system.

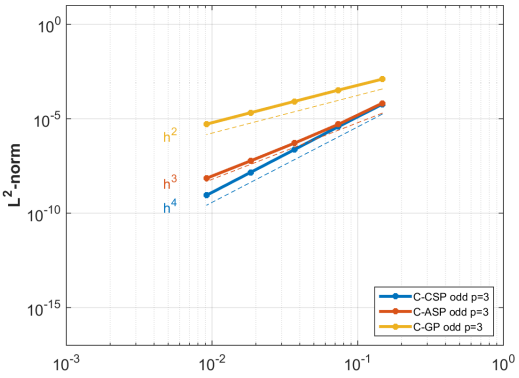
Figure 7.13 **b)** and 7.13 **c)** compares the three isogeometric collocation (IGC) methods for polynomial order $p = 3$ (odd) and $p = 4$ (even) in the L^2-norm. The three methods are IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP), where the only differences are the placement of collocation points as explained in 5.

Figure 7.14 presents results for all five methods with $p = 3 \dots 8$ in L^2-norm, 7.15 in H^1-seminorm and 7.16 in H^2-seminorm. The convergence rates of each subfigure are stated under the respective subfigure. All convergence rates coincide Table 7.1.

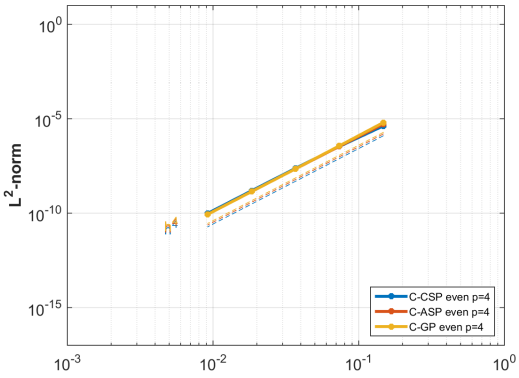
It is observed from the results that many convergence plots starts to drift out of rate much earlier that they normally would do when reaching the machine error. This should be investigated more on a server with more precision in its floating points operations. In the next section the same problem on a sharper trapezoid is considered too examine the effect from non-affinity.



a) Moderate trapezoid: Blue geometry illustrates the coarsest mesh comprising one element with green control points.

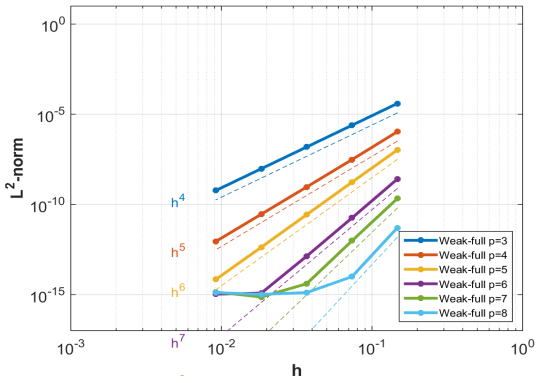


b) Moderate trapezoid: Comparison for odd p=3.

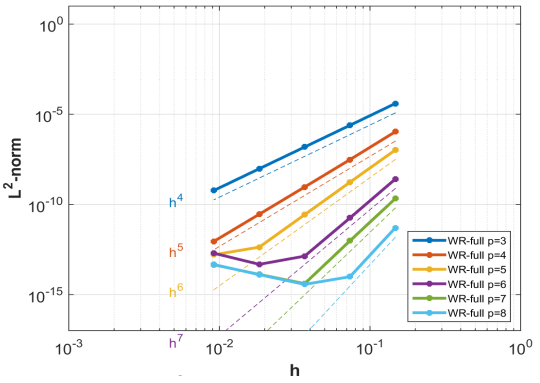


c) Moderate trapezoid: Comparison for even p=4.

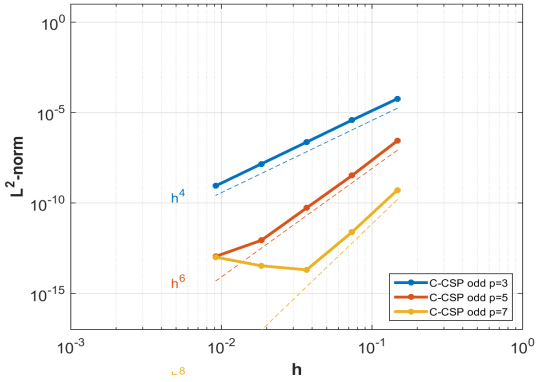
Figure 7.13: Moderate trapezoid: Geometry of coarsest mesh are presented in **a)**. In **b)** isogeometric collocation (IGC) at Greville points(C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP) are compared for odd p in the L^2-norm with smooth solution. In **c)** the same are plotted for even p.



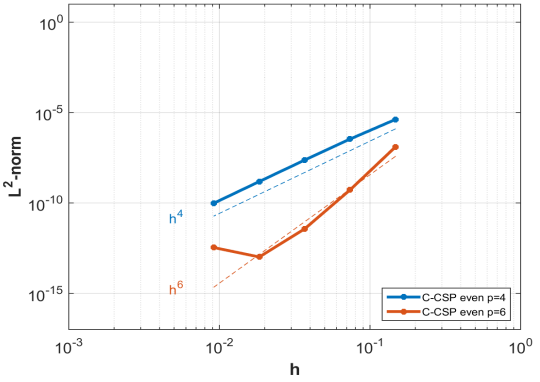
a) Weak-full: rate $p+1$.



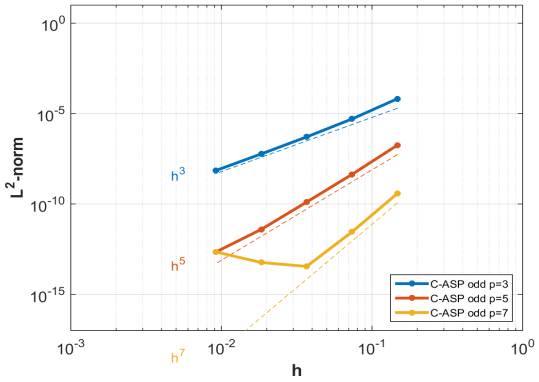
b) WR-full: rate $p+1$.



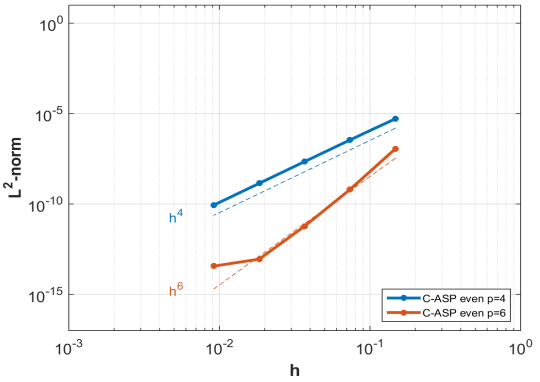
c) C-CSP odd: rate $p+1$.



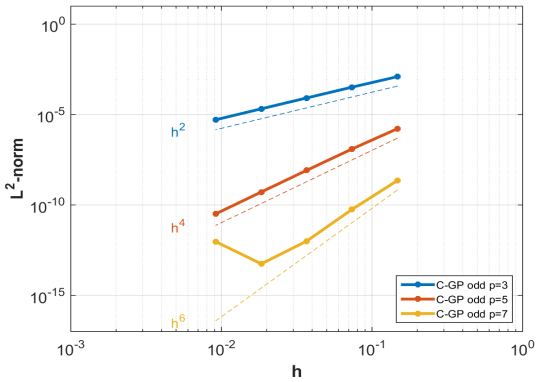
d) C-CSP even: rate p.



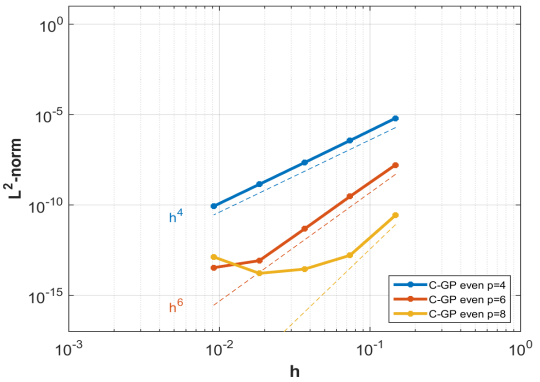
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

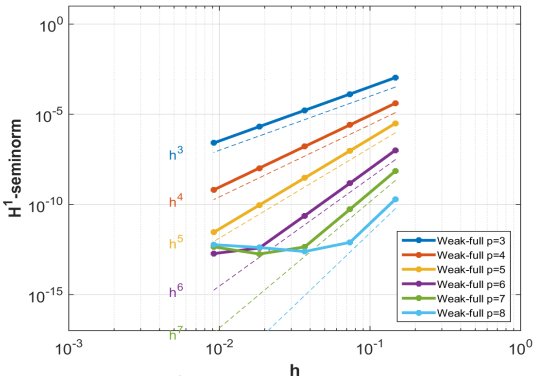


g) C-GP odd: rate $p-1$.

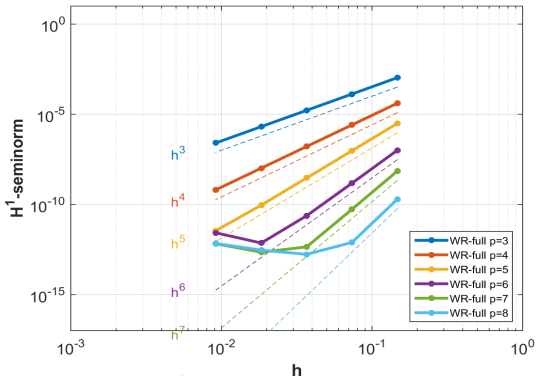


h) C-GP even: rate p.

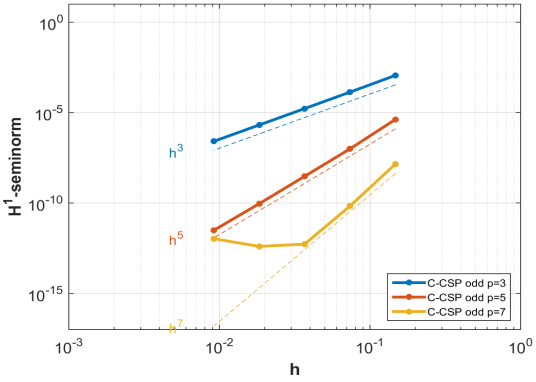
Figure 7.14: Moderate trapezoid, L^2-norm: Convergence plot of L^2 norm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.



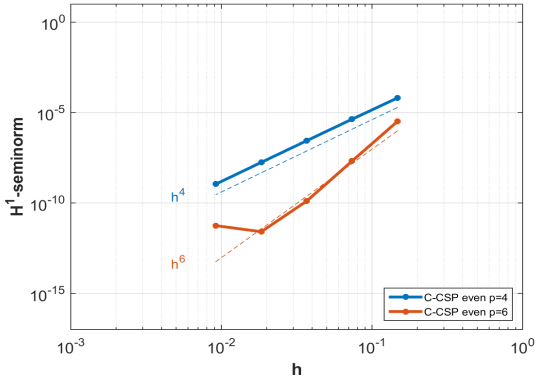
a) Weak-full: rate p.



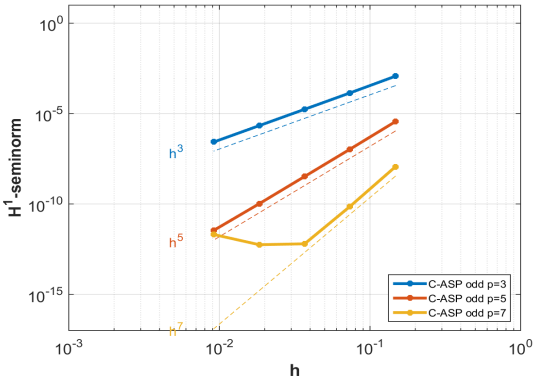
b) WR-full: rate p.



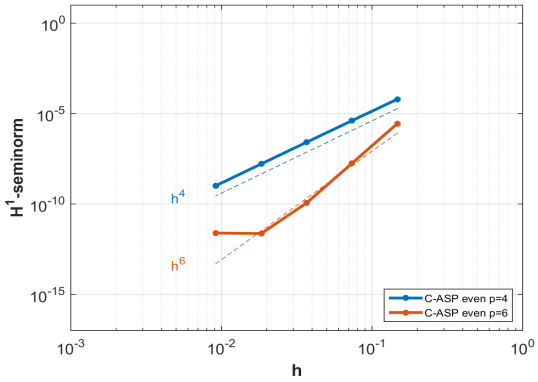
c) C-CSP odd: rate p.



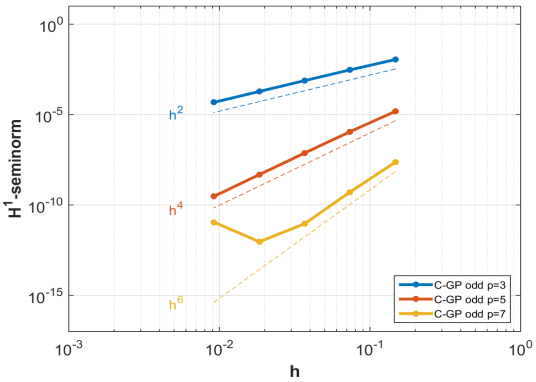
d) C-CSP even: rate p.



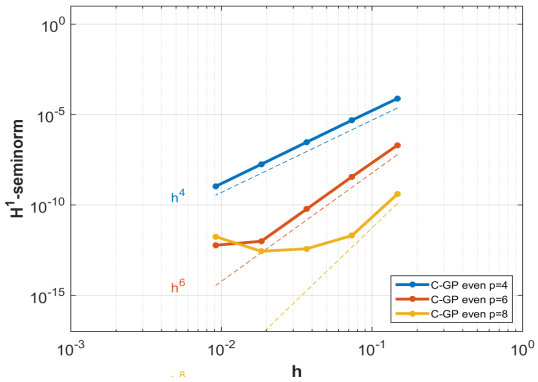
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

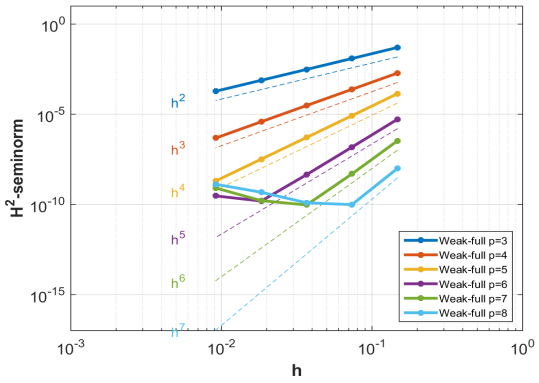


g) C-GP odd: rate $p-1$.

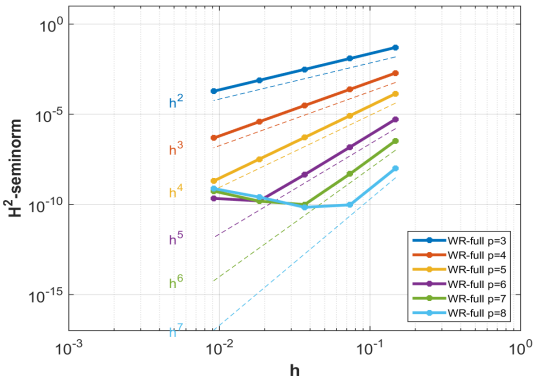


h) C-GP even: rate p.

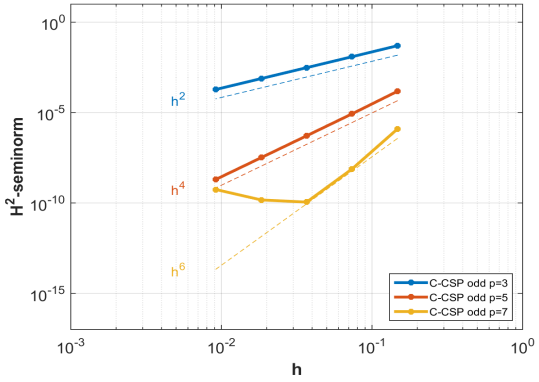
Figure 7.15: Moderate trapezoid, H^1-seminorm: Convergence plot of h^1-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.



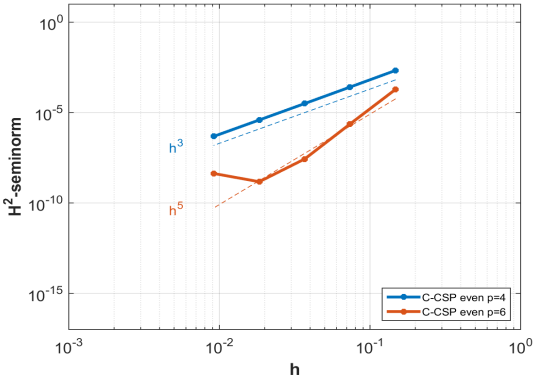
a) Weak-full: rate $p-1$.



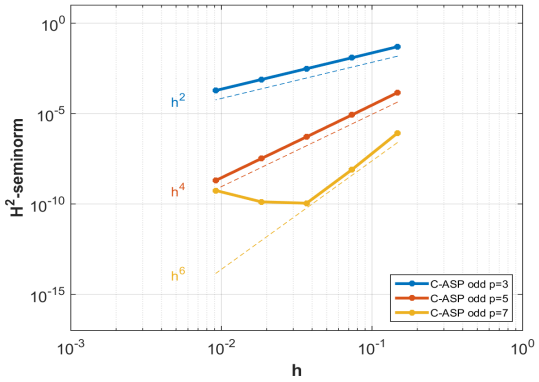
b) WR-full: rate $p-1$.



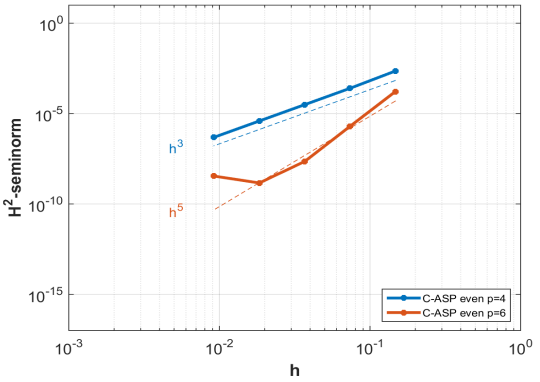
c) C-CSP odd: rate $p-1$.



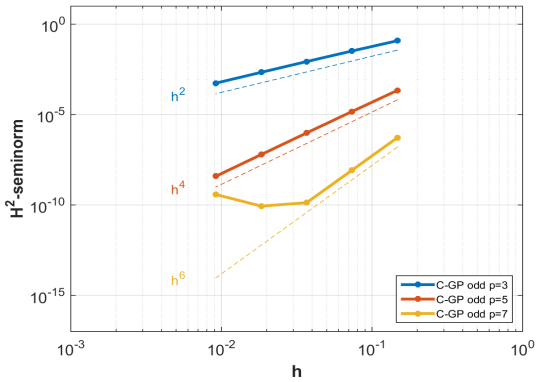
d) C-CSP even: rate $p-1$.



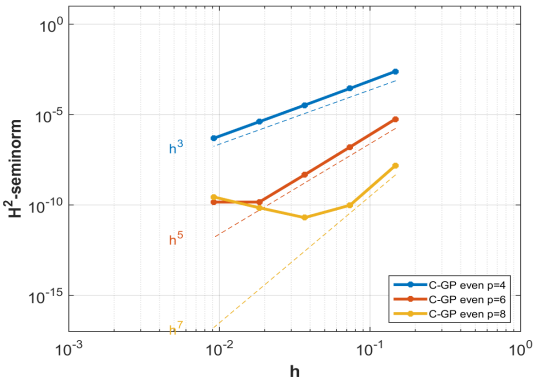
e) C-ASP odd: rate $p-1$.



f) C-ASP even: rate $p-1$.



g) C-GP odd: rate $p-1$.



h) C-GP even: rate $p-1$.

Figure 7.16: Moderate trapezoid, H^2-seminorm: Convergence plot of H^2-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.

7.5 Sharp trapezoid

The fifh problem considered is the again the trapezoid, but where the top is much sharper. This problem is chosen to see the effect non-affinity have on the results. The coarsest possible mesh of one element is shown in Figure 7.17 **a)** for polynomial order $p = 3$. The green points illustrates control points that define the green control polygon. The same manufactured solution as used in Section 7.4

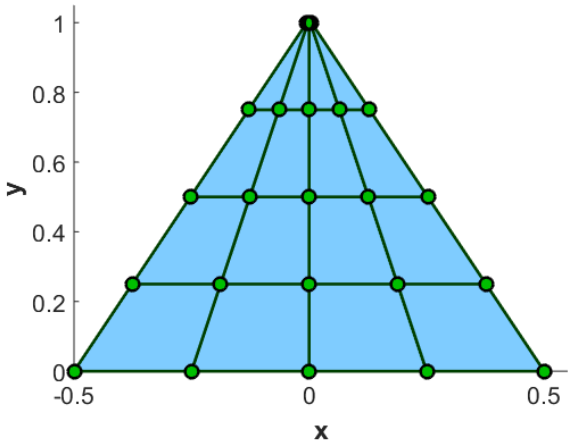
$$u(x, y) = (x^2 + y^2 - 1)(x^2 + y^2 - 16)\sin(x)\sin(y).$$

The non-affined mapping of this geometry leads again to a Jacobian which is not constant and thus, the function will be rational. The non-affinity here is higher than for the moderate trapezoid, so one would expect the results to be more uncontrolled here.

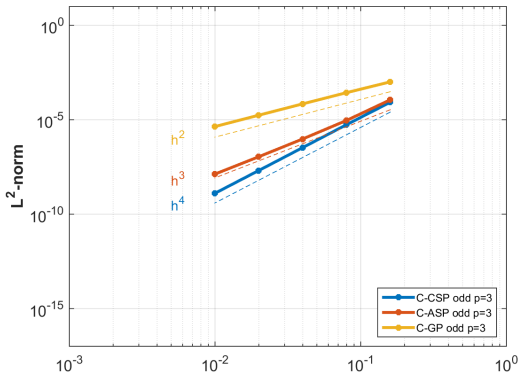
Figure 7.17 **b)** and 7.17 **c)** compares the three isogeometric collocation (IGC) methods for polynomial order $p = 3$ (odd) and $p = 4$ (even) in the L^2-norm. The three methods are IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP), where the only differences are the placement of collocation points as explained in 5.

Figure 7.18 presents results for all five methods with $p = 3 \dots 8$ in L^2-norm, 7.19 in H^1-seminorm and 7.20 in H^2-seminorm. Under each subfigure are the observed convergence rate for each method stated. All observed convergence rates coincide Table 7.1.

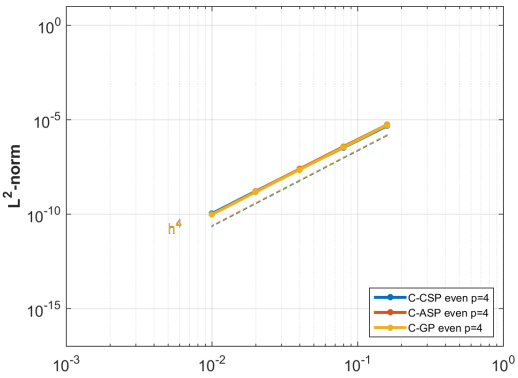
It is observed that the convergence plots of this section does not diverge much from those of Section 7.4. Thus, the superconvergent points might not have moved that much in the mapping from the parameter space, or the method is robust enough to handle it. The displacement of superconvergent points require more investigation.



a) Sharp trapezoid: Blue geometry of the coarsest mesh comprising one element with green control points.

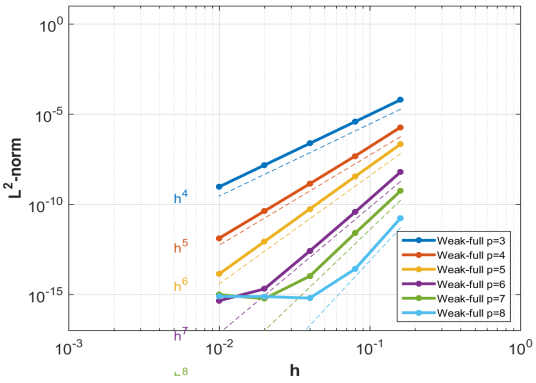


b) Sharp trapezoid: Comparison for odd p=3.

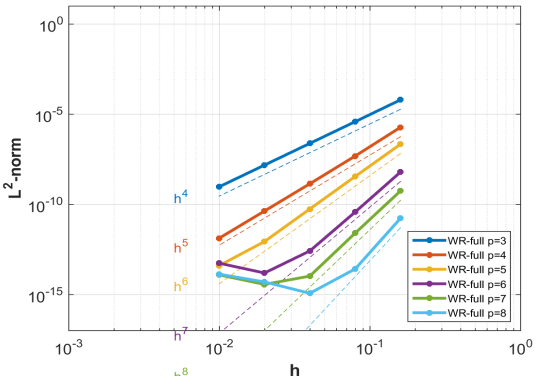


c) Sharp trapezoid: Comparison for even p=4.

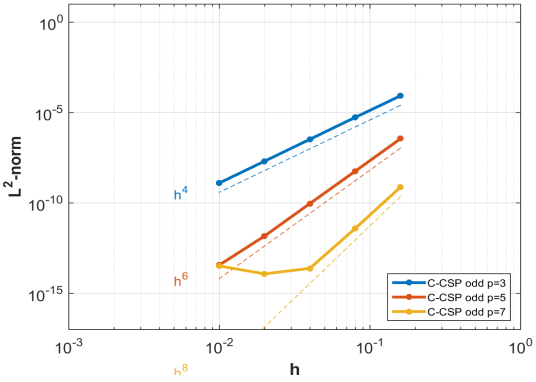
Figure 7.17: Sharp trapezoid: Geometry of coarsest mesh are presented in **a)**. In **b)** isogeometric collocation (IGC) at Greville points(C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP) are compared for odd p in the L^2-norm. In **c)** the same are plotted for even p.



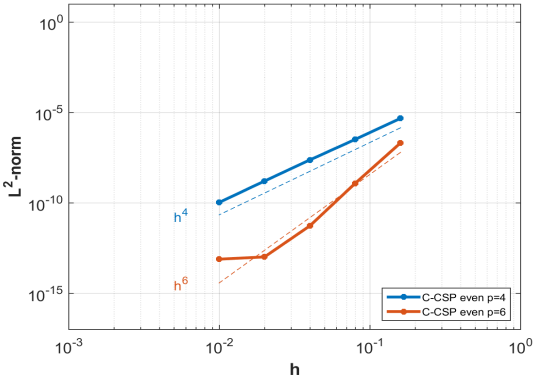
a) Weak-full: rate $p+1$.



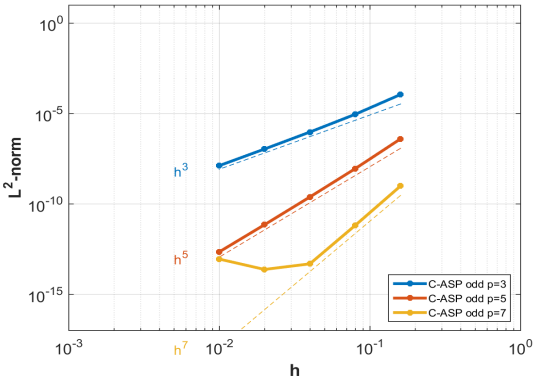
b) WR-full: rate $p+1$.



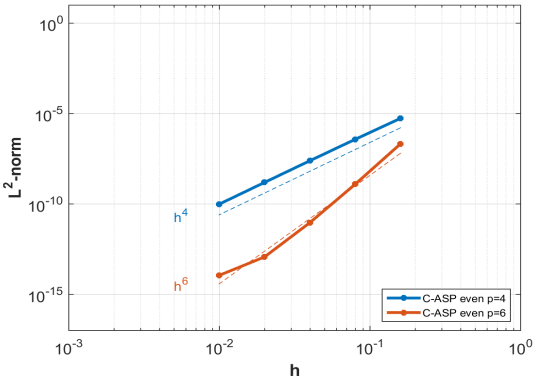
c) C-CSP odd: rate $p+1$.



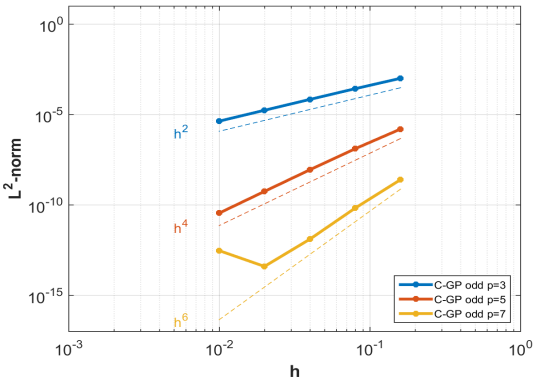
d) C-CSP even: rate p.



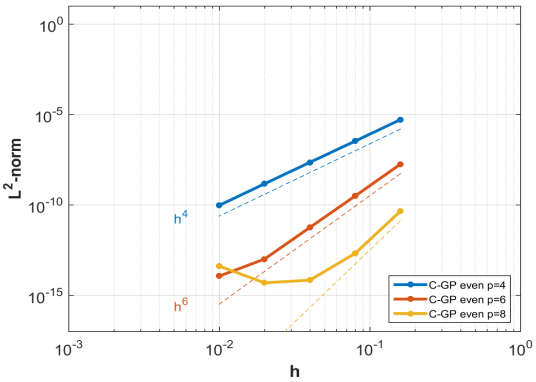
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

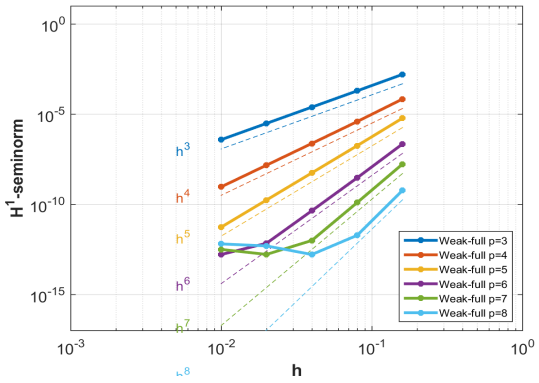


g) C-GP odd: rate $p-1$.

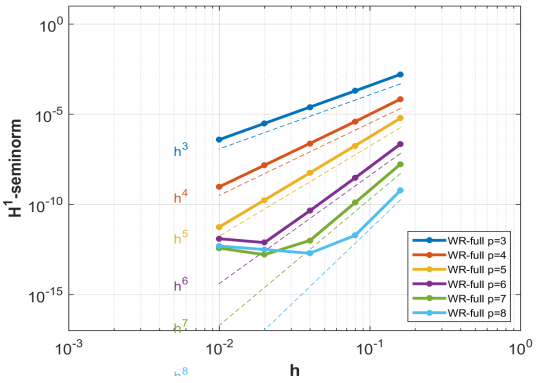


h) C-GP even: rate p.

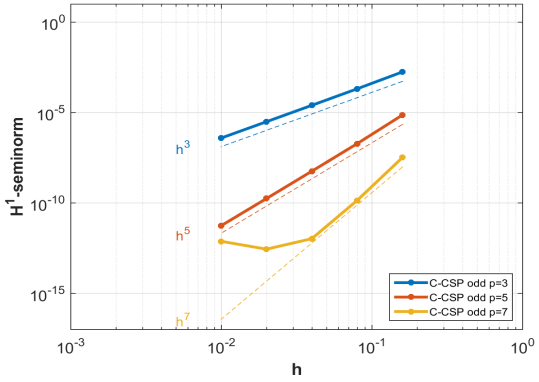
Figure 7.18: Sharp trapezoid, L^2-norm: Convergence plot of L^2 norm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.



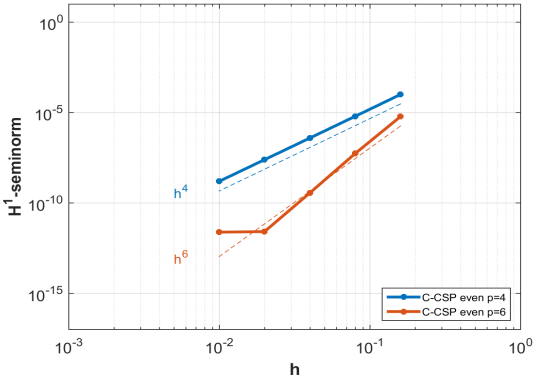
a) Weak-full: rate p.



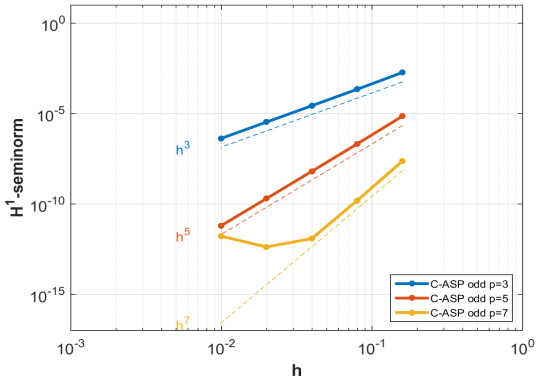
b) WR-full: rate p.



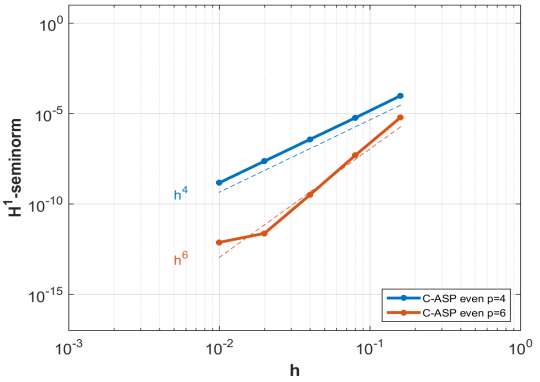
c) C-CSP odd: rate p.



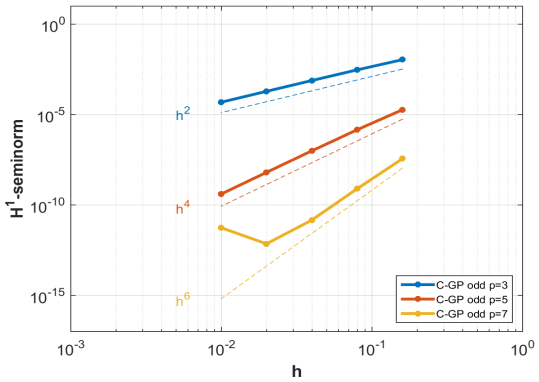
d) C-CSP even: rate p.



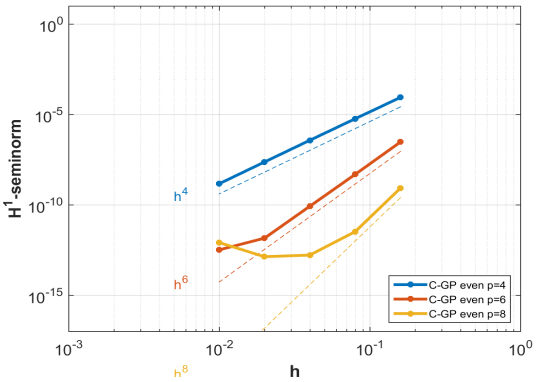
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

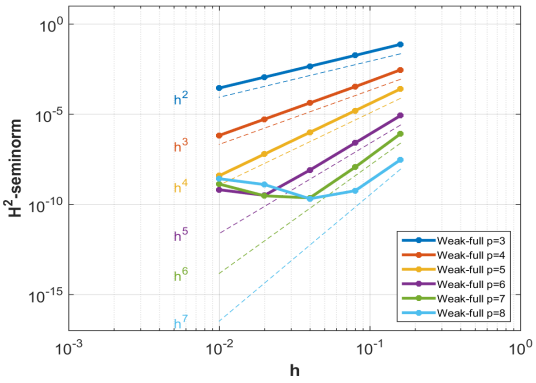


g) C-GP odd: rate $p-1$.

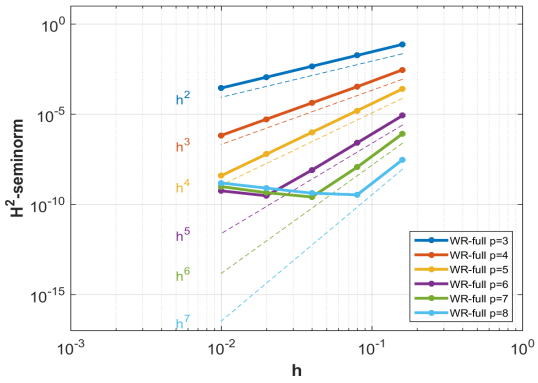


h) C-GP even: rate p.

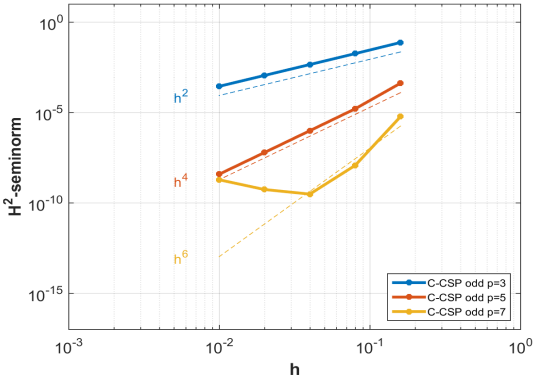
Figure 7.19: Sharp trapezoid, H^1-seminorm: Convergence plot of H^2-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.



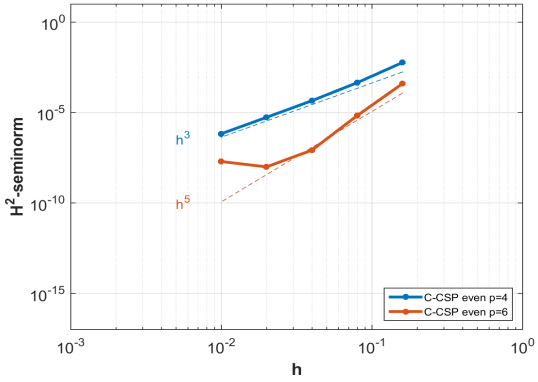
a) Weak-full: rate $p-1$.



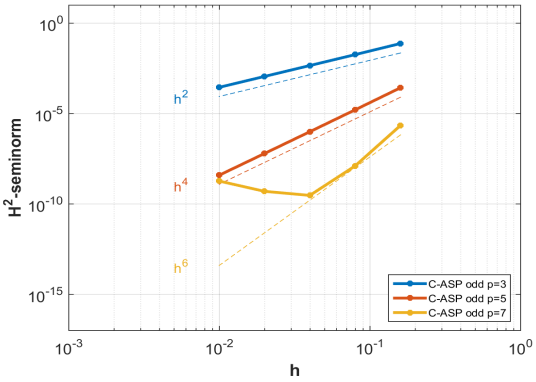
b) WR-full: rate $p-1$.



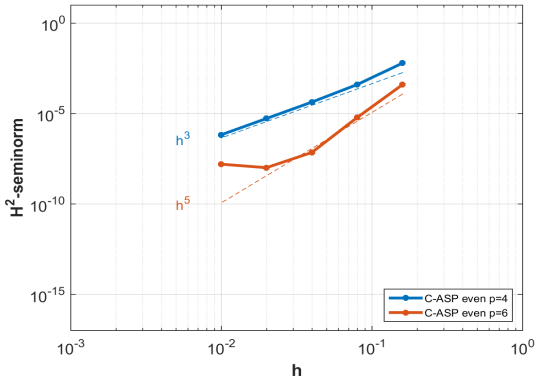
c) C-CSP odd: rate $p-1$.



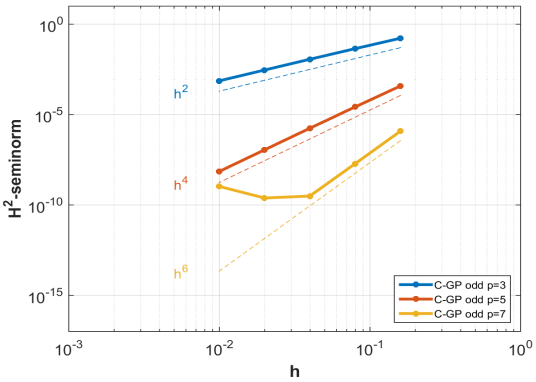
d) C-CSP even: rate $p-1$.



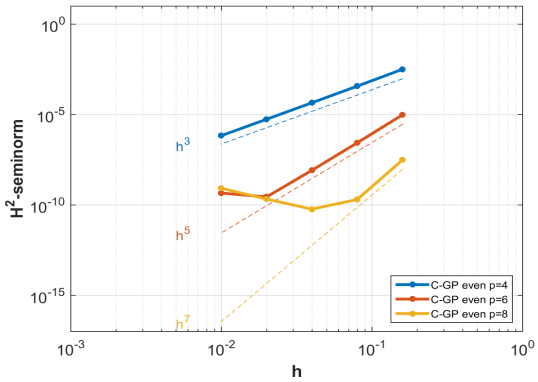
e) C-ASP odd: rate $p-1$.



f) C-ASP even: rate $p-1$.



g) C-GP odd: rate $p-1$.



h) C-GP even: rate $p-1$.

Figure 7.20: Sharp trapezoid, H^2-seminorm: Convergence plot of H^2-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.

7.6 Quarter annulus

This last problem is the quarter annulus where the coarsest possible mesh of one element is shown in Figure 8.2 **a)** for polynomial order $p = 3$. The green points illustrates control points that define the green control polygon. The force f is chosen to conform the solution

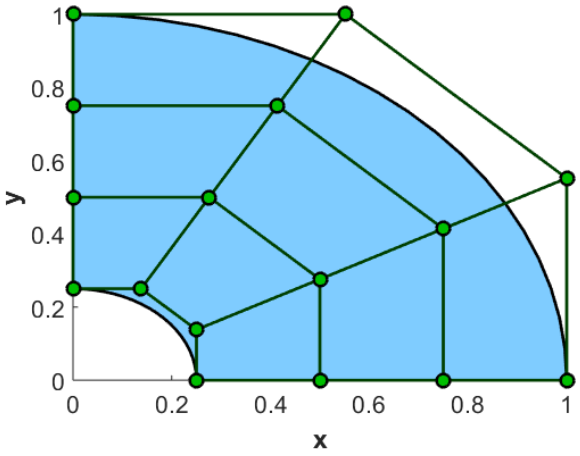
$$u(x, y) = \sin(\pi x)e^x.$$

This is a polynomial manufactured solution that are used in among others [2]. Again, the geometry is non-affine, because the parallel sides and the straight lines are not preserved in the map from the rectangular parameter space.

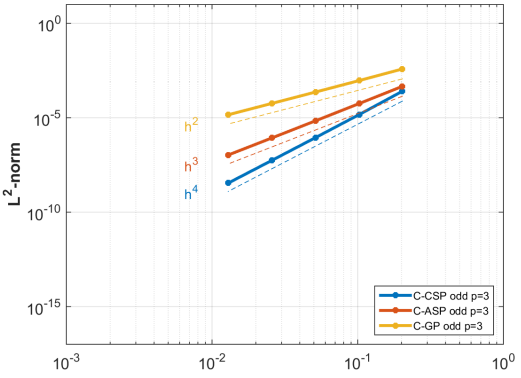
Since the quarter annulus are represented using B-splines the geometry is not perfectly circular, this is only possible when cosidering NURBS which can exactly represent such conic sections. Nevertheless, the geometry remains unchanged during refinement.

Figure 7.21 **b)** and 7.21 **c)** compares the three isogeometric collocation (IGC) methods for polynomial order $p = 3$ (odd) and $p = 4$ (even) in the L^2-norm. The three methods are IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP), where the only differences are the placement of collocation points as explained in 5.

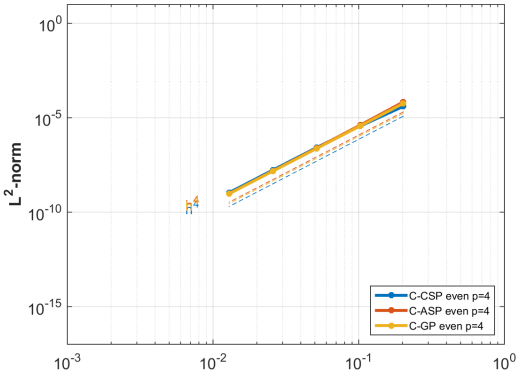
Figure 7.22 presents results from all five methods for $p = 3, \ldots, 8$ in L^2-norm, 7.23 in H^1-seminorm and 7.24 in H^2-seminorm. The convergence rates of each method are stated under the repective subfiguure. All observed convergence rates coincide Table 7.1.



a) Annulus: Blue geometry illustrates the coarsest mesh comprising one element with green control points.

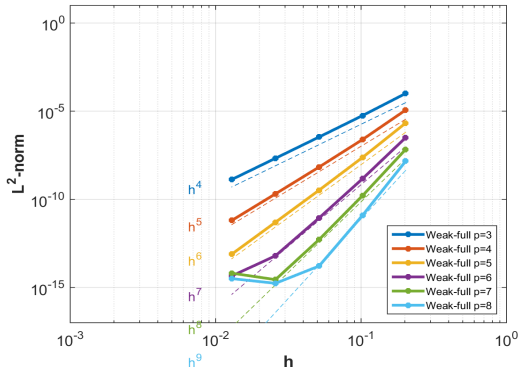


b) Annulus: Comparison for odd p=3.

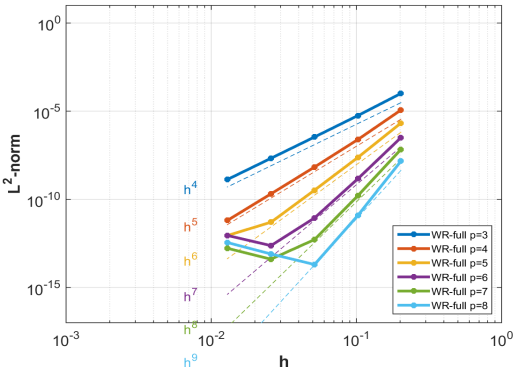


c) Annulus: Comparison for even p=4.

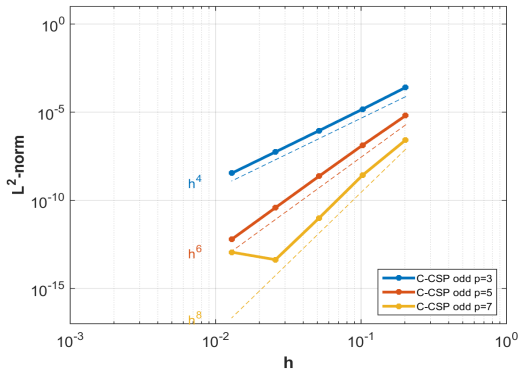
Figure 7.21: Annulus: Geometry of coarsest mesh are presented in **a)**. In **b)** isogeometric collocation (IGC) at Greville points(C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CASP) are compared for odd p in the L^2-norm for smooth $u(x, y)$. In **c)** the same are plotted for even p.



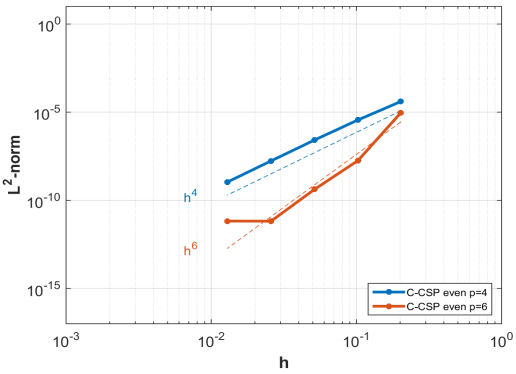
a) Weak-full: rate $p+1$.



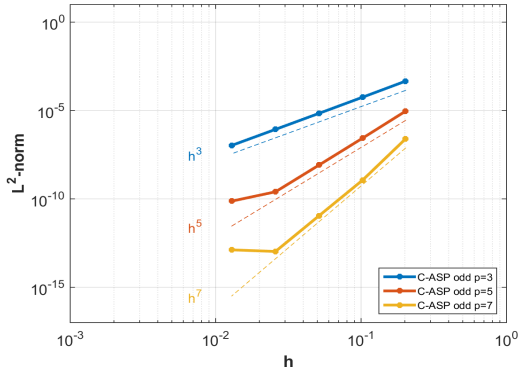
b) WR-full: rate $p+1$.



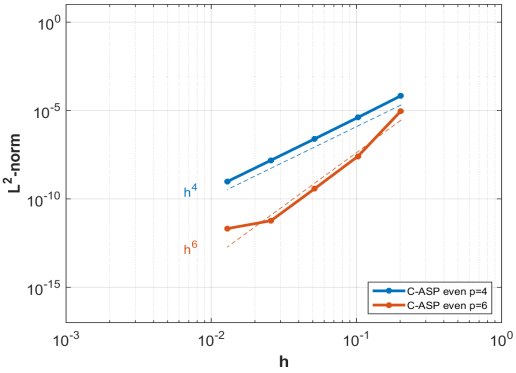
c) C-CSP odd: rate $p+1$.



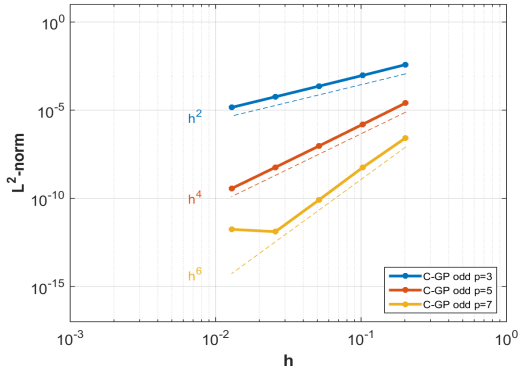
d) C-CSP even: rate p.



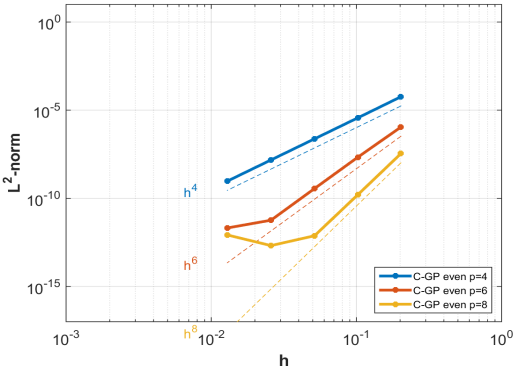
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

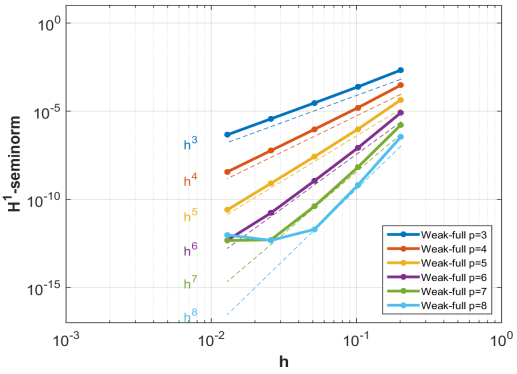


g) C-GP odd: rate $p-1$.

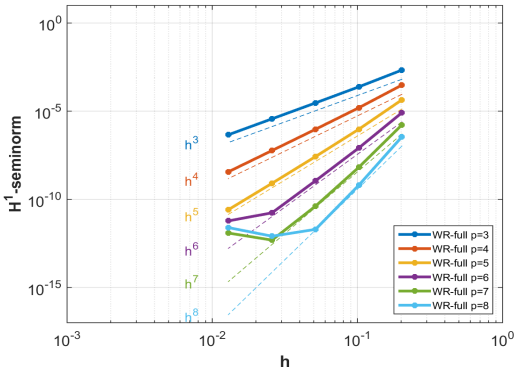


h) C-GP even: rate p.

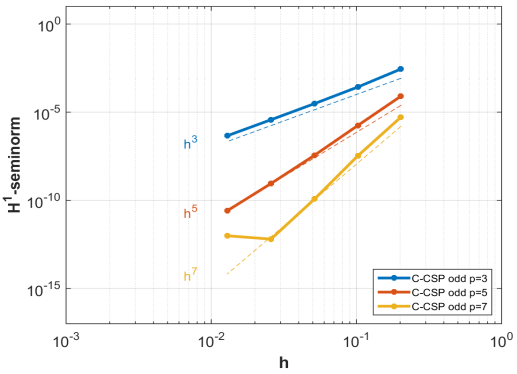
Figure 7.22: Annulus, L^2-norm: Convergence plot of L^2 norm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.



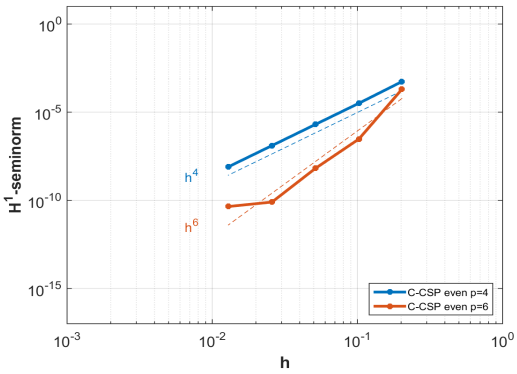
a) Weak-full: rate p.



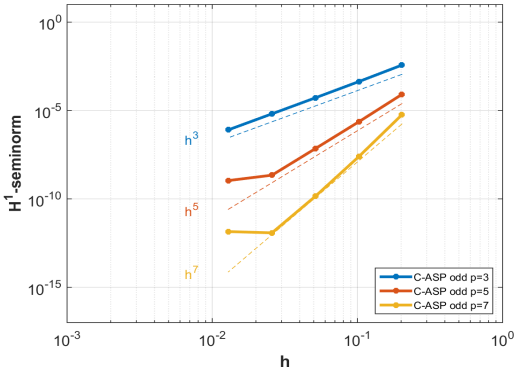
b) WR-full: rate p.



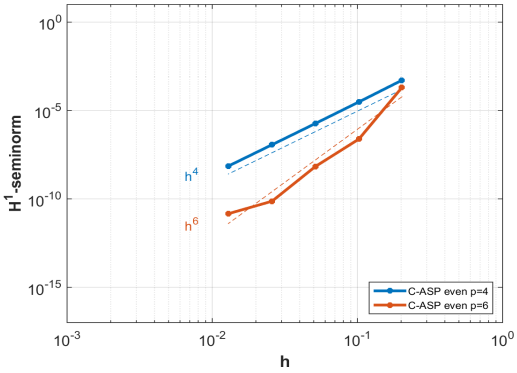
c) C-CSP odd: rate p.



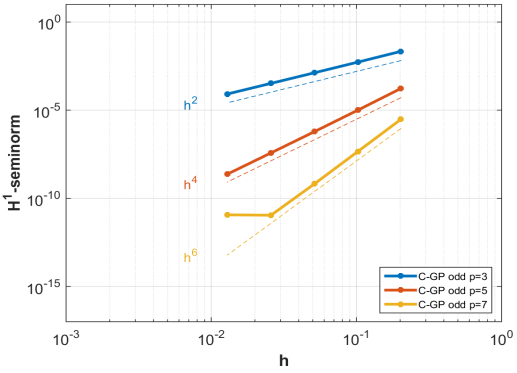
d) C-CSP even: rate p.



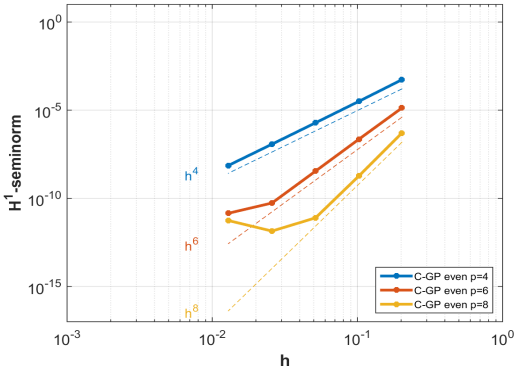
e) C-ASP odd: rate p.



f) C-ASP even: rate p.

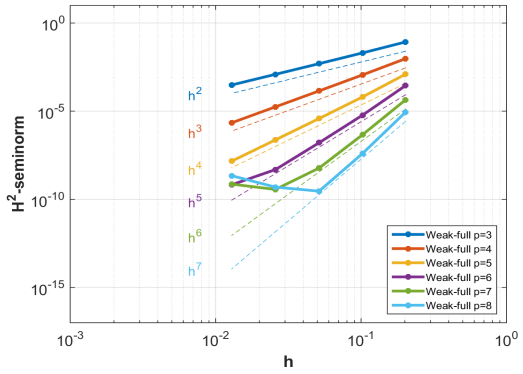


g) C-GP odd: rate $p-1$.

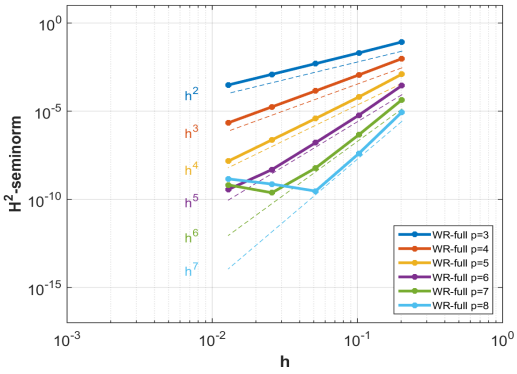


h) C-GP even: rate p.

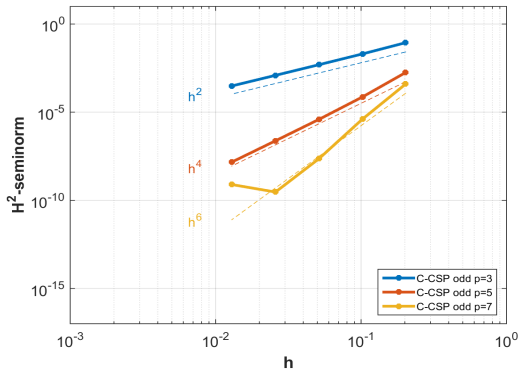
Figure 7.23: Annulus, H^1-seminorm: Convergence plot of H^1-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.



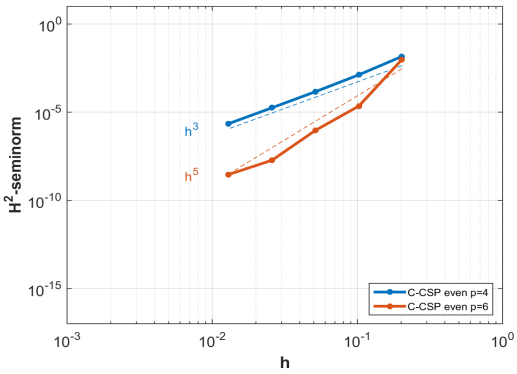
a) Weak-full: rate $p-1$.



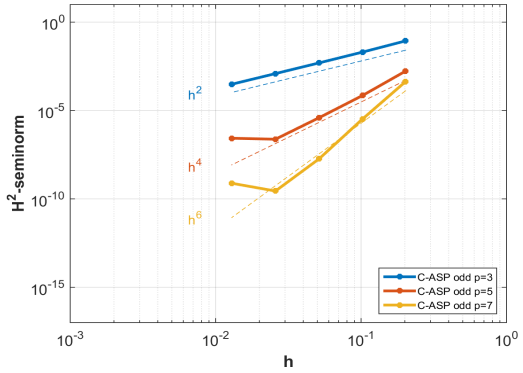
b) WR-full: rate $p-1$.



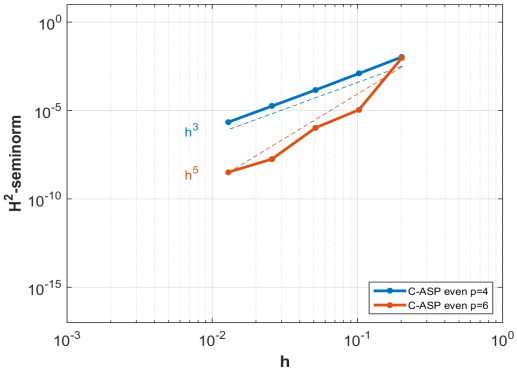
c) C-CSP odd: rate $p-1$.



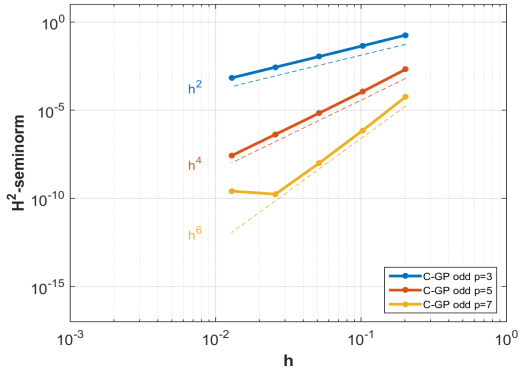
d) C-CSP even: rate $p-1$.



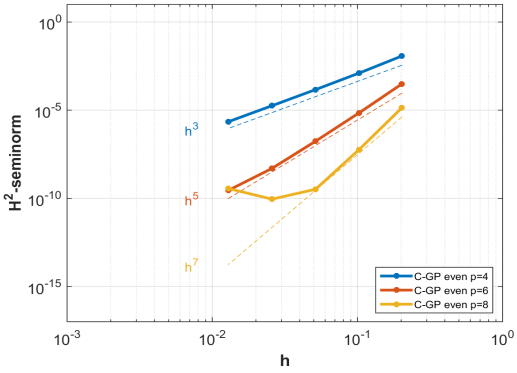
e) C-ASP odd: rate $p-1$.



f) C-ASP even: rate $p-1$.



g) C-GP odd: rate $p-1$.



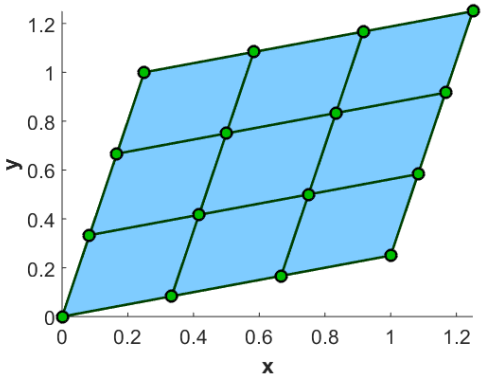
h) C-GP even: rate $p-1$.

Figure 7.24: Annulus, H^2-seminorm: Convergence plot of H^2-seminorm for weak variational method with full quadrature (Weak-full), weighted residual method with with full quadrature (WR-full), IGC at Greville points (C-GP), IGC at alternating superconvergent points (C-ASP) and IGC at clustered superconvergent points (C-CSP), with polynomial order $p = 3, 4, 5, 6, 7, 8$ and smooth solution. Convergence rates are written under respective subfigures.

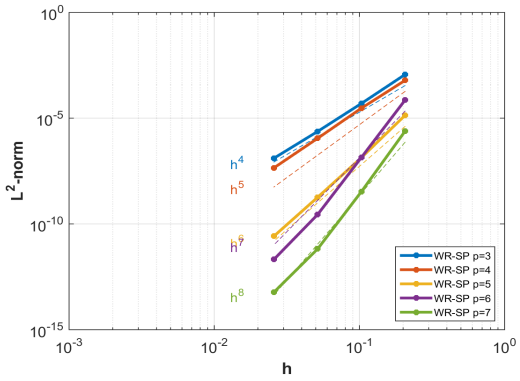
CHAPTER 8

Numerical results of the weighted residual method with superconvergent quadrature points

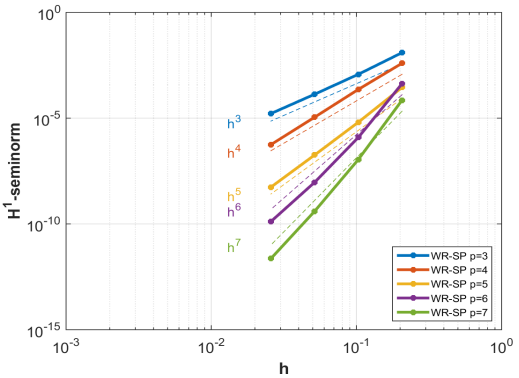
This Chapter presents numerical results of the weighted residual method with integrals approximated by quadratures at superconvergent points, as presented in Chapter 6. The weights chosen are the same as utilized in Gaussian quadrature. The problems considered here are the rhombus presented in Section 7.3 and the quarter annulus as presented in Section 7.6. The nummerical results are presented in Figure 8.1 and in Figure 8.2. Both problems yields optimal rates for all norms considered. This method is thus optimal, but require two evaluation points per element.



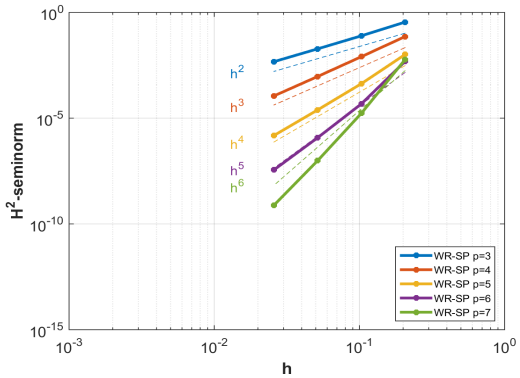
a) Rhombus: Blue geometry illustrates the coarsest mesh comprising one element with green control points.



b) L^2-norm: rate $p+1$.

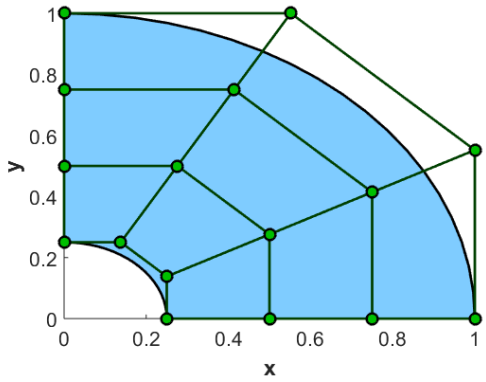


c) H^1-seminorm: rate p.

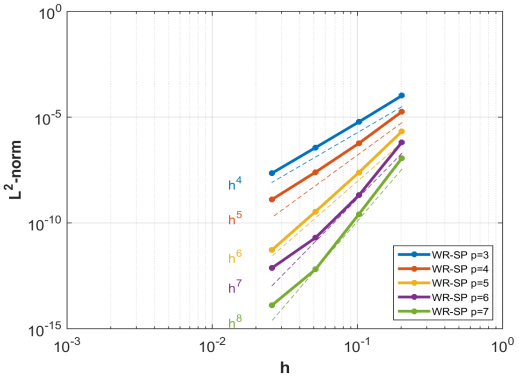


d) H^2-seminorm: rate $p-1$.

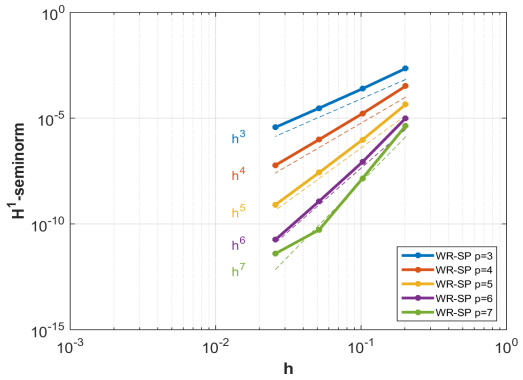
Figure 8.1: Rhombus: weighted residual method with quadrature at superconvergent points



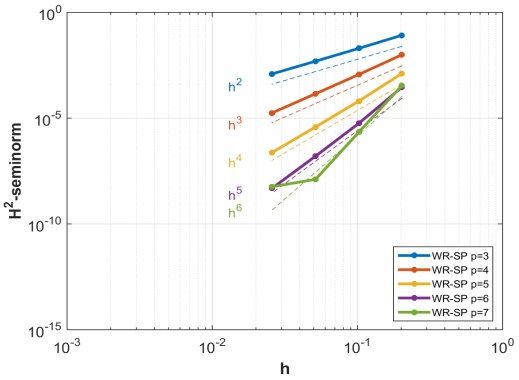
a) Annulus: Blue geometry illustrates the coarsest mesh comprising one element with green control points.



b) L^2-norm: rate $p+1$.



c) H^1-seminorm: rate p.



d) H^2-seminorm: rate $p-1$.

Figure 8.2: Annulus: weighted residual method with quadrature at superconvergent points

CHAPTER 9

Concluding remarks

Isogeometric analysis using the well-established weak variational method and the weighted residual method with full Gaussian quadrature yield optimal convergence rates. The optimal orders for splines of maximum continuity are $p+1$ in L^2-norm, p in H^1-seminorm and $p-1$ in H^2-seminorm. These methods are however inefficient when it comes to the number of evaluation points per degree of freedom. Full Gaussian quadrature is the optimal quadrature choice for traditional FEA but does not account for the additional continuity over element boundaries provided by smooth basis functions used in IGA.

Isogeometric collocation (IGC) aim at optimizing the computational efficiency with one evaluation point per degree of freedom. This lead to a very efficient method but for splines of maximum continuity the method is reported to give suboptimal convergence rates, when collocated at traditional collocation points, such as Greville (C-GP) or Demko abscissas. For even p the method is suboptimal by one order in the L^2-norm. For odd p the method is suboptimal by two orders in the L^2-norm and by one order in the H^1-seminorm. Collocation methods are designed to optimize the H^2-norm and naturally all collocation methods yield optimal convergence rates here.

Superconvergent points are points where the double derivative of the error are smallest. Using these points as collocation points improves the convergence rates. Superconvergent points for even p appear at the position of Greville points for internal elements, which might reason for why C-GP has better accuracy for even p.

Collocation using least-squares of superconvergent points (C-LS) yields optimal convergence rates at all considered norms when p is odd. But for even p the method remain suboptimal by one order. Using least-squares contradict the collocations aim to provide one evaluation point per degree of freedom, so higher computational cost

is a tradeoff for the improved accuracy of C-LS.

By carefully selecting one superconvergent point per degree of freedom the same accuracy as C-LS can be obtained with the low computational cost as traditional collocation methods. IGC at alternating superconvergent points (C-ASP) selects in general one superconvergent point per element, i.e. alternating every second superconvergent point. This method is one order suboptimal in L^2-norm for both odd and even p. Thus, it is one order worse than C-LS for odd p in the L^2-norm, but one order better than C-GP in both the L^2-norm and the H^1-seminorm for odd p.

IGC at clustered superconvergent points (C-CSP) selects points in clusters so that, in general, each element contain a symmetry of superconvergent points. This method obtain the same accuracy as C-LS with just one evaluation point per degree of freedom.

Superconvergent points might be advantageous as quadrature points for IGA. This thesis provides two examples from which it is observed that the weighted residual method with quadrature at these points yield optimal solution. The computational cost of this method is however higher than for IGC method, since two evaluation points per element are required.

CHAPTER 10

Future work

Contributions to optimal quadratures and collocation base don superconvergent points are currently under investigation by multiple research facilities and are under constant development.

The topic on optimal quadratures and collocation based on superconvergent points are constantly developing and are currently investigated by multiple researchers. Following are some areas where the research should provide some future work. Optimal quadrature points are investigated for 1D elements and then generalized to higher dimensions by tensor product. There might be performance improvements by seeking optimal points for the whole 2D or 3D element.

Current contributions to isogeometric collocation based on superconvergent points are restricted to uniform knot vectors. It is evident that finding superconvergent points to non-uniform knot vectors Is not straight forward and require more investigation.

There is still not observed any collocation point that gives the Galerkin convergence rate in H^0-norm for even p. The reason for this is still a mystery and more research are necessary to find the reason and hopefully achieve optimal rates for all norms and p's in the future.

Bibliography

[1] Hector Gomez and Laura De Lorenzis. "The variational collocation method". In: *Computer Methods in Applied Mechanics and Engineering* 309 (2016), pp. 152–181.

[2] M. Montardini, G. Sangalli, and L. Tamellini. "Optimal-order isogeometric collocation at Galerkin superconvergent points". In: *Computer Methods in Applied Mechanics and Engineering* (2016),

[3] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer Methods in Applied Mechanics and Engineering* 194.39–41 (2005), pp. 4135–4195.

[4] Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. "T-splines and T-NURCCs". In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 477–484.

[5] Thomas W. Sederberg, David L. Cardon, G. Thomas Finnigan, Nicholas S. North, Jianmin Zheng, and Tom Lyche. "T-spline Simplification and Local Refinement". In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 276–283.

[6] Thomas W. Sederberg, G. Thomas Finnigan, Xin Li, Hongwei Lin, and Heather Ipson. "Watertight Trimmed NURBS". In: *ACM Trans. Graph.* 27.3 (Aug. 2008), 79:1–79:8.

[7] T.J.R. Hughes, A. Reali, and G. Sangalli. "Efficient quadrature for NURBS-based isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 199.5–8 (2010). Computational Geometry and Analysis, pp. 301–313.

[8] F. Auricchio, F. Calabrò, T.J.R. Hughes, A. Reali, and G. Sangalli. "A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 249–252 (2012). Higher Order Finite Element and Isogeometric Methods, pp. 15–27.

[9] Dominik Schillinger, Shaikh J. Hossain, and Thomas J.R. Hughes. "Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 277 (2014), pp. 1–45.

[10] F. Auricchio, L. Beirão da Veiga, T. J. R. Hughes, A. Reali, and G. Sangalli. "Isogeometric collocation methods". In: *Mathematical Models and Methods in Applied Sciences* 20.11 (2010), pp. 2075–2107.

[11] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. "Isogeometric collocation for elastostatics and explicit dynamics". In: *Computer Methods in Applied Mechanics and Engineering* 249–252 (2012). Higher Order Finite Element and Isogeometric Methods, pp. 2–14.

[12] Dominik Schillinger, John A. Evans, Alessandro Reali, Michael A. Scott, and Thomas J.R. Hughes. "Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical {NURBS} discretizations". In: *Computer Methods in Applied Mechanics and Engineering* 267 (2013), pp. 170–232.

[13] L. De Lorenzis, J.A. Evans, T.J.R. Hughes, and A. Reali. "Isogeometric collocation: Neumann boundary conditions and contact". In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015). Isogeometric Analysis Special Issue, pp. 21–54.

[14] Cosmin Anitescu, Yue Jia, Yongjie Jessica Zhang, and Timon Rabczuk. "An isogeometric collocation method using superconvergent points". In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015). Isogeometric Analysis Special Issue, pp. 1073–1097.

[15] L. Piegl and W. Tiller. *The NURBS Book*. Monographs in Visual Communication. Springer Berlin Heidelberg, 2012.

[16] G.E. Farin. *Curves and Surfaces for CAGD: A Practical Guide*. Computer graphics and geometric modeling. Morgan Kaufmann, 2002.

[17] D.F. Rogers. *An Introduction to NURBS: With Historical Perspective*. The Morgan Kaufmann Series in Computer Graphics Series. Morgan Kaufmann Publishers, 2001.

[18] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[19] M.J Borden, Scott M.A, J.A Evans, and J.R Hughes. "Isogeometric finite element data structures based on Bézier extraction of NURBS". In: *International Journal for Numerical Methods in Engineering,* 87 (2011), pp. 15–47.

[20] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Civil and Mechanical Engineering. Dover Publications, 2012.

[21] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Science, 2005.

[22] K. Bell. *An engineering approach to finite element analysis of linear structural mechanics problems*. Akademika Publishing, 2013.

[23] R.D. Cook. *Concepts and applications of finite element analysis*. Wiley, 2001.

[24] A. Quarteroni. *Numerical Models for Differential Problems*. MS&A. Springer Milan, 2014.

[25] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Studies in Mathematics and its Applications. Elsevier Science, 1978.

[26] Silje I. Hansen. "Isogeometric analysis based on Bézier Extraction of B-splines". MA thesis. Department of Mathematical Sciences: Norwegian University of Science and Technology, 2016.

[27] E. DiBenedetto. *Partial Differential Equations: Second Edition*. Cornerstones. Birkhäuser Boston, 2010.

[28] S.H. Friedberg, A.J. Insel, and L.E. Spence. *Linear Algebra*. Featured Titles for Linear Algebra (Advanced) Series. Pearson Education, 2003.

[29] Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. "Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes". In: *Mathematical Models and Methods in Applied Sciences* 16.07 (2006), pp. 1031–1090.

[30] K.A. Johannessen, T. Kvamsdal, and T. Dokken. "Isogeometric analysis using {LR} B-splines". In: *Computer Methods in Applied Mechanics and Engineering* 269 (2014), pp. 471–514.

[31] L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli. "Some estimates for h–p–k-refinement in Isogeometric Analysis". In: *Numerische Mathematik* 118.2 (2010), pp. 271–305.

[32] M. Kumar, T. Kvamsdal, and K.A. Johannessen. "Simple a posteriori error estimators in adaptive isogeometric analysis". In: *Computers & Mathematics with Applications* 70.7 (2015). High-Order Finite Element and Isogeometric Methods, pp. 1555–1582.

[33] Dominik Schillinger, John A. Evans, Alessandro Reali, Michael A. Scott, and Thomas J.R. Hughes. "Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical {NURBS} discretizations". In: *Computer Methods in Applied Mechanics and Engineering* 267 (2013), pp. 170–232.

[34] E.W. Hobson. *The Theory of Functions of a Real Variable and the Theory of Fourier's Series*. The Theory of Functions of a Real Variable and the Theory of Fourier's Series v. 2. Harren Press, 1950.

[35] C. de Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer New York, 2001.

[36] L. Wahlbin. *Superconvergence in Galerkin Finite Element Methods*. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 1995.

[37] M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, and T.J.R. Hughes. "Isogeometric finite element data structures based on Bézier extraction of T-splines". In: *International Journal for Numerical Methods in Engineering* 88.2 (2011), pp. 126–156.

[38] M.A. Scott, D.C. Thomas, and E.J. Evans. "Isogeometric spline forests". In: *Computer Methods in Applied Mechanics and Engineering* 269 (2014), pp. 222–264.

[39] E.J. Evans, M.A. Scott, X. Li, and D.C. Thomas. "Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015). Isogeometric Analysis Special Issue, pp. 1–20.

[40] D.C. Thomas, M.A. Scott, J.A. Evans, K. Tew, and E.J. Evans. "Bézier projection: A unified approach for local projection and quadrature-free refinement and coarsening of {NURBS} and T-splines with particular application to isogeometric design and analysis". In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015). Isogeometric Analysis Special Issue, pp. 55–105.

[41] D. Schillinger, P.K. Ruthala, and L.H. Nguyen. "Lagrange extraction and projection for NURBS basis functions: A direct link between isogeometric and standard nodal finite element formulations". In: *International Journal for Numerical Methods in Engineering* (2016).

APPENDIX A

Lebesgue and Sobolev space

A *Lebesgue space* is a space containing all p-integrable functions and is defined as

$$L^p(\Omega) = \{u : \Omega \to \mathbb{R} \ : \ \|u\|_{Lp} < \infty\}, \tag{A.1}$$

where

$$\|u\|_{L^p} := \left(\int_\Omega |u(x)|^{\mathrm{d}}\, x\right)^{\frac{1}{p}},$$

is the norm induced by the inner product

$$(u, v) = \int_\Omega u(x)v(x)\,\mathrm{d}x.$$

Thus, a Lebesgue space is a Hilbert space. The space composed of functions with square integrable derivatives up to order k is called a *Sobolev space* and is defined as

$$H^k(\Omega) = \{u \in L^2(\Omega) \ : \ D^{\boldsymbol{\alpha}} u \in L^2(\Omega), \forall\, |\boldsymbol{\alpha}| \leq k\}, \tag{A.2}$$

where the norm and semi-norm are

$$\|u\|_{H^k(\Omega)} := \left[\sum_{|\boldsymbol{\alpha}|\leq k} \|D^\alpha u\|^2_{L^2(\Omega)}\right]^{1/2}, \tag{A.3}$$

$$|u|_{H^k(\Omega)} := \left[\sum_{|\boldsymbol{\alpha}|=k} \|D^\alpha u\|^2_{L^2(\Omega)}\right]^{1/2}, \tag{A.4}$$

and

$$D^{\boldsymbol{\alpha}} u = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}} u, \qquad |\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d \tag{A.5}$$

denotes *weak*, or *distributional derivatives*. For more information on this topics the reader is referred to [24, ch. 2].

APPENDIX B

Isogeometric analysis based on Bézier Extraction

Section 2.2 shows that B-splines of order $p > 0$ are not localized in one element, but span several. This is in contrast to basis functions used in traditional FEA and complicates implementation of the isogeometric analysis. In this chapter a framework based on Bézier extraction is presented. The Bézier extraction operator reduces B-spline and NURBS basis functions to a set of Bernstein polynomials. As seen in Section 2.1, all Bernstein basis functions are local to one element, and allow for C^0-continuous functions across element boundaries. This will provide a element structure for IGA, which can be incorporated into existing FEM codes. The content in this chapter is based on the initial article of this topic presented by Borden et al. in [19].

This chapter presents computation of the Bézier extraction operator in Section B.1 and localize this operator into elements in Section B.2.

B.1 The Bézier Extraction Operator

Section 2.2 states that B-spline basis functions are C^{p-m}-continuous across each element boundary. The continuity depend on the polynomial order p of basis functions and the multiplicity m of the knot value. The multiplicity of a knot may be increased by knot insertion as defined in Section 2.2.5, which result in reduced continuity over the corresponding element boundary. If all internal knots of a knot vector have multiplicity p then exactly one basis function is defined over each element boundary while the rest are localized in the element. This yield C^0-continuous Bézier elements, which may be processed in the same way as traditional FEA elements.

The *Bézier extraction operator* serve as a mapping between B-spline basis functions $N(\xi)$ and Bernstein basis functions $B(\xi)$. Assume a given knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$ with a set of control points $\mathcal{P} = \{\boldsymbol{P_i}\}_{i=1}^{n}$ defining an object, the extraction operator is generated by inserting a set of knots $\bar{\Xi} = \{\bar{\xi}_1, \bar{\xi}_2, \ldots, \bar{\xi}_\mu\}$. For each new knot $\bar{\xi}_j$, $j = 1, 2, \ldots, \mu$, a set of alpha values α_i^j, $i = 1, 2, \ldots, n+j$ are computed from Equation (2.12) to construct the jth extraction operator $\boldsymbol{C}^j \in \mathbb{R}^{(n+j-1)\times(n+j)}$ as

$$\boldsymbol{C}^j = \begin{bmatrix} \alpha_1 & 1-\alpha_2 & 0 & & \cdots & 0 \\ 0 & \alpha_2 & 1-\alpha_3 & 0 & & \vdots \\ & & & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ 0 & \cdots & & 0 & \alpha_{(n+j-1)} & 1-\alpha_{(n+j)} \end{bmatrix}. \tag{B.1}$$

In [19] Borden et al. presents a fast algorithm for constructing the Bézier extraction operator. Let $\overline{\mathcal{P}}^1 = \mathcal{P}$, then Equation (2.11) can be rewritten as a sequence of matrix operations

$$\overline{\mathcal{P}}^{j+1} = (\boldsymbol{C}^j)^T \overline{\mathcal{P}}^j, \qquad j = 1, 2, \ldots, \mu,$$

where the final set of points $\mathcal{P}^B = \overline{\mathcal{P}}^{\mu+1}$ corresponds the control points required to represent the same object with a Bernstein basis. This approach ensures that the parametric and geometric properties of the object remain unchanged. Defining $\boldsymbol{C}^T = (\boldsymbol{C}^m)^T (\boldsymbol{C}^{m-1})^T \cdots (\boldsymbol{C}^1)^T$ yields

$$\mathcal{P}^B = \boldsymbol{C}^T \mathcal{P}.$$

Here $\boldsymbol{C}$ is the Bézier extraction operator with dimension $n \times (n+m)$. Recall that $\mathcal{P}$ has dimension $n \times d$ and so $\mathcal{P}^B$ has dimension $(n+m) \times d$.

Since the parametric and geometric properties of the object remain unchanged, this hold

$$\begin{aligned} \mathcal{P}^T N(\xi) &= (\mathcal{P}^b)^T B(\xi) \\ &= (\boldsymbol{C}^T \mathcal{P})^T B(\xi) \\ &= \mathcal{P}^T \boldsymbol{C} B(\xi), \end{aligned}$$

for arbitrary $\mathcal{P}$. Thus, $\boldsymbol{C}$ is a mapping from a Bernstein basis $\boldsymbol{B}(\xi)$ to a B-spline basis $\boldsymbol{N}(\xi)$ by the following relation

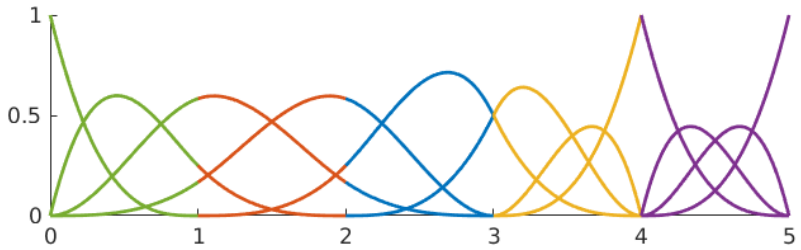
$$N(\xi) = \boldsymbol{C} B(\xi). \tag{B.2}$$

Remark that the construction of $\boldsymbol{C}$ only requires a knot vector and is independent on the basis functions or control points $\mathcal{P}$. Therefore, the same Bézier extraction operator can be applied to NURBS, as presented in [19]. For a NURBS basis the weights must be accounted which lead to a different set of computations.

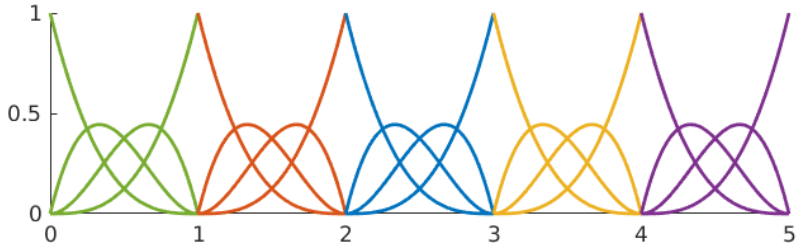
B.2 Localized Extraction Operator

A Bézier extraction operator defined by Equation (B.1) is global to all basis functions in a parameter domain. This section presents a simple example to clarify how the Bézier extraction operator works and further, discusses localization of the global operator into elements.

By knot insertion the B-spline basis of Figure B.1 **a)** can be decomposed to the Bernstein basis in Figure B.1 **b)**. The B-spline basis is defined over five elements by knot vector $\Xi = [0, 0, 0, 0, 1, 2, 3, 3, 4, 4, 4, 5, 5, 5, 5]$ and have polynomial order $p = 3$. Each colored segment correspond one element. Notice how the support of B-spline basis functions are changing with knot multiplicities. In the fifth element all but one B-spline basis functions are supported only in this element, and the basis is identical the elemental Bernstein basis. All Bézier elements have the same Bernstein basis.



a) B-spline basis functions.



b) Bernstein basis functions.

Figure B.1: B-spline basis functions generated over five elements by $\Xi = [0, 0, 0, 0, 1, 2, 3, 3, 4, 4, 4, 5, 5, 5, 5]$ of order $p = 3$ are shown in a). Bézier extraction of the basis results in the Bernstein basis of b). Each colored segment illustrate an element.

By Equation (B.1) a Bézier extractor operator is computed and Equation (B.2) is written as in Figure B.2. All B-spline and Bézier elements support $p + 1$ linear independent functions, which constitutes a basis for all polynomials of order p. Thus, a set of B-spline basis functions $\boldsymbol{N}^e$ localized in element e can be written as a linear

combination of a Bernstein basis $\boldsymbol{B}^e$. Hence,

$$\boldsymbol{N}^e(\xi) = \boldsymbol{C}^e \boldsymbol{B}^e(\xi), \tag{B.3}$$

where $\boldsymbol{C}^e$ is a part of the global $\boldsymbol{C}$. This is illustrated by the colorful boxes in Figure B.2. One local system of equations is illustrated by three boxes with the same color: one surrounding the resulting $p+1$ B-spline basis functions N_i, one surrounding the local Bézier extraction operator $\boldsymbol{C}^e$ and one surrounding the initial $p+1$ Bernstein basis functions B_j.

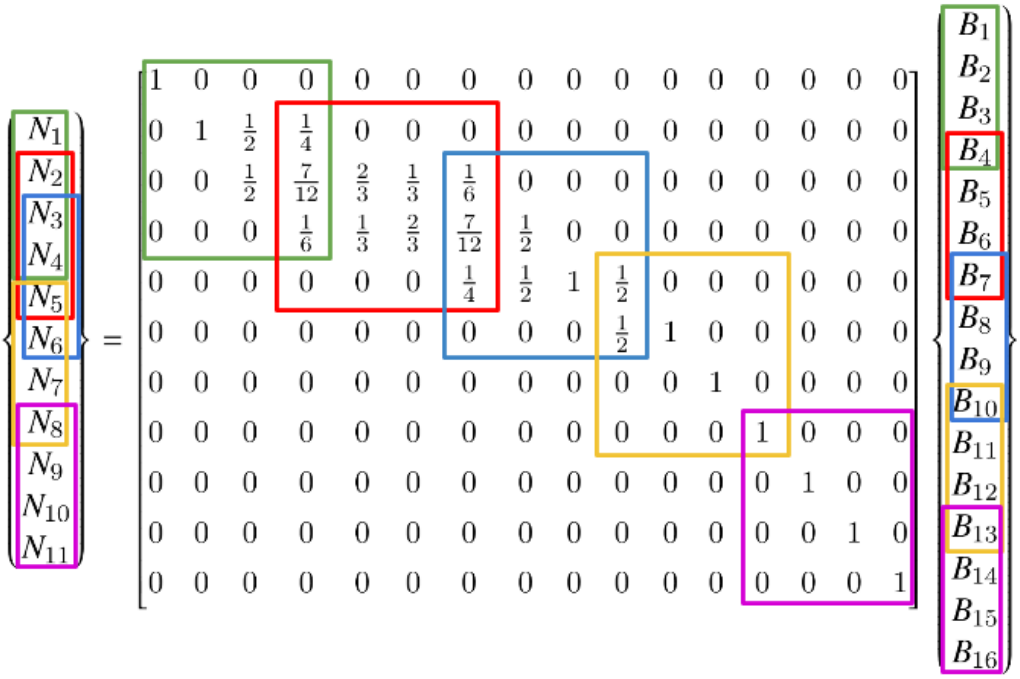


Figure B.2: A system of equations showing the relationship between B-spline basis functions N_i and Bernstein polynomials B_i with respect to the Bézier extraction operator C. Squares represent elements, each with a different color. Squares of the same color represent a local system of equations.

Notice that $\boldsymbol{C}^e$ is more complicated when the multiplicity of its defining knot values are low, e.g. the red box representing $\boldsymbol{C}^2$. When an element is spanned by two knot values of multiplicity $m \geq p$ the B-spline basis is identical to Bernstein basis and $\boldsymbol{C}^e$ become the identity map, this can be seen in the purple box corresponding $\boldsymbol{C}^5$. Also note that there are more overlap between boxes when multiplicities are low. The overlap is due to basis functions that are defined over more elements. When multiplicities are low, knots are spread over more knot values and the basis function

are supported by more elements.

Application of local Bézier extractions with Equation (B.3) coincide the main concept of finite element analysis, which is to divide computations into elements. This will also have positive effects on efficiency and storage requirements of the method. Even though multiple small systems of equations must be solved, the total number of floating point operations is fewer than what is required to solve the global system of Equation (B.2). Further, to store local Bézier extraction operators apart demand less storage space. This is regardless of overlapping data points and is due to the sparsity of the global Bézier extraction operator.

With localized Bézier extractions computation time and storage requirements of Bernstein basis functions are also reduced. Recall from Section 2.1 that all Bézier elements maps from the same parameter element provided with a Bernstein basis. Thus, all Bézier elements comprises the same basis. For the considered example that is

$$\begin{Bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{Bmatrix} = \begin{Bmatrix} B_4 \\ B_5 \\ B_6 \\ B_7 \end{Bmatrix} = \begin{Bmatrix} B_7 \\ B_8 \\ B_9 \\ B_{10} \end{Bmatrix} = \begin{Bmatrix} B_{10} \\ B_{11} \\ B_{12} \\ B_{13} \end{Bmatrix} = \begin{Bmatrix} B_{13} \\ B_{14} \\ B_{15} \\ B_{16} \end{Bmatrix}.$$

Local Bézier extractions by Equation (B.3) require that just one set of $p+1$ Bernstein basis functions are computed, and only once for each needed polynomial order p.

The isogeometric analysis based on Bézier extraction is easily extended to NURBS as presented in [19]. This approach is further generalized to T-splines in [37], hierarchical B-splines in [38] and hierarchical T-splines in [39]. This concept have further been modified to Bézier projection in [40] and Lagrange extraction and projection in [41].