

# **Methods based on B-splines for model representation, numerical analysis and image registration**

## **DISSERTATION**

Zur Erlangung des akademischen Grades eines  
Doktor-Ingenieur  
an der Fakultät Bauingenieurwesen  
der Bauhaus-Universität Weimar

vorgelegt von

M.Sc. Jia, Yue

geboren am 11. Juli 1986 in Shannxi, China

Mentor:

Prof. Dr.-Ing. Timon Rabczuk, Bauhaus-Universität Weimar

Gutachter:

Prof. Dr. rer. nat. Klaus Gürlebeck, Bauhaus-Universität Weimar

Prof. Dr. Alessandro Reali, Università degli Studi di Pavia

Tag der Disputation: 19 November 2015

*To my whole family ...*

## Acknowledgements

I am grateful to all people who have contributed towards shaping this dissertation. First of all, I would like to express my sincere gratitude to my supervisor, Prof. Dr.-Ing Timon Rabczuk, for his guidance and encouragement in my research at the Bauhaus-Universität Weimar. His great enthusiasm in academic research always inspire me along my research road. His advice on both research and as well as on my future career have been priceless.

Furthermore, I would like to thank Prof. Dr. rer. nat Klaus Gürlebeck for his comments during the review and the defense of my thesis. Special thank to Prof. Dr. Alessandro Reali for accepting the presidency of the examination board and his comments to my thesis.

This project would have been impossible without the support of the European Union through the FP7-grant ITN (Marie Curie Initial Training Networks) INSIST (Integrating Numerical Simulation and Geometric Design Technology).

I would like to thank to Dr. Cosmin at ISM of Bauhaus-Universität Weimar for his comments on my thesis and also for his friendly kind cooperation.

During my research, I spent around 10 months external studying with Prof. Jessica Zhang at the Department of the Mechanical Engineering of Carnegie Mellon University in Pittsburgh USA. Many thanks to her supervising and great support.

In addition, thanks also go to my colleagues at ISM of Bauhaus-Universität Weimar for their help and friendly support.

Finally, a special thanks to my family. Words cannot express how grateful I am to my mother, my father, my elder sister and my all dear family members for their emotional support.

Weimar, December 2015

Yue Jia



## Abstract

The thesis consists of inter-connected parts for modeling and analysis using newly developed isogeometric methods. The main parts are reproducing kernel triangular B-splines, extended isogeometric analysis for solving weakly discontinuous problems, collocation methods using superconvergent points, and B-spline basis in image registration applications. Each topic is oriented towards application of isogeometric analysis basis functions to ease the process of integrating the modeling and analysis phases of simulation.

First, we develop reproducing a kernel triangular B-spline-based FEM for solving PDEs. We review the triangular B-splines and their properties. By definition, the triangular basis function is very flexible in modeling complicated domains. However, instability results when it is applied for analysis. We modify the triangular B-spline by a reproducing kernel technique, calculating a correction term for the triangular kernel function from the chosen surrounding basis. The improved triangular basis is capable to obtain the results with higher accuracy and almost optimal convergence rates.

Second, we propose an extended isogeometric analysis for dealing with weakly discontinuous problems such as material interfaces. The original IGA is combined with XFEM-like enrichments which are continuous functions themselves but with discontinuous derivatives. Consequently, the resulting solution space can approximate solutions with weak discontinuities. The method is also applied to curved material interfaces, where the inverse mapping and the curved triangular elements are considered.

Third, we develop an IGA collocation method using superconvergent points. The collocation methods are efficient because no numerical integration is needed. In particular when higher polynomial basis applied, the method has a lower computational cost than Galerkin methods. However, the positions of the collocation points are crucial for the accuracy of the method, as they affect the convergent rate significantly. The proposed IGA collocation method uses superconvergent points instead of the traditional Greville abscissae points. The numerical results show the proposed method can have better accuracy and optimal convergence rates, while the traditional IGA collocation has optimal convergence only for even polynomial degrees.

Lastly, we propose a novel dynamic multilevel technique for handling image registration. It is application of the B-spline functions in image processing. The procedure considered aims to align a target image from a reference image by a spatial transformation. The method starts with an energy function which is the same as a FEM-based image registration. However, we simplify the solving procedure, working on the energy function directly. We dynamically solve for control points which are coefficients of B-spline basis functions. The new approach is more simple and fast. Moreover, it is also enhanced by a multilevel technique in order to prevent instabilities. The numerical testing consists of two artificial images, four real bio-medical MRI brain and CT heart images, and they show our registration method is accurate, fast and efficient, especially for large deformation problems.

# Contents

- 1 Introduction** **1**
  - 1.1 Background, motivation, and challenges . . . . . 1
    - 1.1.1 Could some triangular elements be suitable for IGA? . . . . . 3
    - 1.1.2 How can IGA be modified to maintain the optimal convergence rate when dealing with weak discontinuous problems? . . . . . 3
    - 1.1.3 How to choose the position of the collocation points for maximize accuracy? . . . . . 4
    - 1.1.4 What are some potential applications of IGA in image processing? . . . . . 5
  - 1.2 Contributions of this work . . . . . 6
  - 1.3 Organization of this thesis . . . . . 7
  
- 2 Isogeometric Analysis** **10**
  - 2.1 Introduction . . . . . 10
  - 2.2 B-spline and NURBS . . . . . 10
  - 2.3 Strong and Weak Form of Poisson’s Equations . . . . . 18
  - 2.4 Linear Elasticity . . . . . 22
  - 2.5 Knot insertion and order elevation . . . . . 27
    - 2.5.1 Knot insertion . . . . . 27
    - 2.5.2 Order elevation . . . . . 28
  - 2.6 Conclusions . . . . . 30

## CONTENTS

---

<b>3</b>	<b>Reproducing Kernel Triangular B-spline-based FEM for Solving PDEs</b>	<b>33</b>
3.1	Introduction . . . . .	33
3.2	Reproducing Kernel Triangular B-spline . . . . .	34
3.2.1	Simplex Splines . . . . .	34
3.2.2	Triangular B-splines . . . . .	37
3.2.3	Reproducing Kernel Triangular B-spline . . . . .	41
3.3	Numerical Examples . . . . .	46
3.3.1	Example I (Triangular Domain) . . . . .	47
3.3.2	Example II (L-shaped Domain) . . . . .	50
3.3.3	Example III (Triangle with A Hole) . . . . .	53
3.4	Conclusions . . . . .	57
<b>4</b>	<b>Extended Isogeometric Analysis for Material Interface Problems</b>	<b>58</b>
4.1	Introduction . . . . .	58
4.2	Extended Isogeometric Analysis . . . . .	58
4.2.1	Enriched Basis Functions Selection . . . . .	58
4.2.2	Enrichment Functions . . . . .	61
4.2.3	Greville Abscissae . . . . .	62
4.2.4	Repeating Middle Neighbour Knots . . . . .	63
4.2.5	Inverse Mapping . . . . .	64
4.2.6	Curve Fitting . . . . .	65
4.2.7	Intersection Points . . . . .	68
4.2.8	Triangular Integration . . . . .	69
4.3	Numerical Examples . . . . .	71
4.3.1	Example I (Bimaterial Interface on a Square) . . . . .	72
4.3.2	Example II (Bimaterial Interface on a Disc) . . . . .	76
4.3.3	Example III (More Gernal Interface Problem on a Disc ) . . . . .	80
4.4	Conclusions . . . . .	84

## CONTENTS

---

<b>5</b>	<b>An Isogeometric Collocation Method Using Superconvergent Points</b>	<b>86</b>
5.1	Introduction . . . . .	86
5.2	IGA Super Collocation Method . . . . .	87
5.2.1	Collocation . . . . .	87
5.2.2	Superconvergent Points for Collocation . . . . .	88
5.3	Numerical Examples . . . . .	93
5.3.1	1D Problem . . . . .	94
5.3.2	Annulus Problem . . . . .	96
5.3.3	Infinite Plate with a Circular Hole with prescribed displacements . . . . .	101
5.3.4	Infinite Plate with a Circular Hole under traction boundary conditions . . . . .	105
5.3.5	3D Cube Problem . . . . .	106
5.3.6	Poisson equation on a hollow sphere . . . . .	109
5.4	Analysis of algorithmic efficiency . . . . .	113
5.4.1	Assembling the linear system . . . . .	113
5.4.2	Solving the linear system . . . . .	116
5.4.3	Computational cost vs. computational accuracy . . . . .	119
5.5	Conclusions . . . . .	121
<b>6</b>	<b>A novel multilevel technique for solving image registration</b>	<b>124</b>
6.1	Introduction . . . . .	124
6.2	Dynamic Mathematical Modeling . . . . .	125
6.3	Multilevel Technique . . . . .	130
6.4	Numerical Examples . . . . .	136
6.4.1	Artificial Images . . . . .	137
6.4.2	MRI Brain Images . . . . .	142
6.4.3	CT Heart Images . . . . .	144
6.5	Conclusions . . . . .	151

## CONTENTS

---

<b>7 Conclusions</b>	<b>153</b>
<b>References</b>	<b>156</b>
<b>Curriculum Vitae</b>	<b>173</b>

# List of Figures

2.1	(a) Cubic B-spline basis functions, (b) the first order derivatives of cubic basis functions, and (c) the second order derivatives of cubic basis functions in an open knot vector $\Xi = \{0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 7, 7, 7\}$ .	12
2.2	Quartic ( $p = 4$ ) basis functions for open, non-uniform knot vector $\Xi = \{0, 0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 5, 5, 5, 5, 6, 7, 7, 7, 7, 7\}$ .	13
2.3	A quadratic B-spline curve (a) and its corresponding B-spline basis functions (b) are defined on a given knotvector $\Xi = \{0, 0, 0, 1/4, 1/4, 2/4, 3/4, 1, 1, 1\}$ . The coordinates of the control points are $\{(1, 0.5), (0.5, 2), (2.5, 5), (4, 3.5), (3, 2), (5, 2.5), (4, 0.5)\}$ .	14
2.4	Linear B-splines basis functions (left) and Quadratic B-splines basis functions (right). The knots vectors for the linear B-splines basis functions are given by $\Xi_1 = \mathcal{H}_1 = \{0, 0, 1/3, 2/3, 1, 1\}$ . The knots vectors for the quadratic B-splines basis functions are given by $\Xi_2 = \mathcal{H}_2 = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$ .	15
2.5	Isoparametric mapping between the parametric space (left) and physical domain (right). The corresponding knots vectors and the control points are shown in Table 2.1	16
2.6		22

## LIST OF FIGURES

---

2.7	Knot insertion. Based on the initial knot vector and the control points given by Figure 2.3, the newly inserted knots are $\{1/8, 3/8, 5/8, 7/8\}$ and the corresponding B-spline curve and the basis functions are shown in (a) and (b). Keeping the current knot vector, we insert a new sequence of knots $\{1/16, 3/16, 5/16, 7/16, 9/16, 11/16, 13/16, 15/16\}$ and obtain the B-spline curve (c) and the basis functions plotted in (d).	31
2.8	Based on the initial knot vector and the control points given by Figure 2.3, we apply the order elevation technique and obtain the identical B-spline curves of $p = 3$ (a), $p = 4$ (c) and $p = 5$ (e). Subfigures (b), (d) and (f) are their corresponding basis functions.	32
3.1	Sketch to illustrate the half-open convex hull.	36
3.2	Examples to illustrate the half-open convex hull $[\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2)$ . The solid points belong to the half-open convex hull, the empty ones do not.	36
3.3	Simplex splines examples. (a) is a simplex spline of degree zero whose knots are $\mathbf{t}_0 = (0.2, 0.2)$ , $\mathbf{t}_1 = (0.7, 0.2)$ and $\mathbf{t}_2 = (0.5, 0.6)$ ; (b) is a linear simplex spline whose knots are $\mathbf{t}_0 = (0.2, 0.2)$ , $\mathbf{t}_1 = (0.6, 0.7)$ , $\mathbf{t}_2 = (0.3, 0.6)$ and $\mathbf{t}_3 = (0.7, 0.1)$ ; and (c) is a quadratic simplex spline whose knots are $\mathbf{t}_0 = (0.2, 0.2)$ , $\mathbf{t}_1 = (0.6, 0.1)$ , $\mathbf{t}_2 = (0.9, 0.3)$ , $\mathbf{t}_3 = (0.7, 0.7)$ and $\mathbf{t}_4 = (0.4, 0.6)$ .	37
3.4	Distribution of triangular B-spline basis for quadratic case.	38
3.5	Normalized quadratic triangular B-spline basis functions.	39
3.6	Distribution of knot net for a problem domain. The red dots are the root knot-nodes. The blue squares are the additional knot-nodes associated to each root knot-node.	41



## LIST OF FIGURES

---

3.7	The numerical example I with a triangular domain. (a) is a triangulation of the triangular domain defined by Equation 3.29; (b) is the exact solution defined by Equation 3.31 in the triangular domain; (c) is the distance error produced by the triangular B-spline-based FEM without the reproducing kernel approximation improvement; and (d) is the distance error from the reproducing kernel triangular B-spline-based FEM. . . . .	47
3.8	Comparison of the convergence rates of the $L^2$ norms between the traditional triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM. . . . .	49
3.9	The numerical example II with an $L$ -shaped domain. (a) is a triangulation of the $L$ -shaped define by Equation 3.32; (b) is the exact solution defined by Equation 3.34 bounded by this $L$ -shaped domain; (c) is the distance error from the triangular B-spline based-FEM without the reproducing kernel approximation improvement; and (d) is the distance error from the reproducing kernel triangular B-spline-based FEM. . . . .	51
3.10	Comparison of the convergence rates of the $L^2$ norms between the traditional triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM. . . . .	52
3.11	The numerical example III governed by a triangle with a hole shaped domain. (a) is a triangulation of the triangle with a hole shaped domain defined in Equation 3.35; (b) is the exact solution defined by Equation 3.37 in this triangle with a hole shaped domain; (c) is the distance error produced by the triangular B-spline-based FEM without the reproducing kernel approximation improvement; and (d) is the distance error from the reproducing kernel triangular B-spline-based FEM. . . . .	54
3.12	Comparison of the convergence rates of the $L^2$ norms between the traditional triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM. . . . .	56

## LIST OF FIGURES

---

4.1	Applying the signed-distance function to distinguish the types of elements. . . . .	60
4.2	the enrichment function $\psi_1(x, y)$ . . . . .	62
4.3	the basis functions distribution before repeating middle neighbour knots (left) after repeating middle neighbour knots (right) . . . . .	64
4.4	The element is crossed by the curved material interface (left). The enriched element is divided by triangulation (right). . . . .	69
4.5	Triangle transformation . . . . .	71
4.6	The function $u$ is continuous, but $\nabla_x u$ is discontinuous along the middle axis. . . . .	73
4.7	Comparison of the $x$ -derivative approximation between two methods. $p = 2$ and $q = 2$ . (a) is the error producing by IGA. (b) is the error producing by XIGA with Ramp enrichment functions $\phi_1$ and $\phi_2$ . (c) is considered Ramp enrichment function and Greville Abcissae but without repeating middle neighbor knots. (d) is considered Ramp enrichment function, Greville Abcissae and also repeating middle neighbor knots. . . . .	75
4.8	Comparison of the convergence rates of the energy norms between IGA and XIGA with different techniques. . . . .	76
4.9	the curved material interface model . . . . .	79
4.10	Comparison of the $x$ -derivative approximation errors between the XIGA enriched by Moës enrichment function but without repeating the middle neighbor knots (left) and the XIGA enriched by Moës enrichment function with repeating the middle neighbor knots. . . . .	79
4.11	Compare the convergence rates of the $L^2$ norm errors between IGA and XIGA. . . . .	80
4.12	The effect of the inverse mapping for a circle . . . . .	81
4.13	Compare the convergence rates of the energy norm of the errors between IGA and XIGA. . . . .	83

## LIST OF FIGURES

---

4.14	Compare the convergence rates of the $L^2$ norm of the errors between IGA and XIGA. . . . .	84
5.1	The 1D problem solved by (a) IGA collocation, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $L^\infty$ -norm error. . . . .	95
5.2	The 1D problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $W^{1,\infty}$ -norm error. . . . .	95
5.3	The 1D problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $W^{2,\infty}$ -norm error. . . . .	96
5.4	Domain $\Omega$ for the quarter of an annulus problem . . . . .	97
5.5	Greville abscissae points in (a) parametric domain and (b) physical domain for $p = 4$ ; (c-d) IGA-SC collocation points in the parametric domain and the physical domain for $p = 4$ and (e-f) $p = 5$ . The black lines are the knot (mesh) lines and the blue asterisks represent the collocation points. . . . .	99
5.6	The quarter annulus problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. $\log_{10}(\text{relative error}) = \log_{10}(\ u_{ex} - u_h\ _{L^\infty} / \ u_{ex}\ _{L^\infty})$ . . . .	100
5.7	The quarter annulus problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. $\log_{10}(\text{relative error}) = \log_{10}( u_{ex} - u_h _{W^{1,\infty}} /  u_{ex} _{W^{1,\infty}})$ . . . .	100
5.8	The quarter annulus problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. $\log_{10}(\text{relative error}) = \log_{10}( u_{ex} - u_h _{W^{2,\infty}} /  u_{ex} _{W^{2,\infty}})$ . . . .	100
5.9	$\sigma_{xx}$ plots of the 2D elastic plate with a circular hole (a) analytic solution, (b) Error of the IGA-G, (c) Error of the IGA-C and (d) Error of the IGA-SC . The applied force is $T_x = 10$ . The polynomial degree of the basis is $p = 5$ and there are $16 \times 32$ elements ( $h = 1/16$ in the parametric space). . . . .	103
5.10	The elastic plate with a circular hole problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $L^2$ -norm error. . . . .	104

## LIST OF FIGURES

---

5.11	The elastic plate with a circular hole problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $H^1$ -norm error. . . . .	104
5.12	The elastic plate with a circular hole problem solved by (a) IGA-C, (b) IGA-SC and (d) IGA-G. The relative error is calculated by $H^1$ -norm error. . . . .	106
5.13	Unit cube model problem. (a) exact solution $u$ , (b) error of the IGA-G $u - u_h^G$ , (c) error of the IGA-C $u - u_h^C$ and (d) error of the IGA-SC $u - u_h^{SC}$ . The B-spline basis functions are of polynomial degree 5 in each direction. The domain is discretized by $10 \times 10 \times 10$ elements. Shown is the section of the cube with the volume coordinates $[0, 1] \times [0.1, 0.9] \times [0.2, 0.8]$ . . . . .	107
5.14	Unit cube problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $L^\infty$ -norm error. . . . .	108
5.15	Unit cube problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $W^{1,\infty}$ -norm error. . . . .	108
5.16	Unit cube problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $W^{2,\infty}$ -norm error. . . . .	108
5.17	8 <sup>th</sup> of a hollow sphere . . . . .	110
5.18	Hollow sphere problem with the solution and errors plotted on the mid-surface of the sphere (at radius $r = R$ ). (a) exact solution $u$ , (b) relative error of the IGA-G $(u - u_h^G) /  u $ , (c) relative error of the IGA-C $(u - u_h^C) /  u $ and (d) relative error of the IGA-SC $(u - u_h^{SC}) /  u $ . The B-spline basis functions are of polynomial degree 5 in each direction. The domain is discretized by $10 \times 10 \times 10$ elements. . . . .	111
5.19	Hollow sphere problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $L^2$ -norm error. . . . .	112
5.20	Hollow sphere problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $H^1$ -norm error. . . . .	112

## LIST OF FIGURES

---

5.21	Hollow sphere problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G. The relative error is calculated by $H^2$ -norm error. . . . .	112
5.22	The sparsity pattern and number of non-zero entries of the left hand side matrix of the linear system for the three methods considered, with $p = 4, d = 2, n = 30$ . . . . .	117
5.23	time vs. relative error of the 2D annulus example. . . . .	122
5.24	Accuracy-to-time comparison of the 3D sphere example. . . . .	123
6.1	Registration results from an ellipsoid to a rectangle. The reference image and the target image are shown in (a) and (b). The deformed meshes are shown in (c-f). The corresponding deformed images are shown in (g-j). (k-n) are the inverse registration results from the rectangle to the ellipsoid. . . . .	135
6.2	A comparison between the single level methods and the multilevel method. . . . .	136
6.3	Register a gear-shaped object (b) from a star-shaped one (a). (c) and (d) are the first level registration results. (e) is the second level registration result. The initial differences between (a) and (b) and the final differences between (e) and (b) are shown in (f) and (g), respectively. .	140
6.4	Register a three-star image (b) from a disk image (a). (c-e) are the registration results at the first level, the second level and the third level, respectively. (f) shows the initial differences between (a) and (b). (g) shows the final differences between our registration result (e) and the target image (b). . . . .	141
6.5	Registration results from method in <sup>1</sup> . (a) is the result after the 90 <sup>th</sup> iteration steps with $44 \times 44$ control points. (b) is the result after the 90 <sup>th</sup> iteration step with $54 \times 54$ control points. (c) shows the differences between (a) and the target image. (d) shows the differences between (b) and the target image. . . . .	142

## LIST OF FIGURES

---

6.6	Brain image 1. Register the target image (b) from the reference image (a). (c-f) are the first level registration results with different iteration steps. (g) and (h) are the fifth and sixth level results, respectively. (i) shows the initial differences between (a) and (b), and (j) shows the differences between (h) and (b). . . . .	145
6.7	Brain image 2. Register the target image (b) from the reference image (a). (c-f) are the first level registration results with different iteration steps. (g) and (h) are the fifth and sixth level results, respectively. (i) shows the initial differences between (a) and (b), and (j) shows the differences between (h) and (b). . . . .	146
6.8	Brain image 3. Register the target image (b) from the reference image (a). (c-f) are the first level registration results with different iteration steps. (g) and (h) are the fifth and sixth level results, respectively. (i) shows the initial differences between (a) and (b), and (j) shows the differences between (h) and (b). . . . .	147
6.9	Registration results from method in <sup>1</sup> . (a) is the result after getting stopped in the 25 <sup>th</sup> iteration step with $54 \times 54$ control points. (b) is the result after getting stopped in the 50 <sup>th</sup> iteration step with $64 \times 64$ control points. (c) shows the differences between (a) and the target image. (d) shows the differences between (b) and the target image. . . . .	148
6.10	Cardiac image 1. The reference and target images are shown in (a) and (b). (c-e) are our registration results during the first level. (g) and (h) are solved after the fifth level and sixth level. (i) is the initial differences, and (j) shows the differences between our final solved result (h) and (b). . . . .	149

## LIST OF FIGURES

---

6.11 Cardiac image 2. The reference and the target images are shown in (a) and (b). (c) is the initial differences between (a) and (b). (d) is the registration result using the method in <sup>1</sup> . (e) shows the differences between (d) and (b). (f) is the registration result using our method, and (g) shows the differences between (f) and (b). . . . .	150
--	-----

# List of Tables

2.1	Knots vectors and control points for the example of Figure 2.5 . . . . .	17
3.1	$L^2$ norm of the error by the triangular B-spline . . . . .	49
3.2	$L^2$ norm of the error by the reproducing kernel triangular B-spline . . . . .	50
3.3	$L^2$ norm of the error by the triangular B-spline . . . . .	53
3.4	$L^2$ norm of the error by the reproducing kernel triangular B-spline . . . . .	53
3.5	$L^2$ norm of the error by the triangular B-spline . . . . .	55
3.6	$L^2$ norm of the error by the reproducing kernel triangular B-spline . . . . .	55
4.1	Energy Norm of the Errors (Ramp Enrichment + Repeating Middle Neighbour Knots + Greville Abscissae) . . . . .	74
4.2	The Initial Control Points (the third coordinates are weights) . . . . .	78
4.3	$L^2$ Norm of the Errors (Moës Enrichment + Repeating Middle Neighbor Knots) . . . . .	78
4.4	Energy Norm of the Errors with IGA . . . . .	82
4.5	Energy Norm of the Errors with XIGA . . . . .	82
4.6	$L^2$ Norm of the Errors with IGA . . . . .	82
4.7	$L^2$ Norm of the Errors with XIGA . . . . .	83
5.1	The superconvergent points for B-spline basis of degree $p$ on interval $[-1,1]$ . . . . .	91
5.2	Number of flops for assembling the linear system in IGA-C, IGA-SC and IGA-G . . . . .	115



## LIST OF TABLES

---

5.3	Lower/Upper bandwidth of the linear system matrices in IGA-C, IGA-SC and IGA-G for solving a scalar problem with $n$ knot-spans in each direction and polynomial degree $p$ , when $n \gg p$ . . . . .	118
5.4	The convergence rates of the current numerical methods . . . . .	120
6.1	Similarity ratios ( $R_s$ ) and running time for all examples studied . . .	137
6.2	Comparison of Our Method with the Method in <sup>1</sup> in terms of similarity ratios ( $R_s$ ), mean square difference (MSD), and running time. . . . .	139

# Nomenclature

Symbols and abbreviations used throughout most of the thesis are listed. Symbols less frequently used, or that have different meanings in different contexts, are defined where they are used.

<b>Symbols</b>	<b>Description</b>
$\Delta$	Laplace operator or Laplacian
$\nabla$	Gradient operator
$\nabla \cdot$	Divergence operator
$\nabla_{(x,y)}$	Gradient operator is applied respect to the physical coordinates $(x,y)$
$\Gamma_e$	Dirichlet boundary or essential boundary
$\Gamma_n$	Neumann boundary or natural boundary
$\Gamma$	Whole boundary of an open set
$\Omega$	An open domain with Lipschitz continuous boundary
$\partial\Omega$	Whole boundary of an open set
$J$	Jacobian matrix
$\mathbf{K}$	Material property matrix
$\mathcal{S}$	An infinite-dimensional set
$\mathcal{V}$	An infinite-dimensional set
$\mathcal{S}^h$	A finite-dimensional set
$\mathcal{V}^h$	A finite-dimensional set
$\in$	Set membership
$\notin$	Negation of set membership
$E$	Elastic Young's modulus
$\nu$	Poisson's ration
$\sigma$	Stress
$\varepsilon$	Strain
$\bar{\mathbf{u}}$	Prescribed boundary displacement

$\bar{\mathbf{t}}$	Prescribed boundary traction
$\mathbf{n}$	Outward unit normal vector at the boundary
$N_{i,p}$	The $i^{\text{th}}$ basis function with polynomial degree $p$
$N_{i,p}^k$	The $k^{\text{th}}$ differential of the $i^{\text{th}}$ basis function with polynomial degree $p$
$\Xi$	A knotvector
$\xi_i$	The $i^{\text{th}}$ knot in $\Xi$
$\mathbf{P}$	Control Point
$\lambda$	Barycentric coordinate
$\det(\cdot)$	Determinant
$[\cdot]$	Convex hull
$[\cdot)$	Half open convex hull
$\delta_{i,j}$	Kronecker delta
$\mathcal{L}$	A differential operator
$\mathcal{G}$	A vector operator

<b>Abbreviations</b>	<b>Description</b>
CAD	Computer-Aided Design
CT	Computed Tomography
DOF	Degree(s) of Freedom
FEM	Finite Element Method
FEA	Finite Element Analysis
IGA	Isogeometric Analysis
IGA-C	Isogeometric Analysis Collocation
IGA-G	Isogeometric Analysis Galerkin
IGA-SC	Isogeometric Analysis Superconvergent Collocation
MRI	Magnetic Resonance Imaging
NURBS	Non-uniform Rational B-splines
PU	Partition of Unity

PDE	Partial Differential Equation
RKPM	Reproducing Kernel Approximation Method
XFEM	Extended Finite Element Method
XIGA	Extended Isogeometric Analysis

# Chapter 1

## Introduction

### 1.1 Background, motivation, and challenges

Isogeometric analysis (IGA) is a numerical method introduced by Hughes *et al.*<sup>2,3</sup> in 2005. The initial motivation of IGA is to bridge computer-aided design (CAD) and finite element analysis (FEA), applying the same basis functions for both geometric design and the PDEs analysis. IGA is usually based on NURBS (Non-uniform Rational B-splines) basis functions, since NURBS represent the standard geometries in contemporary CAD. The high continuity (up to  $C^{p-1}$  for a basis of polynomial degree  $p$ ) is the most distinct and important property of NURBS, thus NURBS can give a very smooth geometry representation. IGA has been shown to be more accurate than  $p$ -FEM on a degree of freedom basis<sup>4,5,6</sup>. In addition, it is possible to obtain more accurate, even exact, geometry representations on very coarse meshes which facilitate numerical analysis<sup>7</sup>. It has been proven that robustness of isogeometric elements is increased with order<sup>8</sup>, while the high order Lagrange elements are sensitive to mesh distortion. Furthermore, the smooth surface representation and the non-negative basis functions advantages of IGA make it superior to FEM when dealing with contact problems<sup>9,10,11</sup>.

The method has been successfully used in a variety of applications such as plate and

## 1.1 Background, motivation, and challenges

---

shell problems<sup>12,13,14,15,16</sup>, fluid mechanics<sup>17,18,19,20</sup>, structural analysis<sup>6,21</sup>, phase-field analysis<sup>22,23,24,25</sup>, shape optimization<sup>26,27,28</sup> and electromagnetism<sup>29</sup>. Another approach to bridge the geometry generation and the analysis is T-splines based IGA approaches<sup>30,31,32,33</sup>, since T-splines have shown prominent application on isogeometric discretization<sup>34,35</sup>, and also developed in several commercial software such as Rhino<sup>36</sup>. So far, NURBS-based IGA has been mostly used in the framework of the Galerkin method (<sup>2,3,6,21</sup>) which handles weak forms of PDEs, but recently collocation approaches (<sup>37,38,39,40</sup>) which deal with strong forms of PDEs are receiving growing attention. Moreover, the traditional IGA methods have been augmented by many other techniques. For example, the enrichment functions from the extended finite element method (XFEM)<sup>41,42,43,44,45,46</sup> have been successfully applied in IGA, the resulting method is commonly called the extended isogeometric analysis (XIGA)<sup>47,48,49</sup>. It has shown good performance in modeling discontinuous problems such as fracture and material interface. Recently, considering the computing efficiency issue, reduced quadrature techniques<sup>50,51</sup> have also been used in IGA and can apply to solve large computation problems. Despite many achievements made by IGA in recent years, several still open questions or topics related to IGA should be pursued. In this dissertation, we try to answer the four following questions:

- Could some triangular elements be suitable for IGA?
- How can IGA be modified to maintain the optimal convergence rate when dealing with discontinuous problems?
- How to choose the position of the collocation points to maximize accuracy?
- What are some potential applications of IGA in image processing?

In the following paragraphs, we would like to address each question, explaining the background, motivation and challenges.

## 1.1 Background, motivation, and challenges

---

### 1.1.1 Could some triangular elements be suitable for IGA?

Until now, almost all the IGA studies are based on quadrilateral and hexahedral elements. This is because the most commonly used basis functions of IGA are NURBS or B-splines which are based on the tensor product definition and require rectangular elements. However, it is still a challenge to model complex geometries that do not fit in a tensor product simulation. This limitation could be overcome by introducing triangular elements into IGA. Triangular B-splines are a novel type of B-splines, which were first proposed in<sup>52</sup>. They are derived from simplex splines and B-patches, and inherit many of their features and nice properties. This kind of B-spline has been developed for applications in computer graphics<sup>53</sup>. Compared to NURBS and T-splines, triangular B-splines can model arbitrary features due to their special triangular mesh structure. This is a very important advantage which can extend the use of IGA in a lot of applications. However, the standard triangular B-spline is difficult to use in analysis.

### 1.1.2 How can IGA be modified to maintain the optimal convergence rate when dealing with weak discontinuous problems?

Problems with material interfaces where the solution is only  $C^0$ -continuous solutions are common in both computational and material engineering. The discontinuity in the gradient field occurs because of the different material properties on the two sides of the material interface. These are also known as weak discontinuities, and they are not accurately modeled by the standard FEM without a conforming mesh. Matching the material geometry at the interface as closely as possible would improve the accuracy of the solution. However, it has been widely accepted that creating an adequate conforming mesh is difficult or cumbersome for problems where the material interface is curved. To solve this class of problems, several advanced methods have been proposed in the past few years. A straight forward approach is to extend the FEM, i.e the Extended Finite Element Method (XFEM)<sup>41,54</sup>, which has been developed to model discontinuities without remeshing or with minimum mesh construction. The

## 1.1 Background, motivation, and challenges

---

basic idea of this method is to add local enrichment functions into the approximation space similar to Partition of Unity (PU)<sup>55</sup> methods. Other PU methods employ meshless shape functions (see<sup>56,57,58,59</sup>) and take advantage of their higher continuity<sup>60,61,62</sup>. More recently, several researchers have developed the Extended Isogeometric Analysis (XIGA)<sup>47,48</sup>. The XIGA methods developed so far were aimed at solving problems with strong discontinuities, such as fracture. To the best of our knowledge, there has been no XIGA implementation for weakly discontinuous problems. There are also some special techniques needed to obtain optimal convergence rates in the context of XIGA.

### 1.1.3 How to choose the position of the collocation points for maximize accuracy?

Most of the commonly used PDE solvers are based on the so-called weak (or variational form) of the PDE. Gaussian quadrature rules are most commonly used, however, as the polynomial degree of the approximation increases, more Gauss points are needed to accurately evaluate the resulting integrals. For the IGA method in particular, Gauss quadrature does not fully take into account the higher continuity of the basis functions which has led to the development of reduced-quadrature methods<sup>50</sup> and other customized quadrature rules<sup>63,64,65,66,67</sup>. A more promising approach, which eliminates integration, is to work directly with the strong form of the PDE. The IGA collocation method<sup>37,38,39,68,69</sup> has been studied since 2010 and, to a large extent, it is combining the accuracy and the smoothness advantages of the IGA method with the computational efficiency of the collocation method. Since there are no volume integrals in the IGA collocation, the method is considerably cheaper from a computational point of view. It is also relatively easy to implement, since it only requires point evaluation of the shape functions and the right-hand side at the chosen collocation points. The boundary conditions are imposed as additional constraints in the linear system, which is typically non-symmetric even for self-adjoint problem but more sparse compared to



## 1.1 Background, motivation, and challenges

---

the Galerkin method. Until now, the mathematical theory of collocation has not been very well developed. It has been shown only in 1-dimension that the isogeometric collocation method has optimal convergence of  $O(h^{p-1})$  for the 2<sup>nd</sup> derivative norms, where  $p$  is the polynomial degree and  $h$  is the maximum element length. However, in the existing numerical studies<sup>37,39</sup>, it has been observed that the convergence rate for the first derivative norms when  $p$  is odd is also of the order  $O(h^{p-1})$ , which is suboptimal compared to the IGA Galerkin.

### 1.1.4 What are some potential applications of IGA in image processing?

Image registration techniques, developed in recent years, aim to align two images by finding a spatial transformation<sup>70,71,72</sup> between them. These methods fall mainly into three basic categories, the landmark-based registration<sup>73,74,75</sup>, the segmentation-based registration<sup>76,77,78</sup> and the image intensity-based registration<sup>79,80</sup>. They have many applications such as medical imaging<sup>81,82</sup>, remote sensing<sup>83,84</sup> and computer vision<sup>85</sup>. In addition to various types of spatial transformations, image registration approaches can be divided into rigid registration and non-rigid (or deformable) registration. From a theoretical point of view, the intensity-based registration methods are the most flexible because they use all the available information throughout the registration process. Recently, an intensity-based registration method using FEM was proposed<sup>1</sup>, and it has shown promising performance, demonstrating more accurate results than a traditional linear interpolation method and an optical flow-based method. But the computational requirements for the proposed method<sup>1</sup> are more intensive than the others. This is because the matrix assembly and solving the resulting large system are expensive.

## 1.2 Contributions of this work

Since 2005 when IGA was first proposed, there have been many researchers who added contributions to this advanced numerical method. According to the motivations presented in Section 1.1, we have made some advancements. We briefly summarize our contributions as follows:

- We develop a novel way to deal with the instabilities of the triangular B-spline based on the reproducing kernel approximation methods<sup>86,87,88,89</sup>. Adopting the compact support property of the triangular B-spline, we choose the non-zero subset basis functions and apply the reproducing kernel technique to calculate the correction term for the triangular kernel function. In addition, we further study the reproducing kernel triangular B-spline in solving the Poisson's equation, and is also compared with the original triangular B-spline. The results show an improvement of the triangular B-spline performance with an almost optimal convergence rate.
- We develop an optimal XIGA method to solve material interface problems, by combining the advantages between XFEM and IGA. We show that the XIGA achieves optimal convergence rates while the IGA only converges at suboptimal rates for the Poisson's equation with weakly discontinuous solutions. The method is implemented for solving three numerical test problems including bi-material and curved material interface problems, and is also compared with the traditional IGA. The results show much better approximation properties and an optimal convergence rate.
- We improve the accuracy and the convergence rate of the existing IGA collocation by selecting the collocation point at the zeros of a polynomial defined on a reference interval and scaled to each knot-span. This method is similar to the orthogonal collocation method for B-splines developed by<sup>90,91</sup>, where the collocation points are chosen at the Gauss points (or the roots of the Legendre poly-

### 1.3 Organization of this thesis

---

nomials). However, this requires that each knot is repeated  $p - 1$  times which means that more degrees of freedom are required and the approximation scheme is only  $C^1$  continuous. To preserve the smoothness of the computed solution, we choose instead the collocation points at the so-called “superconvergent points” for the 2<sup>nd</sup> derivative in the Galerkin method. This allows an approximation that is close in accuracy to the Galerkin approximation, but at the cost of extra basis function evaluations compared to the standard collocation. An efficiency analysis indicates the added cost of the method (in terms of computational time) is offset by the increased accuracy for odd polynomial degrees. Therefore, this method can be used to complement standard collocation, which has optimal convergence only for even degrees of the polynomial basis.

- We present a relatively simple approach to deal with image registration problems. It does not require the assembly and solution of a large matrix system because we are working on the energy functional directly. Therefore, the method performs much faster. A novel multilevel technique is first combined with a dynamic model, making the model more flexible, and the presented technique handles large deformable registration very well, while at the same time being more stable and efficient. The large deformations are interpreted as large differences between the target image and reference image in our current study.

### 1.3 Organization of this thesis

Chapter 2 will introduce the idea of IGA, using the basic equations of Poisson’s and linear elasticity examples. We also interpret the B-spline, NURBS basis functions and their properties. Knot insertion and order elevation techniques are also reviewed in the chapter.

Chapter 3 presents a particular triangular B-spline and its application on dealing with PDEs. The triangular B-spline is very flexible for analysis applications because of its particular definition. However, this property also results in less accuracy and

### 1.3 Organization of this thesis

---

there are no specific and optimal ways of countering it. This drawback makes it unsuitable for analysis. We apply the reproducing kernel approximation method to the triangular B-spline, choosing the surrounding basis and calculating a correction term for the triangular kernel function. The enhanced basis has been tested on three numerical examples for solving the Poisson's equation, and the results show significant improvement with almost optimal convergence rate.

Chapter 4 proposes a XIGA method for solving the material interface problems. The original IGA basis space is augmented by special functions such as the ramp and Moës enrichment functions. We present several techniques to solve curved material interface problems. Three numerical examples show the proposed XIGA method obtains optimal convergence rates and outperforms the pure IGA method.

Chapter 5 focuses on IGA collocation (IGA-C), a “strong form”-based method, which is different from the general “weak form”-based IGA Galerkin (IGA-G) method. It is simple to implement, and it has been shown to be more efficient than IGA-G in terms of computational time. However, the accuracy is a drawback of the standard collocation method is less accuracy compared with strong form methods. One reason is that the position of collocation points is needs to be carefully chosen. Different from the Greville abscissae collocation normally used, we apply IGA collocation to the superconvergent points (IGA-SC) for the Galerkin method. This allows the approximation results of IGA-SC to be close to those of IGA-G. The numerical examples show that IGA-SC is much more accurate than IGA-C, especially for odd polynomial degrees whereas IGA-C has optimal convergence only for even cases.

Chapter 6 explores the B-spline basis in image registration applications. The aim is to align a reference image to a target image through a spatial transformation. We develop a dynamic multilevel technique to search for the control points which are the to-be-determined coefficients of the spatial transformation. The method uses the same energy function applied in a FEM-based image registration, but works on the energy function directly, saving the assembly and solving of a large linear system. The simplified method is relatively simple and fast. In addition, a multilevel technique

### **1.3 Organization of this thesis**

---

is used to deal with the stability. Several artificial images and real bio-medical MRI brain and CT heart image examples show the method is stable, accurate and efficient, especially in handling large deformation problems.

Chapter 7 contains the conclusions of the dissertation. We will briefly review the contents and summarize the main achievements, ending with some discussions of the future work.

# Chapter 2

## Isogeometric Analysis

### 2.1 Introduction

IGA is a recently developed computational approach to integrate the design and modeling phase of product development. The basis functions in IGA are B-splines or NURBS (Non-uniform rational B-splines). The basis functions are used for an isoparametric mapping between the parametric space (unit square) and the physical space (which may have complex geometry). In this chapter, we will review the basics of IGA, including the commonly used B-splines and NURBS basis functions and the isoparametric mapping. We will also show two example problems to illustrate IGA in analysis. The chapter will end by a review of  $h$ -refinement (knot insertion) and  $p$ -refinement (order elevation) of IGA.

### 2.2 B-spline and NURBS

Non-uniform rational B-splines (NURBS) are the basis functions widely used in CAD and also used for analysis in IGA<sup>2</sup>. They are weighted rational B-spline functions. Let  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$  be a non-decreasing sequence of real numbers.  $p$  is the polynomial degree of the basis and  $n$  is the number of basis functions. The  $\xi_i$  for

## 2.2 B-spline and NURBS

---

$i = 1, \dots, n + p + 1$  are called knots, and  $\Xi$  is the knot vector. Then an univariate B-spline basis function can be defined using a recurrence relation. We start with:

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

which is a basis of piecewise constant ( $p = 0$ ) functions. If  $\xi_i = \xi_{i+1}$ , then  $N_{i,0} = 0$ . For  $p > 0$ ,

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

In Equation (2.2) and subsequent discussion, the denominators could be zero, in which case, the quotient term is defined to be zero.

The derivative of a B-spline function is given by

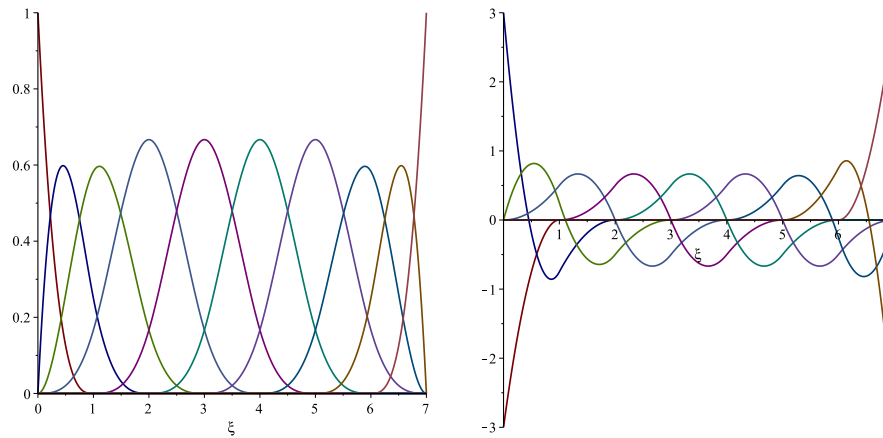
$$\frac{dN_{i,p}(\xi)}{d\xi} = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \quad (2.3)$$

For the  $k^{\text{th}}$ -derivative of  $N_{i,p}(\xi)$ , by repeatedly differentiating Equation (2.3) we get the general formula

$$N_{i,p}^{(k)}(\xi) = p \left( \frac{N_{i,p-1}^{(k-1)}(\xi)}{\xi_{i+p} - \xi_i} - \frac{N_{i+1,p-1}^{(k-1)}(\xi)}{\xi_{i+p+1} - \xi_{i+1}} \right). \quad (2.4)$$

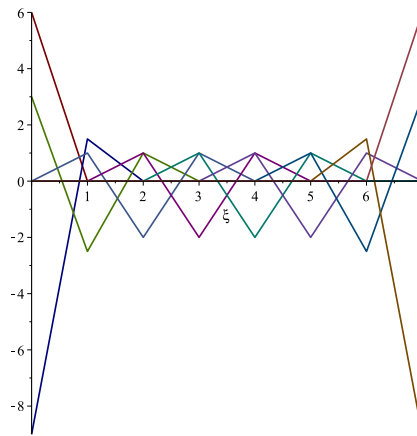
The derivative order  $k$  should not exceed the polynomial order  $p$  since all higher derivatives are zero. A 1D cubic B-spline basis and their corresponding derivatives are plotted in Figure 2.1.

## 2.2 B-spline and NURBS



(a) B-spline basis functions

(b) the first order derivatives



(c) the second order derivatives

Figure 2.1: (a) Cubic B-spline basis functions, (b) the first order derivatives of cubic basis functions, and (c) the second order derivatives of cubic basis functions in an open knot vector  $\Xi = \{0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 7, 7, 7\}$ .

We now list several important properties of the B-spline basis functions. Suppose we have a basis of B-spline functions of polynomial degree  $p$ , defined by a knotvector  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ .

- Local support property:  $N_{i,p} = 0$  if  $\xi \notin [\xi_i, \xi_{i+p+1})$ . For example, in Figure 2.2  $N_{1,4}(\xi)$  has only zero functional value when  $\xi$  are chosen out of the interval



## 2.2 B-spline and NURBS

$[\xi_1, \xi_6)$  where  $\xi_1 = 0$  and  $\xi_6 = 1$  in this example.

- Nonnegativity:  $N_{i,p} \geq 0$  for all  $i, p$  and  $\xi$ .
- Partition of unity:  $\sum_{i=1}^n N_{i,p}(\xi) = 1$  for all  $\xi \in [\xi_1, \xi_n]$ .
- In any given knot span  $[\xi_i, \xi_{i+1})$ , at most  $p + 1$  of  $\{N_{i,p}(\xi)\}$  are nonzero.
- $N_{i,p}(\xi)$  is  $p - k$  times continuously differentiable at a knot of multiplicity  $k$ . For example, in Figure 2.2  $N_{1,4}(\xi)$  has  $C^{-1}$  continuity at  $\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 0$ .  $N_{5,4}(\xi)$  has  $C^2$  continuity at  $\xi_7 = \xi_8 = 2$ .  $N_{7,4}(\xi)$  has  $C^1$  continuity at  $\xi_9 = \xi_{10} = \xi_{11} = 3$ .  $N_{12,4}(\xi)$  has  $C^0$  continuity at  $\xi_{13} = \xi_{14} = \xi_{15} = \xi_{16} = 5$ .
- For  $p > 0$ ,  $N_{i,p}(\xi)$  attains exactly one maximum value.

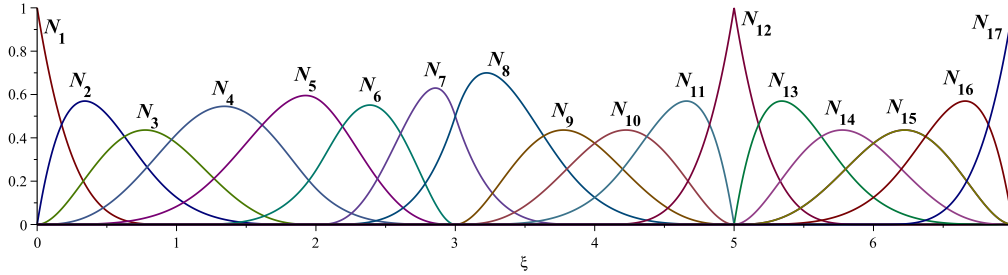


Figure 2.2: Quartic ( $p = 4$ ) basis functions for open, non-uniform knot vector  $\Xi = \{0, 0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 5, 5, 5, 5, 6, 7, 7, 7, 7, 7\}$ .

Based on B-spline basis functions, we can build B-spline curves. A  $p^{\text{th}}$ -degree B-spline curve is defined by

$$\mathbf{C}(\xi) = \sum_i N(\xi_{i,p}) \mathbf{P}_i \quad a \leq \xi \leq b, \quad (2.5)$$

where the  $\{\mathbf{P}_i\}$  are the control points, and the  $\{N_{i,p}(\xi)\}$  are the  $p^{\text{th}}$  degree B-spline basis function (2.2), which are defined on an open knot vector  $\Xi = \underbrace{\{a, \dots, a\}}_{p+1}, \dots, \underbrace{\{b, \dots, b\}}_{p+1}$ .

## 2.2 B-spline and NURBS

An example plots of a B-spline curve and their corresponding basis functions are given in Figure 2.3. Notice that besides the beginning and ending control points being interpolated, the third control point is also interpolated (Figure 2.3 (a)) because of the third basis function has  $C^0$  continuity at knot  $1/4$  (Figure 2.3 (b)).

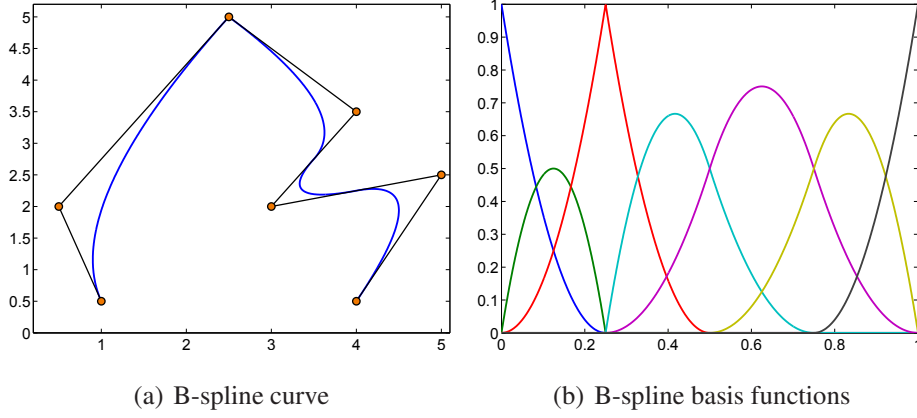


Figure 2.3: A quadratic B-spline curve (a) and its corresponding B-spline basis functions (b) are defined on a given knotvector  $\Xi = \{0, 0, 0, 1/4, 1/4, 2/4, 3/4, 1, 1, 1\}$ . The coordinates of the control points are  $\{(1, 0.5), (0.5, 2), (2.5, 5), (4, 3.5), (3, 2), (5, 2.5), (4, 0.5)\}$ .

The NURBS basis is defined as:

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{W(\xi)} = \frac{N_{i,p}(\xi)w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi)w_{\hat{i}}}, \quad (2.6)$$

where  $w_i$  is the weight corresponding to the  $i^{\text{th}}$  B-spline function  $N_{i,p}(\xi)$ . The 2D and 3D NURBS basis can be defined analogously, for the parametric coordinates  $\eta, \zeta$ , and associated polynomial degrees  $q$  and  $r$ :

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{\sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)w_{\hat{i},\hat{j}}}, \quad (2.7)$$

## 2.2 B-spline and NURBS

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}}{\sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^l N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)L_{\hat{k},r}(\zeta)w_{\hat{i},\hat{j},\hat{k}}}, \quad (2.8)$$

where  $M_{j,q}(\eta)$  and  $L_{k,r}(\zeta)$  are the B-spline basis defined in the  $\eta$  and  $\zeta$  directions. Note that B-splines are NURBS where all the weights are equal to 1. Two example plots of 2D B-spline basis are shown in Figure 2.4. NURBS basis functions have some desirable properties inherited from B-splines, such as nonnegativity, completeness, local support, strong convex hull, etc. Readers interested in more details are referred to 92,93 .

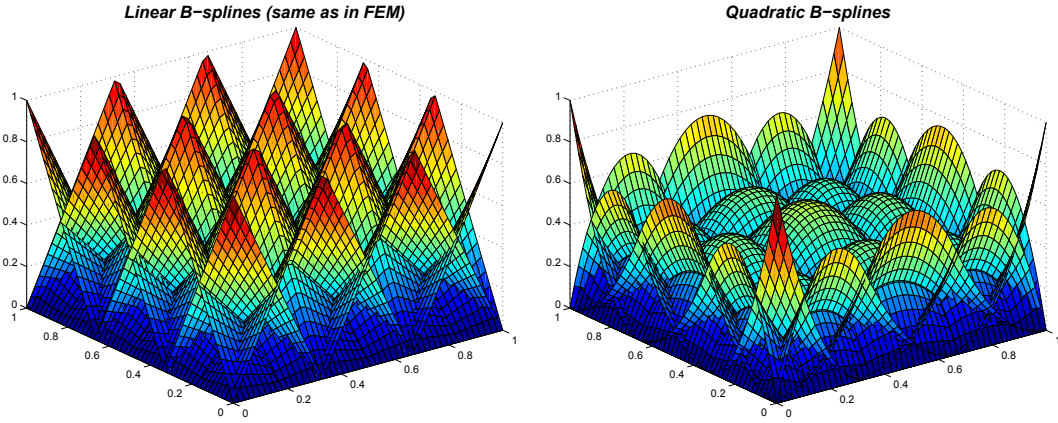


Figure 2.4: Linear B-splines basis functions (left) and Quadratic B-splines basis functions (right). The knots vectors for the linear B-splines basis functions are given by  $\Xi_1 = \mathcal{H}_1 = \{0, 0, 1/3, 2/3, 1, 1\}$ . The knots vectors for the quadratic B-splines basis functions are given by  $\Xi_2 = \mathcal{H}_2 = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$ .

The NURBS shape functions are used for an isoparametric mapping between a parametric space (unit square) and a physical space (which may have a complex geometry). The physical space is the space in which the actual model exists. The parametric space of IGA is different from that of FEM. The elements in the parametric space of FEM are usually called parent elements or reference elements which are mapped to each single element in the physical space, and each element in the physical space has its own mapping. Conversely, the parametric space in IGA is identified with patches

## 2.2 B-spline and NURBS

---

which are made up of multiple elements. Each element in the physical space is the image of a corresponding element in the parametric space, but the mapping is carried out between a whole patch and a group of physical elements together. For a given parametric space defined by the coordinates  $(\xi, \eta, \zeta)$  and a physical space given by the usual  $(x, y, z)$  coordinates, then the isoparametric mapping can be written as (in 3D):

$$\mathbf{F}(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) \mathbf{C}_{i,j,k} = [x, y, z]^T, \quad (2.9)$$

where  $R_{i,j,k}^{p,q,r}$  are the NURBS basis functions and  $\mathbf{C}_{i,j,k}$  are the corresponding control points defined in the physical space. We give an example of isoparametric mapping in Figure 2.5. There a  $4 \times 4$  patch in the parametric space is mapped to a disk in the physical space. The knots vectors and control points used in the mapping are given in Table 2.1. The red colored control points in Table 2.1 are repeated control points.

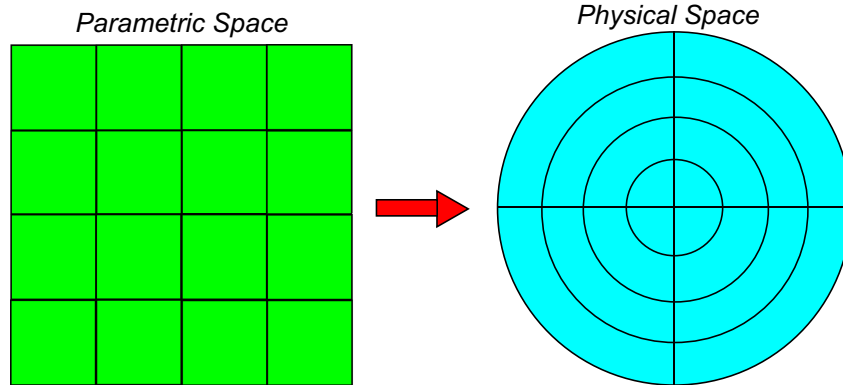


Figure 2.5: Isoparametric mapping between the parametric space (left) and physical domain (right). The corresponding knots vectors and the control points are shown in Table 2.1

## 2.2 B-spline and NURBS

Table 2.1: Knots vectors and control points for the example of Figure 2.5

knots vectors								
$\Xi$	{0, 0, 0, 1/4, 1/4, 1/2, 1/2, 3/4, 3/4, 1, 1, 1}							
$\mathcal{H}$	{0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1}							
control points and weights								
x-coord	y-coord	weight	x-coord	y-coord	weight	x-coord	y-coord	weight
0	-1	1	0	-0.875	1	0	-0.625	1
1	-1	$\frac{\sqrt{2}}{2}$	0.875	-0.875	$\frac{\sqrt{2}}{2}$	0.625	-0.625	$\frac{\sqrt{2}}{2}$
1	0	1	0.875	0	1	0.625	0	1
1	1	$\frac{\sqrt{2}}{2}$	0.875	0.875	$\frac{\sqrt{2}}{2}$	0.625	0.625	$\frac{\sqrt{2}}{2}$
0	1	1	0	0.875	1	0	0.625	1
-1	1	$\frac{\sqrt{2}}{2}$	-0.875	0.875	$\frac{\sqrt{2}}{2}$	-0.625	0.625	$\frac{\sqrt{2}}{2}$
-1	0	1	-0.875	0	1	-0.625	0	1
-1	-1	$\frac{\sqrt{2}}{2}$	-0.875	-0.875	$\frac{\sqrt{2}}{2}$	-0.625	-0.625	$\frac{\sqrt{2}}{2}$
0	-1	1	0	-0.875	1	0	-0.625	1
0	-0.375	1	0	-0.125	1	0	0	1
0.375	-0.375	$\frac{\sqrt{2}}{2}$	0.125	-0.125	$\frac{\sqrt{2}}{2}$	0	0	$\frac{\sqrt{2}}{2}$
0.375	0	1	0.125	0	1	0	0	1
0.375	0.375	$\frac{\sqrt{2}}{2}$	0.125	0.125	$\frac{\sqrt{2}}{2}$	0	0	$\frac{\sqrt{2}}{2}$
0	0.375	1	0	0.125	1	0	0	1
-0.375	0.375	$\frac{\sqrt{2}}{2}$	-0.125	0.125	$\frac{\sqrt{2}}{2}$	0	0	$\frac{\sqrt{2}}{2}$
-0.375	0	1	-0.125	0	1	0	0	1
-0.375	-0.375	$\frac{\sqrt{2}}{2}$	-0.125	-0.125	$\frac{\sqrt{2}}{2}$	0	0	$\frac{\sqrt{2}}{2}$
0	-0.375	1	0	-0.125	1	0	0	1

In the following sections, we will use IGA approximations of the form

$$u_h(x, y) = \sum_{i \in \mathbf{S}} N_i(x, y) u_i, \quad (2.10)$$

where  $N_i$  are the NURBS or B-splines basis functions,  $u_i$  are the field variables obtained by solving a linear matrix system, and  $\mathbf{S}$  contains the indices of all global basis functions.

### 2.3 Strong and Weak Form of Poisson's Equations

The Poisson's equation in  $d$  dimensions is given by:

$$\begin{cases} -\nabla \cdot (\mathbf{K}(\nabla u)^T) = f & \text{in } \Omega \\ u = \bar{u} & \text{on } \Gamma_e \\ \frac{\partial u}{\partial \mathbf{n}} = \bar{q} & \text{on } \Gamma_n, \end{cases} \quad (2.11)$$

where the  $d$ -dimensional domain  $\Omega$  is an open set with a Lipschitz continuous boundary  $\partial\Omega$ .  $\mathbf{K} \subset \mathbb{R}^d \times \mathbb{R}^d$  is a symmetric positive definite matrix. When  $\mathbf{K}$  is the identity matrix, (2.11) is identical to the classical Poisson's model problem. In other numerical examples,  $\mathbf{K}$  describes different material properties in different regions of the domain.  $f : \Omega \rightarrow \mathbb{R}$  is a given function.  $\bar{u}$  and  $\bar{q}$  are known variables which denote the boundary conditions.  $\mathbf{n}$  is the outward normal unit vector on the boundary  $\Gamma_n$ . The normal derivative  $\frac{\partial u}{\partial \mathbf{n}}$  can also be represented as  $\frac{\partial u}{\partial \mathbf{n}} = \nabla u \cdot \mathbf{n}$ .  $\Gamma_e$  and  $\Gamma_n$  are called the Dirichlet boundary and Neumann boundary, respectively.  $\Gamma_n \cap \Gamma_e = \emptyset$  and  $\Gamma_n \cup \Gamma_e = \partial\Omega$ .

Let us define two sets  $\mathcal{S}$  and  $\mathcal{V}$  :

$$\mathcal{S} = \{u \mid u \in H^1, u|_{\Gamma_e} = \bar{u}\}, \quad (2.12)$$

$$\mathcal{V} = \{v \mid v \in H^1, v|_{\Gamma_e} = 0\}, \quad (2.13)$$

where  $H^1$  is the Sobolev space of functions with square integrable first derivatives.

$$H^1 = H^1(\Omega) = \{w \mid w \in L^2; D^1 w \in L^2\}, \quad (2.14)$$

where

$$L^2 = L^2(\Omega) = \{w \mid \int_{\Omega} w^2 \, d\Omega < \infty\}. \quad (2.15)$$

The functions in  $\mathcal{S}$  satisfy the Dirichlet boundary condition of the problem (2.11).

### 2.3 Strong and Weak Form of Poisson's Equations

---

The homogeneous counterpart of the Dirichlet boundary condition is satisfied by the functions in  $\mathcal{V}$ .

In this thesis, the error is estimated by various norms. In the following, we use the usual notation for function norms and seminorms, namely for the  $L^\infty$ -norm, and  $W^{1,\infty}$ ,  $W^{2,\infty}$  seminorms,  $L^2$ -norm,  $H^1$  and  $H^2$  seminorms, which are defined as:

$$\begin{aligned} \|\cdot\|_{L^\infty} &= \sup |\cdot|, \\ |\cdot|_{W^{1,\infty}} &= \sup |D^1 \cdot|, \\ |\cdot|_{W^{2,\infty}} &= \sup |D^2 \cdot|, \\ \|\cdot\|_{L^2}^2 &= \int_{\Omega} |\cdot|^2, \\ |\cdot|_{H^1}^2 &= \int_{\Omega} |D^1 \cdot|^2, \\ |\cdot|_{H^2}^2 &= \int_{\Omega} |D^2 \cdot|^2, \end{aligned}$$

where  $D^1, D^2$  are the first and second derivative operators, and the sum of the squares of each component is taken for the square norms and seminorms in higher dimensions.

The weak form of (2.11) is obtained by multiplying a test function  $v$  ( $v \in \mathcal{V}$ ), and integrating over the domain  $\Omega$ ,

$$-\int_{\Omega} v \nabla \cdot (\mathbf{K}(\nabla u)^T) \, d\Omega = \int_{\Omega} v f \, d\Omega, \quad (2.16)$$

$$(2.17)$$

$$-\oint_{\Gamma} v \mathbf{K}(\nabla u)^T \cdot \mathbf{n} \, d\Gamma + \int_{\Omega} \nabla v \mathbf{K}(\nabla u)^T \, d\Omega = \int_{\Omega} v f \, d\Omega. \quad (2.18)$$

The weak form reads: find  $u \in \mathcal{S}$ , such that for all  $v \in \mathcal{V}$

$$\int_{\Omega} \nabla v \mathbf{K}(\nabla u)^T \, d\Omega = \int_{\Omega} v f \, d\Omega + \int_{\Gamma_n} v \mathbf{K}(\nabla u)^T \cdot \mathbf{n} \, d\Gamma_n. \quad (2.19)$$

Galerkin projection is to replace the above infinite-dimensional sets  $\mathcal{S}$  and  $\mathcal{V}$  by their finite-dimensional subsets  $\mathcal{S}^h$  and  $\mathcal{V}^h$ . Similar to the infinite dimensional sets

### 2.3 Strong and Weak Form of Poisson's Equations

---

$\mathcal{S}$  (2.12) and  $\mathcal{V}$  (2.13), the finite dimensional sets  $\mathcal{S}^h$  and  $\mathcal{V}^h$  are defined by

$$\mathcal{S}^h = \{u^h \mid u^h \in H^1, u^h|_{\Gamma_e} = \bar{u}\} \quad (2.20)$$

and

$$\mathcal{V}^h = \{v^h \mid v^h \in H^1, v^h|_{\Gamma_e} = 0\}. \quad (2.21)$$

After applying the Galerkin projection, the problem reads: find  $u^h \in \mathcal{S}^h$ , such that for all  $v^h \in \mathcal{V}^h$

$$\int_{\Omega} \nabla v^h \mathbf{K} (\nabla u^h)^T d\Omega = \int_{\Omega} v^h f d\Omega + \int_{\Gamma_n} v^h \mathbf{K} (\nabla u^h)^T \cdot \mathbf{n} d\Gamma_n . \quad (2.22)$$

In (2.22),  $\nabla$  are applied respect to the physical coordinates. For a two-dimensional problem ( $d = 2$ ), we also explicitly this by  $\nabla_{(x,y)}$ . The NURBS basis functions are defined in the parametric space. Using the chain rule, (2.23) can be written as:

$$\nabla_{(x,y)} u^h(x,y) = J(\xi, \eta)^{-1} \nabla_{(\xi,\eta)}^T u^h(\xi, \eta) , \quad (2.23)$$

where  $J$  is  $2 \times 2$  Jacobian matrix,

$$J(\xi, \eta) = \begin{pmatrix} \frac{\partial F_1}{\partial \xi} & \frac{\partial F_1}{\partial \eta} \\ \frac{\partial F_2}{\partial \xi} & \frac{\partial F_2}{\partial \eta} \end{pmatrix} , \quad (2.24)$$

and  $\mathbf{F}(\xi, \eta) = (F_1, F_2)$  is a transformation from the parametric coordinates  $(\xi, \eta)$  into the physical coordinates  $(x, y)$ :

$$\mathbf{F}(\xi, \eta) = \sum_{i \in \mathbf{S}} N_i(\xi, \eta) \mathbf{C}_i , \quad (2.25)$$

where  $\mathbf{C}_i$  are control points defined in the physical space. This transformation  $\mathbf{F}$  plays an important role in IGA. Because the same basis functions are used in the geometry



### 2.3 Strong and Weak Form of Poisson's Equations

---

description as in analysis, the method is called “isogeometric”.

Substituting (2.23) into (2.22), we obtain the following identities:

$$\int_{\Omega} \nabla v^h \mathbf{K} \nabla u^h \, dx dy = \int_{\Omega_0} (J^{-1}(\xi, \eta) \nabla v^h(\xi, \eta)) \mathbf{K} (J^{-1}(\xi, \eta) \nabla u^h(\xi, \eta)) | \det J(\xi, \eta) | \, d\xi d\eta , \quad (2.26)$$

$$\int_{\Omega} v^h f \, dx dy = \int_{\Omega_0} (v^h f)(J(\xi, \eta)) | \det J(\xi, \eta) | \, d\xi d\eta , \quad (2.27)$$

$$\int_{\Gamma_n} v^h \mathbf{K} \nabla u^h \cdot \mathbf{n} \, d\Gamma_n = \int_{\Gamma_{n_0}} (v^h \mathbf{K} \nabla u^h \cdot \mathbf{n})(J(\xi, \eta)) | \det J(\xi, \eta) | \, d\Gamma . \quad (2.28)$$

For  $n$ -dimensional space with  $n$  basis functions  $\mathbb{N} = \{ N_1, N_2, \dots, N_n \}$ , we choose  $v^h = N_i$ ,  $i = 1 \dots n$ , and  $u^h = \sum_{i=1}^n N_i c_i$ , as in the standard Galerkin method approximation. We are looking for  $n$  unknown coefficients  $c_i$ , with  $i = 1 \dots n$  for following linear problem:

$$\sum_{j=1}^n c_j a(N_j, N_i) = (f, N_i) + (N_i, \bar{q})_{\Gamma_n} , \quad (2.29)$$

where

$$a(N_j, N_i) = \int_{\Omega_0} \nabla N_j \mathbf{K} (\nabla N_i)^T \, d\Omega , \quad (2.30)$$

$$(f, N_i) = \int_{\Omega_0} f N_i \, d\Omega , \quad (2.31)$$

$$(N_i, \bar{q})_{\Gamma_n} = \int_{\Gamma_{n_0}} N_i \bar{q} \, d\Gamma_n . \quad (2.32)$$

We will now give a standard error estimate for the computed solution  $u^h$ . Suppose the exact solution  $u$  possesses  $r$  square integrable generalized derivatives<sup>94</sup>, i.e.  $u \in H^r(\Omega)$  where

$$H^r(\Omega) = \{ u \mid D^\alpha u \in L^2(\Omega), | \alpha | \leq r \} . \quad (2.33)$$

## 2.4 Linear Elasticity

---

Then, the norm associated with  $H^r(\Omega)$  is given by

$$\|u\|_r^2 = \sum_{|\alpha| \leq r} \int_{\Omega} (D^\alpha u) \cdot (D^\alpha u) \, d\Omega. \quad (2.34)$$

An a priori error estimator is given by the formula:

$$\|u - u^h\|_m \leq Ch^\beta \|u\|_r, \quad (2.35)$$

where  $\|\cdot\|_m$  and  $\|\cdot\|_r$  are the norms defined in Sobolev spaces  $H^m$  and  $H^r$ <sup>95</sup>.  $h$  is a characteristic length scale related to the size of the elements in the mesh.  $p$  is the polynomial degree of the basis.  $\beta = \min(p + 1 - m, r - m)$  is the convergence rate.  $C$  is a constant, which is independent of  $u$  and  $h$ . This fundamental FEM error estimate has been extended to IGA<sup>2</sup>.

## 2.4 Linear Elasticity

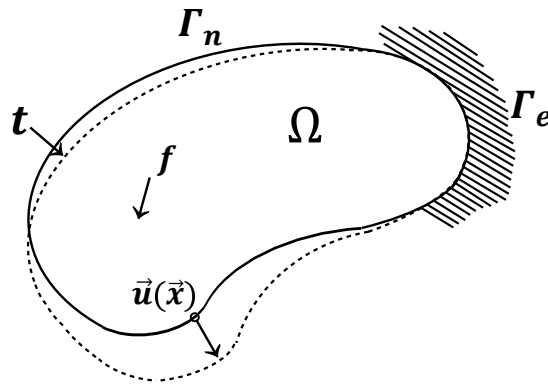


Figure 2.6:

Another example presented here is the linear elastic plane stress problem. Consider a homogeneous isotropic elastic body shown in Figure 2.6 occupying the boundary domain  $\Omega \subset \mathbb{R}^2$  with boundary  $\partial\Omega$ . The boundary  $\partial\Omega$  is decomposed into  $\Gamma_e$  and

## 2.4 Linear Elasticity

---

$\Gamma_n$  which are the Dirichlet boundary and Neumann boundary, respectively. Let the body be acted upon by a body force load  $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  and a boundary load  $\mathbf{t} : \vec{\Gamma}_n \rightarrow \mathbb{R}^2$  on  $\Gamma_n$ . Suppose that the displacement  $\vec{\mathbf{u}}$  is prescribed along the boundary  $\Gamma_e$ . We derive some basic equations of linear elasticity. For the plane stress problem, there are two components of the  $\vec{\mathbf{u}}$ , denoted by displacement,  $u$  and  $v$  with respect to global coordinates  $\vec{\mathbf{x}} = (x, y)$  respectively. This can be written as

$$\vec{\mathbf{u}}(x, y) = \begin{pmatrix} u(x, y) \\ v(x, y) \end{pmatrix}. \quad (2.36)$$

Strains are obtained from the partial derivatives of the displacements. The equations are called the kinematic equations, which are written as

$$\{\boldsymbol{\varepsilon}\} = \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{pmatrix}, \quad (2.37)$$

where  $\{\boldsymbol{\varepsilon}\} = (\varepsilon_x, \varepsilon_y, \gamma_{xy})^T$  stands for the strains. The constitutive equations state the relationship between the stresses and strains. For an isotropic material and the plane stress problem, the constitutive equations become

$$\{\boldsymbol{\sigma}\} = \begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = [\mathbf{K}]\{\boldsymbol{\varepsilon}\} = \begin{pmatrix} \frac{E}{1-\nu^2} & \frac{\nu E}{1-\nu^2} & 0 \\ \frac{\nu E}{1-\nu} & \frac{E}{1-\nu^2} & 0 \\ 0 & 0 & \frac{E}{2(1+\nu)} \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix}, \quad (2.38)$$

## 2.4 Linear Elasticity

---

where  $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \tau_{xy})^T$  denotes the stresses.  $[\mathbf{K}]$  is the material property matrix and  $E$  and  $\nu$  are the elastic modulus and Poisson's ratio, respectively. Finally, the equilibrium equations are defined as

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + f_x = 0 \quad (2.39)$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + f_y = 0, \quad (2.40)$$

where  $\vec{\mathbf{f}} = (f_x, f_y)$  is the body force. Combining Equations (2.37) (2.38) (2.39) and (2.40), we have eight unknowns (three stresses, three strains and two displacements) and eight equations (two equilibrium, three constitutive and three kinematic equations).

We shall assume that  $\partial\Omega$  admits decomposition

$$\Gamma_e \cup \Gamma_n = \partial\Omega, \quad \Gamma_e \cap \Gamma_n = \emptyset, \quad \Gamma_e \neq \emptyset. \quad (2.41)$$

Essential or displacement boundary conditions are applied on the boundary displacement part  $\Gamma_e$ , as

$$\mathbf{u} = \bar{\mathbf{u}}, \quad (2.42)$$

where  $\bar{\mathbf{u}}$  is given. Different from Equation (2.11),  $\mathbf{u}$  and  $\bar{\mathbf{u}}$  are vector-valued rather than scalar-valued. Natural or traction boundary conditions are applied on the traction boundary displacement part  $\Gamma_n$ , so that

$$\sigma_x n_x + \tau_{xy} n_y = \bar{t}_x \quad (2.43)$$

$$\tau_{xy} n_x + \sigma_y n_y = \bar{t}_y, \quad (2.44)$$

where  $\bar{\mathbf{n}} = (n_x, n_y)$  is the outward unit normal vector at  $\Gamma_n$ ,  $\bar{\mathbf{t}} = (t_x, t_y)$  is given. The equilibrium Equations (2.39) and (2.40) are also called the strong form of the boundary-value problem.

To derive the weak form, we define the trial solution set  $\mathcal{S} \times \mathcal{S}$  similarly to (2.12) and also the variational set  $\mathcal{V} \times \mathcal{V}$  similarly to (2.13). The weak form of (2.37) is

## 2.4 Linear Elasticity

---

obtained by multiplying the test functions  $w_x$  ( $w_x \in \mathcal{V}$ ),  $w_y$  ( $w_y \in \mathcal{V}$ ), and integrating over the region  $\Omega$ ,

$$\int_{\Omega} \begin{Bmatrix} w_x \left( \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} \right) \\ w_y \left( \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} \right) \end{Bmatrix} d\Omega + \int_{\Omega} \begin{Bmatrix} w_x f_x \\ w_y f_y \end{Bmatrix} d\Omega = 0. \quad (2.45)$$

Then using integration by parts and Gaussian divergence theorem on the first terms of Equation (2.45) yields

$$-\int_{\Omega} \begin{Bmatrix} \frac{\partial w_x}{\partial x} \sigma_x + \frac{\partial w_x}{\partial y} \tau_{xy} \\ \frac{\partial w_y}{\partial x} \tau_{xy} + \frac{\partial w_y}{\partial y} \sigma_y \end{Bmatrix} d\Omega + \int_{\Omega} \begin{Bmatrix} w_x f_x \\ w_y f_y \end{Bmatrix} d\Omega + \int_{\Gamma_t} \begin{Bmatrix} w_x \bar{t}_x \\ v_y \bar{t}_y \end{Bmatrix} d\Gamma_t = 0. \quad (2.46)$$

Substituting the constitutive Equation (2.38) and the kinematic Equation (2.37) results in

$$\int_{\Omega} \begin{bmatrix} \frac{\partial w_x}{\partial x} & 0 & \frac{\partial w_x}{\partial y} \\ 0 & \frac{\partial w_y}{\partial y} & \frac{\partial w_y}{\partial x} \end{bmatrix} [\mathbf{K}] \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} d\Omega = \int_{\Omega} \begin{Bmatrix} w_x f_x \\ w_y f_y \end{Bmatrix} d\Omega + \int_{\Gamma_t} \begin{Bmatrix} w_x \bar{t}_x \\ w_y \bar{t}_y \end{Bmatrix} d\Gamma_t. \quad (2.47)$$

After applying Galerkin projection, the problem reads: find  $\mathbf{u}^h = (u_x^h, u_y^h) \in \mathcal{S}^h \times \mathcal{S}^h$ , for all  $v_x^h \in \mathcal{V}^h$ ,  $v_y^h \in \mathcal{V}^h$  such that

$$\int_{\Omega} \begin{bmatrix} \frac{\partial w_x^h}{\partial x} & 0 & \frac{\partial w_x^h}{\partial y} \\ 0 & \frac{\partial w_y^h}{\partial y} & \frac{\partial w_y^h}{\partial x} \end{bmatrix} [\mathbf{K}] \begin{bmatrix} \frac{\partial u^h}{\partial x} \\ \frac{\partial v^h}{\partial y} \\ \frac{\partial u^h}{\partial y} + \frac{\partial v^h}{\partial x} \end{bmatrix} d\Omega = \int_{\Omega} \begin{Bmatrix} w_x^h f_x \\ w_y^h f_y \end{Bmatrix} d\Omega + \int_{\Gamma_t} \begin{Bmatrix} w_x^h \bar{t}_x \\ w_y^h \bar{t}_y \end{Bmatrix} d\Gamma_t. \quad (2.48)$$

$\mathcal{S}^h$  and  $\mathcal{V}^h$  are finite dimensional subsets  $\mathcal{S}$  and  $\mathcal{V}$ . For  $n$ -dimensional space with

## 2.4 Linear Elasticity

---

$n$  basis functions,  $\mathbb{N} = \{N_1, N_2, \dots, N_n\}$ ,  $u$  and  $v$  are interpolated using the same basis functions, such that

$$u^h = \sum_{i=1}^n N_i u_i, \quad v^h = \sum_{i=1}^n N_i v_i. \quad (2.49)$$

The displacements can also be expressed as

$$\begin{Bmatrix} u^h \\ v^h \end{Bmatrix} = \begin{pmatrix} N_1 & 0 & N_2 & 0 & \dots & N_n & 0 \\ 0 & N_1 & 0 & N_2 & \dots & 0 & N_n \end{pmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{Bmatrix} = [\mathbf{N}]\{\mathbf{d}\}. \quad (2.50)$$

Use of this expression for strains yields

$$\begin{Bmatrix} \frac{\partial u^h}{\partial x} \\ \frac{\partial v^h}{\partial y} \\ \frac{\partial u^h}{\partial y} + \frac{\partial v^h}{\partial x} \end{Bmatrix} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \dots & 0 & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & \dots & \frac{\partial N_{n-1}}{\partial y} & 0 & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \dots & \frac{\partial N_{n-1}}{\partial x} & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} \end{pmatrix} \{\mathbf{d}\} = [\mathbf{B}]\{\mathbf{d}\}. \quad (2.51)$$

To compute the Galerkin projection, we let successively  $w_x^h = N_i$ ,  $i = 1, \dots, n$ ,  $w_y^h = N_i$ ,  $i = 1, \dots, n$ . Then plugging Equations (2.51) into the first terms of Equation (2.48). For the two terms on the righthand side of Equation (2.48), following a similar procedure, we obtain

$$[\mathbf{D}]\{\mathbf{d}\} = \{F\} + \{\Phi\} \quad (2.52)$$

## 2.5 Knot insertion and order elevation

---

where

$$[\mathbf{D}]\{\mathbf{d}\} = \int_{\Omega_0} [\mathbf{B}]^T \mathbf{K} [\mathbf{B}] d\Omega \{\mathbf{d}\}. \quad (2.53)$$

$$\{\mathbf{F}\} = \int_{\Omega_0} [\mathbf{N}]^T \{\mathbf{f}\} d\Omega \quad (2.54)$$

and

$$\{\Phi\} = \int_{\Gamma_{t_0}} [\mathbf{N}]^T \{\mathbf{t}\} d\Gamma_t. \quad (2.55)$$

Here, Equation (2.53), (2.54) and (2.55) are the element stiffness matrix, body force vector and boundary traction vector, respectively.

## 2.5 Knot insertion and order elevation

### 2.5.1 Knot insertion

Given a particular NURBS discretization of degree  $p$  in terms of knot vectors, control points, weights, we will discuss some refinement strategies. Suppose the initial knot vector  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$  and the corresponding control points  $\{\mathbf{P}_{A=1}^n\}$  are given. We insert a new knot vector  $\bar{\xi}_k$  into  $\Xi$ . The new knot vector is written as  $\bar{\Xi} = \{\xi_1, \xi_2, \dots, \bar{\xi}_k, \dots, \xi_{m+p+1}\}$  with  $m + p + 1$  knots and  $m = n + 1$ . It requires  $m$  new basis functions defined using Equations (2.1) and (2.2) based on  $\bar{\Xi}$ . The  $m$  new control points represented by  $\{\bar{\mathbf{P}}_A\}_{A=1}^m$  are calculated by

$$\bar{\mathbf{P}}_A = \begin{cases} \mathbf{P}_1 & A = 1 \\ \alpha_A \mathbf{P}_A + (1 - \alpha_A) \mathbf{P}_{A-1} & 1 < A < m \\ \mathbf{P}_n & m = A \end{cases}$$

## 2.5 Knot insertion and order elevation

---

where

$$\alpha_A = \begin{cases} 1 & 1 \leq A \leq k-p \\ \frac{\bar{\xi} - \xi_A}{\xi_{A+p} - \xi_A} & k-p+1 \leq A \leq k \\ 0 & A \geq k+1 \end{cases} .$$

The knot values may be inserted multiple times. But the continuity of the basis is reduced by one for each repetition of insertion knot value. However, the B-spline curve generated by the new knot vector and the resulting control points is geometrically identical to the original one. An example of knot insertion is shown in Figure 2.7 which of the initial knot vector and the control points are given by Figure 2.3. The new curves in Figures 2.3 (a) and (c) are identical to the unrefined curve in Figure 2.3 (a), however more control points and basis functions are introduced.

### 2.5.2 Order elevation

Here, given a geometry parametrization of degree  $p$ , we want to increase the polynomial degree of the discretization to  $p+1$ . Comparing to the knot insertion, order elevation is more complicated and needs three main steps. The first step is Bézier decomposition, which represents a piecewise polynomial curve by Bézier segments. The process follows the knot insertion in section 2.5.1, repeating the interior knots  $p$  times.

Then second step is order elevation for each Bézier segment. We present a well-known formula<sup>92,96</sup> to elevate from degree  $p$  to degree  $p+1$ . A Bézier segment curve is written as

$$\mathbf{C}_p(\xi) = \sum_{i=0}^p B_{i,p}(\xi) \mathbf{P}_i \quad 0 \leq \xi \leq 1, \quad (2.56)$$

where  $\{\mathbf{P}_i\}$  are the control points generated after the Bézier decomposition step.  $B_{i,p}(\xi)$  are Bernstein basis functions<sup>92,97</sup> given by

$$B_{i,p}(\xi) = \frac{p!}{i!(p-i)!} \xi^i (1-\xi)^{p-i}. \quad (2.57)$$



## 2.5 Knot insertion and order elevation

---

The continuity of the  $(p+1)^{\text{th}}$  degree Bézier curve has to be the same as that of the  $p^{\text{th}}$  degree Bézier curve. This reduced continuity property is obtained by increasing the multiplicities of all knots by one. Then the  $(p+1)^{\text{th}}$  degree curve is represented as

$$\mathbf{C}_{p+1}(\xi) = \sum_{i=0}^{p+1} B_{i,p+1}(\xi) \mathbf{Q}_i, \quad (2.58)$$

where  $\{\mathbf{Q}_i\}_{i=0}^{p+1}$  are the new control points which are unknown. Since the  $p^{\text{th}}$  degree curve is identical to the  $(p+1)^{\text{th}}$  degree curve, we have

$$\sum_{i=0}^{p+1} B_{i,p+1} \mathbf{Q}_i = \sum_{i=0}^p B_{i,p} \mathbf{P}_i. \quad (2.59)$$

The final formula for calculating  $\mathbf{Q}_i$  is written as

$$\mathbf{Q}_i = (1 - \alpha_i) \mathbf{P}_i + \alpha_i \mathbf{P}_{i-1} \quad (2.60)$$

where

$$\alpha_i = \frac{i}{p+1} \quad i = 0, \dots, p+1. \quad (2.61)$$

Note the complete derivation from Equation (2.59) to Equation (2.61) can be found in literature<sup>92</sup>.

Finally the third step is to remove the repeated knots. A general formula for unnecessary knot removal are given by

$$\left\{ \begin{array}{l} r - p \leq A \leq \frac{1}{2}(2r - p - s - 1) \\ \bar{\mathbf{P}}_A = \frac{\mathbf{P}_A - (1 - \alpha_A) \bar{\mathbf{P}}_{A-1}}{\alpha_A} \\ \alpha_A = \frac{\bar{\xi} - \xi_A}{\xi_{A+p-1} - \xi_A} \end{array} \right. , \quad (2.62)$$

## 2.6 Conclusions

---

$$\left\{ \begin{array}{l} \frac{1}{2}(2r - p - s + 2) \leq B \leq r - s \\ \bar{\mathbf{P}}_B = \frac{\mathbf{P}_B - \alpha_B \bar{\mathbf{P}}_{B+1}}{1 - \alpha} \\ \alpha_B = \frac{\bar{\xi} - \xi_B}{\xi_{B+p+1} - \xi_B} \end{array} \right. , \quad (2.63)$$

where  $\{\mathbf{P}\}$  is initially given.  $\bar{\xi} = \xi_r \neq \xi_{r+1}$  stands for the removing knot.  $A = r - p$ ,  $B = r - s$  and  $B > A$  are required. For detail implementation of the knot removal, we refer to<sup>92</sup>. An example of order elevation is shown in Figure 2.8 which of the initial knot vector and the control points are given by Figure 2.3. The new curves in Figure 2.8 (a), (c) and (e) are identical to the unrefined curve in Figure 2.3 (a). Some excellent and useful open source IGA tools written in MATLAB are given in<sup>98</sup>.

## 2.6 Conclusions

In this section, we gave an overview of IGA and introduced the main ideas through the Poisson's equation and linear elasticity examples. We discuss the B-spline and NURBS basis functions and their common properties. At the end we summarize two refinement techniques, the knot insertion and the order elevation, respectively. The original contributions of this dissertation are divided into four topics and presented in following chapters. Some particular notations, formulas and properties are only given in the chapter where they are applied.

## 2.6 Conclusions

---

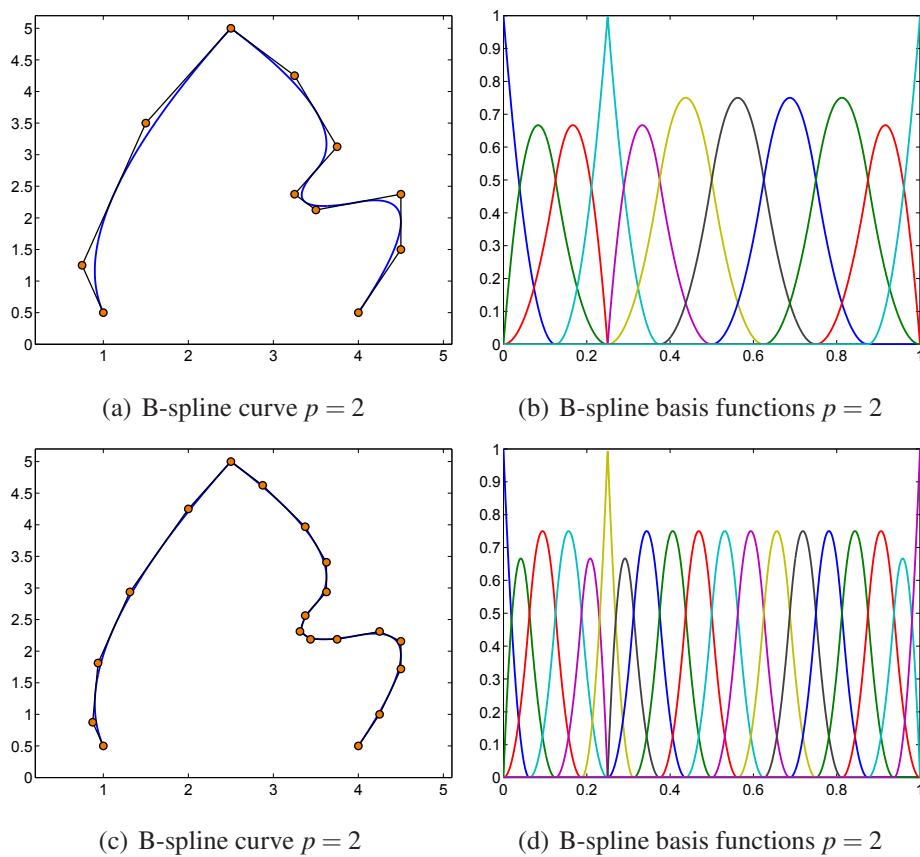


Figure 2.7: Knot insertion. Based on the initial knot vector and the control points given by Figure 2.3, the newly inserted knots are  $\{1/8, 3/8, 5/8, 7/8\}$  and the corresponding B-spline curve and the basis functions are shown in (a) and (b). Keeping the current knot vector, we insert a new sequence of knots  $\{1/16, 3/16, 5/16, 7/16, 9/16, 11/16, 13/16, 15/16\}$  and obtain the B-spline curve (c) and the basis functions plotted in (d).

## 2.6 Conclusions

---

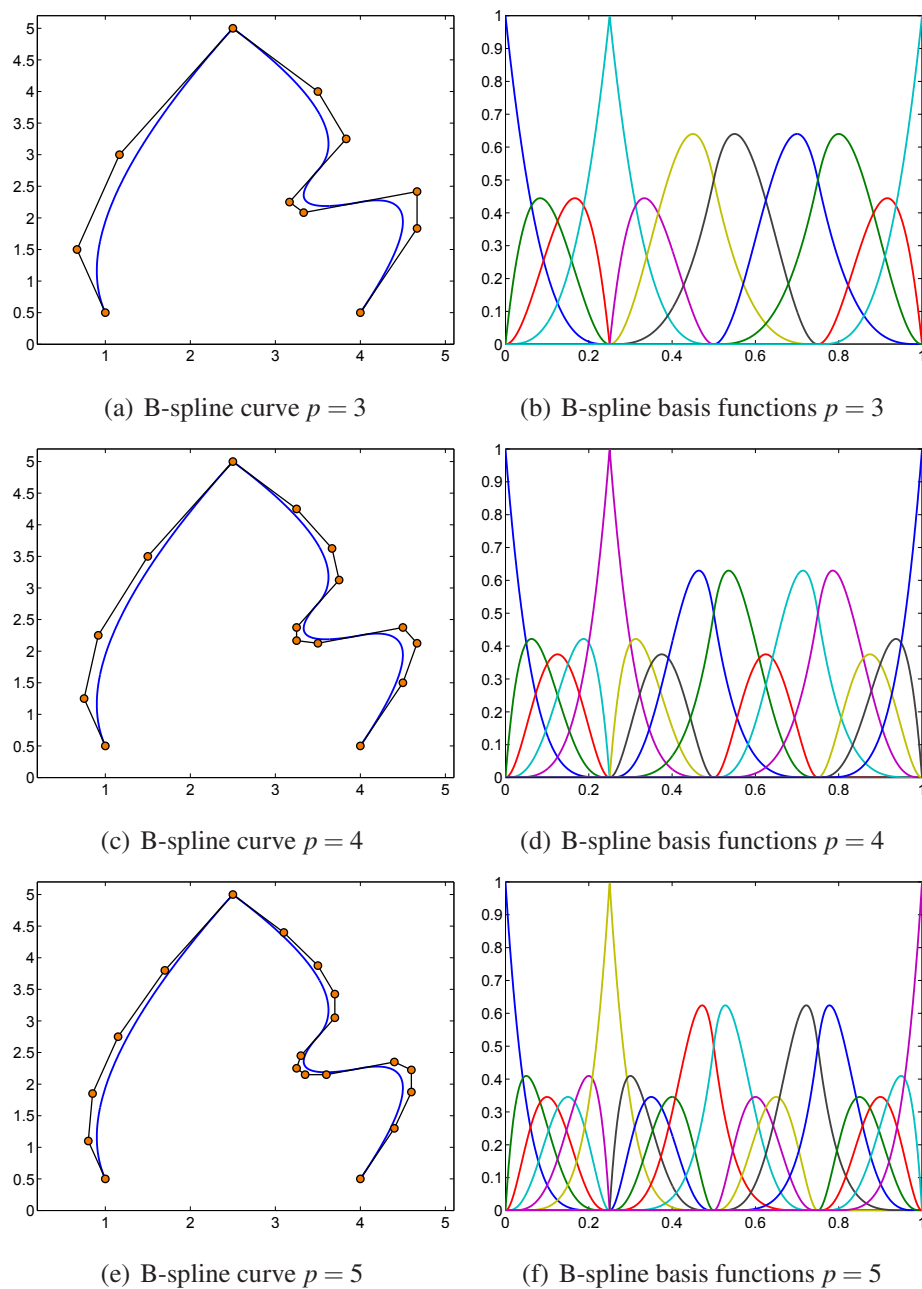


Figure 2.8: Based on the initial knot vector and the control points given by Figure 2.3, we apply the order elevation technique and obtain the identical B-spline curves of  $p = 3$  (a),  $p = 4$  (c) and  $p = 5$  (e). Subfigures (b), (d) and (f) are their corresponding basis functions.

## Chapter 3

# Reproducing Kernel Triangular B-spline-based FEM for Solving PDEs

### 3.1 Introduction

We propose a reproducing kernel triangular B-spline-based finite element method (FEM) as an improvement to the conventional triangular B-spline element for solving partial differential equations (PDEs). In the latter, unexpected errors can occur throughout the analysis domain mainly due to the excessive flexibility in defining the triangular B-splines. The performance therefore becomes unstable and cannot be controlled in a desirable way. To address this issue, the proposed improvement adopts the reproducing kernel approximation in the calculation of B-spline kernel function. Three types of PDE problems are tested to validate the present element and compare against the conventional triangular B-splines. It has been shown that the improved triangular B-splines satisfy the partition of unity condition even for extreme conditions including corners and holes.

### 3.2 Reproducing Kernel Triangular B-spline

#### 3.2.1 Simplex Splines

We start from the simplex splines discussion, since a lot of concepts and properties of the simplex splines can be extended to the triangular B-splines directly. Let  $\mathbf{V} = \{\mathbf{t}_0, \mathbf{t}_1, \dots, \mathbf{t}_{n+2}\}$  be a finite set of points in  $\mathbb{R}^2$ . These points in  $\mathbf{V}$  are called knots and the set  $\mathbf{V}$  itself is called the knot-set of a simplex spline. A simplex spline defined over  $\mathbf{V}$  is a piecewise polynomial of degrees  $n$ . The definition of a simplex spline is given by a recursive equation

$$M(\mathbf{x} | \mathbf{V}) = \begin{cases} 0 & \mathbf{x} \notin [\mathbf{V}], \\ \frac{1}{|\det(\mathbf{V})|} & |\mathbf{V}| = 3 \quad \mathbf{x} \in [\mathbf{V}], \\ \sum_{i=0}^2 \lambda_i(\mathbf{x} | \mathbf{W}) M(\mathbf{x} | \mathbf{V} \setminus \{\mathbf{w}_i\}) & |\mathbf{V}| > 3, \end{cases} \quad (3.1)$$

where  $\det(\mathbf{V})$  is the determinant of points. When  $\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)$  is a triple of points in  $\mathbb{R}^2$ , the determinant of  $\mathbf{V}$  is defined as

$$\det(\mathbf{V}) = \det \begin{pmatrix} 1 & 1 & 1 \\ v_{0x} & v_{1x} & v_{2x} \\ v_{0y} & v_{1y} & v_{2y} \end{pmatrix}, \quad (3.2)$$

where  $\mathbf{v}_0 = (v_{0x}, v_{0y})$ ,  $\mathbf{v}_1 = (v_{1x}, v_{1y})$ , and  $\mathbf{v}_2 = (v_{2x}, v_{2y})$ .

$\lambda_i(\mathbf{x} | \mathbf{W})$  are the barycentric coordinates of  $\mathbf{x}$  with respect to the three knots in a subset  $\mathbf{W} = \{\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2\}$ . The barycentric coordinates for the planar case have the following properties:

- The sum of barycentric coordinates equals to 1;

### 3.2 Reproducing Kernel Triangular B-spline

---

- The barycentric coordinates relation expression is an affine mapping, which means the barycentric coordinates do not change after the linear transformation;
- If  $\mathbf{x}$  lies on one edge of the triangle, at least one barycentric coordinate equals to zero;
- If  $\mathbf{x}$  lies on one vertex of the triangle, two of barycentric coordinates equal to zero; and
- If  $\mathbf{x}$  lies outside of the triangle, then at least one barycentric coordinate is negative.

If the three points  $\mathbf{w}_0$ ,  $\mathbf{w}_1$  and  $\mathbf{w}_2$  in  $\mathbf{W}$  and  $\mathbf{x}$  point are known, we can determine the barycentric coordinates by the Cramer's rule

$$\lambda_0 = \frac{\text{area}(\mathbf{x}, \mathbf{w}_1, \mathbf{w}_2)}{\text{area}(\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2)}, \lambda_1 = \frac{\text{area}(\mathbf{w}_0, \mathbf{x}, \mathbf{w}_2)}{\text{area}(\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2)}, \lambda_2 = \frac{\text{area}(\mathbf{w}_0, \mathbf{w}_1, \mathbf{x})}{\text{area}(\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2)}, \quad (3.3)$$

where  $\text{area}(\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2) = \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ w_{0x} & w_{1x} & w_{2x} \\ w_{0y} & w_{1y} & w_{2y} \end{vmatrix}$ ,  $\mathbf{w}_0 = (w_{0x}, w_{0y})$ ,  $\mathbf{w}_1 = (w_{1x}, w_{1y})$ , and

$\mathbf{w}_2 = (w_{2x}, w_{2y})$ .  $[\mathbf{V}]$  stands for the half open convex hull. Its definition is defined as

$$[\mathbf{V}] = \{\mathbf{x} \in [\mathbf{V}] \mid \exists \varepsilon_0, \varepsilon_1 > 0 (\forall 0 \leq \alpha_1 \leq \alpha_0 \leq 1 (\mathbf{x} + \alpha_0 \varepsilon_0 \mathbf{e}_0 + \alpha_1 \varepsilon_1 \mathbf{e}_1 \in [\mathbf{V}]))\},$$

where  $[\mathbf{V}]$  denotes the convex hull of  $\mathbf{V}$ .  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are the unit vectors for  $x$ -direction and  $y$ -direction, respectively. Figures 3.1 and 3.2 sketch the half open convex hull and a few examples, respectively. Constant, linear and quadratic simplex B-splines examples are shown in Figure 3.3.

### 3.2 Reproducing Kernel Triangular B-spline

---

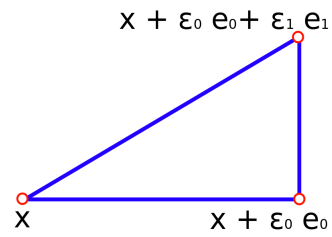


Figure 3.1: Sketch to illustrate the half-open convex hull.

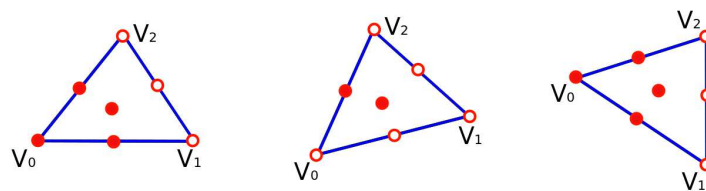
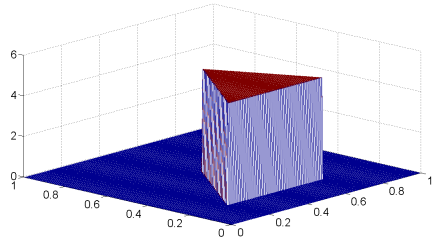


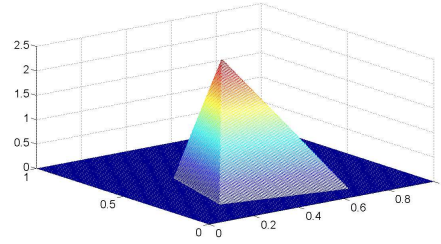
Figure 3.2: Examples to illustrate the half-open convex hull  $[V_0, V_1, V_2)$ . The solid points belong to the half-open convex hull, the empty ones do not.



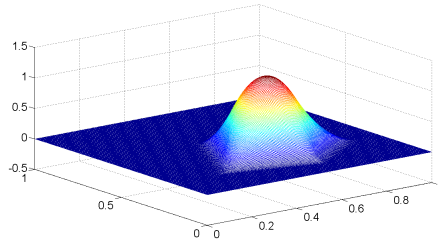
## 3.2 Reproducing Kernel Triangular B-spline



(a) simplex spline of degree zero



(b) linear simplex spline



(c) quadratic simplex spline

Figure 3.3: Simplex splines examples. (a) is a simplex spline of degree zero whose knots are  $\mathbf{t}_0 = (0.2, 0.2)$ ,  $\mathbf{t}_1 = (0.7, 0.2)$  and  $\mathbf{t}_2 = (0.5, 0.6)$ ; (b) is a linear simplex spline whose knots are  $\mathbf{t}_0 = (0.2, 0.2)$ ,  $\mathbf{t}_1 = (0.6, 0.7)$ ,  $\mathbf{t}_2 = (0.3, 0.6)$  and  $\mathbf{t}_3 = (0.7, 0.1)$ ; and (c) is a quadratic simplex spline whose knots are  $\mathbf{t}_0 = (0.2, 0.2)$ ,  $\mathbf{t}_1 = (0.6, 0.1)$ ,  $\mathbf{t}_2 = (0.9, 0.3)$ ,  $\mathbf{t}_3 = (0.7, 0.7)$  and  $\mathbf{t}_4 = (0.4, 0.6)$ .

### 3.2.2 Triangular B-splines

Triangular B-splines, also well-known as DMS-splines, were first developed by Dahmen, Micchelli and Seidal<sup>52</sup>. It was derived from simplex splines and B-patches. A lot of concepts and theorems of B-patches can be extended to triangular B-splines directly. More information of the B-patches discussion can be found in<sup>93</sup>.

Compared with the knots set  $\mathbf{V}$  defined earlier, the knot set for defining triangular B-splines is larger and more complicated. Following the definition given in<sup>99</sup>, we use knot net instead of knot set in the triangular B-splines study area. Let

### 3.2 Reproducing Kernel Triangular B-spline

$\mathbf{K} = \{\mathbf{t}_{0,0}, \dots, \mathbf{t}_{0,n}, \dots, \mathbf{t}_{1,n}, \dots, \mathbf{t}_{2,n}\}$  be a knot net associated with the triangle  $\triangle = [\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2] = [\mathbf{t}_{0,0}, \mathbf{t}_{1,0}, \mathbf{t}_{2,0}]$ . For each root knot  $\mathbf{t}_i = \mathbf{t}_{i,0}$  ( $i = 0, 1, 2$ ), there are  $n$  extra knots connecting to the root knot. The set of points  $\mathbf{C}_i = \{\mathbf{t}_{i,0}, \dots, \mathbf{t}_{i,n}\}$  is called the knot cloud associated with the root knot  $\mathbf{t}_i = \mathbf{t}_{i,0}$ .  $n$  is also the polynomial order of triangular B-splines. For one knot net  $\mathbf{K}$ , we can define  $\frac{(n+1)(n+2)}{2}$  triangular B-spline basis functions. For each triangular B-spline basis, we need to define a certain knot set  $\mathbf{V}_{ijk}^{\mathbf{K}} \subseteq \mathbf{K}$ .  $\mathbf{V}_{ijk}^{\mathbf{K}} = \{\mathbf{t}_{0,0}, \dots, \mathbf{t}_{0,i}, \mathbf{t}_{1,0}, \dots, \mathbf{t}_{1,j}, \mathbf{t}_{2,0}, \dots, \mathbf{t}_{2,k}\}$  is a subset of the knot net  $\mathbf{K}$ , and it requires  $i + j + k = n$ . There are  $n + 3$  knots in  $\mathbf{V}_{ijk}^{\mathbf{K}}$ . The normalized triangular B-spline basis are defined by

$$N_{ijk}^{\mathbf{K}}(\mathbf{x}) = |\det(\mathbf{t}_{0,i}, \mathbf{t}_{1,j}, \mathbf{t}_{2,k})| M(\mathbf{x} | \mathbf{V}_{ijk}^{\mathbf{K}}), \quad (3.4)$$

where  $M(\mathbf{x} | \mathbf{V}_{ijk}^{\mathbf{K}})$  is the simplex spline among the knot set  $\mathbf{V}_{ijk}^{\mathbf{K}}$  at point  $\mathbf{x}$ .  $|\det(\mathbf{t}_{0,i}, \mathbf{t}_{1,j}, \mathbf{t}_{2,k})|$  is the normalizing factor. For the sake of simplicity and uniformity, the notation  $N_{ijk}^{\mathbf{K}}(\mathbf{x})$  will henceforth be used for the triangular B-spline basis function evaluated at point  $\mathbf{x}$ . Figures 3.4 and 3.5 show the distribution of the quadratic triangular B-spline and the examples of the quadratic triangular B-spline basis, respectively.

The triangular B-spline basis  $N_{ijk}^{\mathbf{K}}(\mathbf{x})$  are linearly independent, and they also satisfy the partition of unity property, i.e.,  $\sum_{i+j+k=n} N_{ijk}^{\mathbf{K}}(\mathbf{x}) = 1$ . The proof of the two properties can be referred to<sup>99</sup>.

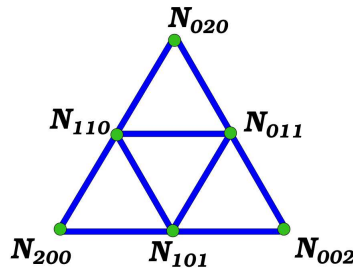


Figure 3.4: Distribution of triangular B-spline basis for quadratic case.

### 3.2 Reproducing Kernel Triangular B-spline

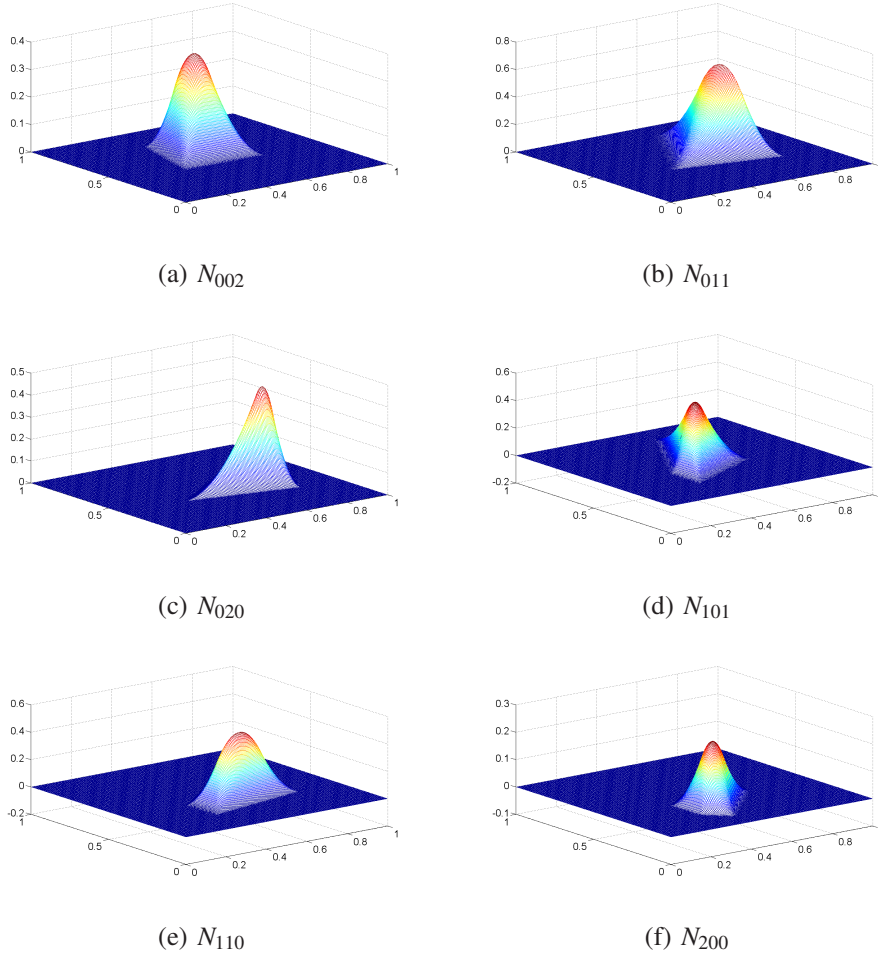


Figure 3.5: Normalized quadratic triangular B-spline basis functions.

The application of triangular B-splines here is for solving PDEs. The main idea follows FEM by solving the weak form which is equivalent to the strong form. We need to find the derivative of the basis function. There could be several ways to define it. Here we follow one efficient way proposed in <sup>100</sup>. The derivative of one triangular B-spline is given by

$$\begin{aligned}
 \nabla_{\mathbf{w}} N_{ijk}^{\mathbf{K}}(\mathbf{x} \mid \mathbf{V}_{ijk}^{\mathbf{K}}) &= |\det(\mathbf{t}_{0,i}, \mathbf{t}_{1,j}, \mathbf{t}_{2,k})| \nabla_{\mathbf{w}} M(\mathbf{x} \mid \mathbf{V}_{ijk}^{\mathbf{K}}) \\
 &= |\det(\mathbf{t}_{0,i}, \mathbf{t}_{1,j}, \mathbf{t}_{2,k})| n \sum_{i=0}^2 \mu_i M(\mathbf{x} \mid \mathbf{V}_{ijk}^{\mathbf{K}} \setminus \{\mathbf{w}_i\}),
 \end{aligned} \tag{3.5}$$

### 3.2 Reproducing Kernel Triangular B-spline

---

where  $n$  is the order of the triangular B-spline. The coefficients  $\mu_i$  require  $\sum_{i=0}^2 \mu_i = 0$  and  $\sum_{i=0}^2 \mu_i \mathbf{t}_i = \mathbf{w}$ . From Equation 3.5, taking the derivative with respect to the triangular B-splines is transferred by taking the derivative with respect to the coefficients of the simplex splines  $\lambda_i$ , i.e.,  $\mu_i = \nabla_{\mathbf{w}} \lambda_i$ . Here the operator  $\nabla_{\mathbf{w}}$  stands for the directional derivative.

The directional derivative is a special case of the Gâteaux derivative.  $\lambda_i(\mathbf{x} | \mathbf{W})$  is a scalar function. According to the definition of the directional derivative,  $\nabla_{\mathbf{w}} \lambda_i$  is calculated by

$$\nabla_{\mathbf{w}} \lambda_i(\mathbf{x} | \mathbf{W}) = \lim_{h \rightarrow 0} \frac{\lambda_i(\mathbf{x} + h\mathbf{w} | \mathbf{W}) - \lambda_i(\mathbf{x} | \mathbf{W})}{h}, \quad (3.6)$$

and two important relation expressions for calculating the derivative of the triangular B-spline basis function in our case are given by

$$\nabla_{\mathbf{w}} \lambda_i(\mathbf{x} | \mathbf{W}) |_{\mathbf{w}=(1,0)} \iff \nabla_x \lambda_i(\mathbf{x} | \mathbf{W}), \quad (3.7)$$

and

$$\nabla_{\mathbf{w}} \lambda_i(\mathbf{x} | \mathbf{W}) |_{\mathbf{w}=(0,1)} \iff \nabla_y \lambda_i(\mathbf{x} | \mathbf{W}). \quad (3.8)$$

Then, we can obtain

$$\nabla \lambda_0(\mathbf{x} | \mathbf{W}) = \left( \frac{w_{1y} - w_{2y}}{\det(\mathbf{W})}, \frac{w_{2x} - w_{1x}}{\det(\mathbf{W})} \right), \quad (3.9)$$

$$\nabla \lambda_1(\mathbf{x} | \mathbf{W}) = \left( \frac{w_{2y} - w_{0y}}{\det(\mathbf{W})}, \frac{w_{0x} - w_{2x}}{\det(\mathbf{W})} \right), \quad (3.10)$$

and

$$\nabla \lambda_2(\mathbf{x} | \mathbf{W}) = \left( \frac{w_{0y} - w_{1y}}{\det(\mathbf{W})}, \frac{w_{1x} - w_{0x}}{\det(\mathbf{W})} \right). \quad (3.11)$$

**Remark:** We need to derive the derivatives of triangular B-splines because they are required to calculate the variational form in FEM. Although there are many papers studying triangular B-splines<sup>101,102,103</sup>, only a few<sup>104,105</sup> mentioned their derivatives with some constrains. This gives us some difficulty to calculate the coefficients  $\mu_i$ .<sup>100</sup>

## 3.2 Reproducing Kernel Triangular B-spline

---

is the first paper providing us the idea to get  $\mu_i$  by calculating the directional derivative of  $\lambda_i$ . Herein, we derive our own mathematical formulation as above for the derivatives of triangular B-splines in 2D case. Its proof is a little bit long but it is not difficult to obtain, so we do not provide here.

For a problem domain, we need a large knot net including many extra knot cloud points. One example of knot net distribution for a problem domain is shown in Figure 3.6.

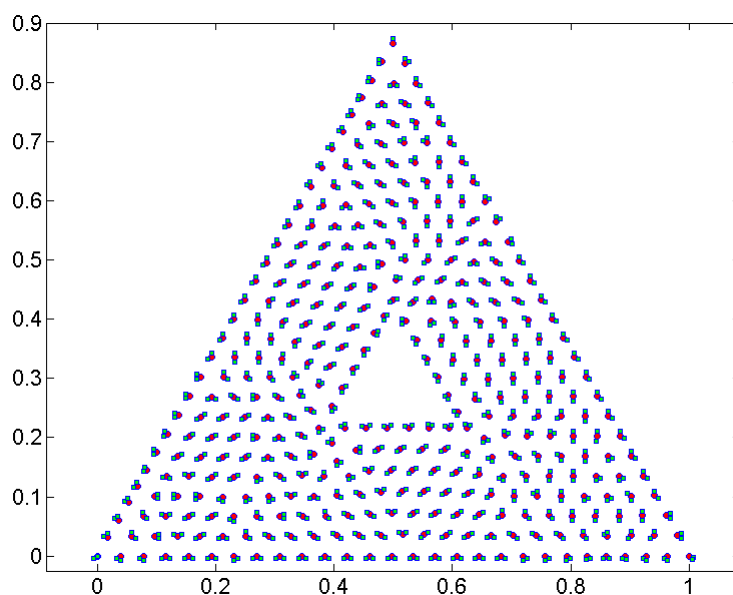


Figure 3.6: Distribution of knot net for a problem domain. The red dots are the root knot-nodes. The blue squares are the additional knot-nodes associated to each root knot-node.

### 3.2.3 Reproducing Kernel Triangular B-spline

The triangular B-spline has been named for around two decades, and it has also been studied by many people. The triangular B-spline is well-known for its triangular structure. However, till now there is no explicit rule or criteria showing how to construct

### 3.2 Reproducing Kernel Triangular B-spline

---

analysis-suitable triangular B-splines. The challenge is due to the flexibilities of the function itself. In analysis, the performance of the triangular B-spline is influenced by the quality of triangulation and also depends on the distribution of knot clouds. Nowadays both of them still need a lot of study and development<sup>106</sup>. In our study, due to such unstable nature of the triangular B-spline, there could be many unexpected error occurring if we apply it as the basis function directly. So we follow the reproducing kernel approximation technique to improve the triangular B-spline basis. This idea was originally derived from RKPM<sup>86,87,88,89</sup>, which is one of the most popular meshfree methods. It improves the first generation of the SPH meshfree method by multiplying the original basis function by some correction term.

The classical FEM approximation is written as

$$u_h(x, y) = \sum_{i \in \mathbf{S}} N_i(x, y) u_i, \quad (3.12)$$

where  $N_i$  are the triangular B-spline basis functions,  $u_i$  are the field variables obtained by solving a linear matrix system, and  $\mathbf{S}$  contains the indices of all global basis functions. The improved approximation is defined in a similar way,

$$u_h(x, y) = \sum_{i \in \mathbf{S}} \Phi_i(x, y) a_i, \quad (3.13)$$

where  $\Phi_i$  are the improved triangular B-splines containing correction parts.  $a_i$  are the field variables obtained by solving a linear matrix system corresponding to the new basis functions, and  $\mathbf{S}$  contains the indices of all global basis functions with the same number as in Equation 3.12.

In the above Equations 3.12 and 3.13,  $N_i$  are called kernel/weight/smoothing basis functions in SPH, and  $\Phi_i$  are called the corrected kernel basis functions in RKPM. They satisfy the following three properties:

- Compact support, i.e.,  $\Phi_i(\mathbf{x}) = 0$  if  $\mathbf{x} \notin \Omega_e$  where  $\Omega_e$  is the support of the basis  $\Phi_i$ ;

### 3.2 Reproducing Kernel Triangular B-spline

---

- Partition of unity, i.e.,  $\sum_{i \in \mathbf{S}} \Phi_i(\mathbf{x}) = 1$ ; and
- Polynomial reproducing constraint, i.e.,  $\sum_{i \in \mathbf{S}} \Phi_i \mathbf{x}_i = \mathbf{x}$ .

Following the pioneer work in<sup>86,87,88,89</sup>, we should emphasize the procedure of defining those basis functions<sup>107,108</sup>. For very detailed mathematical theorems and completed proofs please refer to<sup>109,110</sup>. The reproducing kernel particle theorems also promoted the development of RKEM<sup>111,112,113,114</sup> and the moving particle finite element method<sup>115</sup>. Some related partition of unity (PU) properties are described in<sup>116</sup>. Here, we define the reproducing kernel triangular B-spline by

$$\Phi_i(x, y) = \mathbf{H}^T(x - x_i, y - y_i) \mathbf{b}(x, y) N_i(x, y), \quad (3.14)$$

where  $\mathbf{H}^T$  is a set of polynomial basis  $\{x^\alpha y^\beta\}_{|\alpha+\beta| \leq n}$  ( $n$  is the order of the triangular B-spline basis),  $\mathbf{b}(x, y)$  are the coefficients of the polynomial basis in  $\mathbf{H}^T$ ,  $N_i(x, y)$  is the traditional triangular B-spline, and  $(x_i, y_i)$  are the coordinates of the triangle node point  $i$  associated to the global basis.

In Equation 3.14,  $\mathbf{H}^T(x - x_i, y - y_i) \mathbf{b}(x, y)$  are called the correction terms for the kernel functions, which can be obtained via polynomial reproducing conditions. Actually the traditional triangular B-splines already satisfy the partition of unity property, but in some situations this property may be influenced by the unstable nature of the function itself. For example, when the calculated points are quite close to the boundary of the triangular element, this property may not be guaranteed. This is the main reason why we apply the reproducing kernel approximation technique to reproduce the basis functions. In this way, the improved basis can always maintain the partition of unity property even for some hard situations.

In Equation 3.14, the coefficients are determined by the following conditions

$$\sum_{i \in \mathbf{S}} \Phi_i(x, y) = 1, \quad (3.15)$$

### 3.2 Reproducing Kernel Triangular B-spline

---

$$\sum_{i \in \mathbf{S}} \Phi_i(x, y) (x_i^\alpha, y_i^\beta) = x^\alpha y^\beta, \quad (3.16)$$

and

$$\sum_{i \in \mathbf{S}} \Phi_i(x, y) ((x - x_i)^\alpha, (y - y_i)^\beta) = \delta_{|\alpha + \beta|, 0}, \quad |\alpha + \beta| \leq n. \quad (3.17)$$

Equation 3.17 provides the way to calculate the coefficients vector  $\mathbf{b}(x, y)$ , but it is not efficient. For one certain point in the problem domain, we do not need to consider the values of all the basis functions in the set  $\mathbf{S}$  at that point. Since there are only a few global basis functions having supports containing that point, and many others do not contain that point. The value of one point for a basis is considered when we use the Gaussian quadrature rules<sup>66,117</sup> to calculate the variational form. For each basis function, if we loop all the basis to find the coefficients for the correction part, it will be a very laborious calculation.

In<sup>107,108</sup>, the reproducing kernel approximation technique was applied for the kernel correction term globally with all the basis and node points in the domain considered. It is certainly not a desirable way. Here we carry out the reproducing approximation technique locally for the kernel correction term calculation. The compact support property of the triangular B-spline gives  $N_i(\mathbf{x}) = 0$  if  $\mathbf{x}$  is outside the half open convex hull. We can take the advantage of this property, and use a few set of basis related to the studied triangle element instead of the large set  $\mathbf{S}$ . This technique is tested in solving three numerical examples in section 3.3.

For one triangle element, if the order of the triangular B-spline basis is  $n$ , there will be  $N = \frac{(n+1)(n+2)}{2}$  basis functions related to it.  $\mathbf{S}_t$  stands for the set of Triangular B-spline basis in this triangle. Then we rewrite Equations 3.15, 3.16 and 3.17 by

$$\sum_{i \in \mathbf{S}_t} \Phi_i(x, y) = 1, \quad (3.18)$$

$$\sum_{i \in \mathbf{S}_t} \Phi_i(x, y) (x_i^\alpha, y_i^\beta) = x^\alpha y^\beta, \quad (3.19)$$



### 3.2 Reproducing Kernel Triangular B-spline

---

and

$$\sum_{i \in \mathbf{S}_t} \Phi_i(x, y) ((x - x_i)^\alpha, (y - y_i)^\beta) = \delta_{|\alpha||\beta|, 0}, \quad |\alpha + \beta| \leq n. \quad (3.20)$$

Continuing our calculation, we have

$$\sum_{i \in \mathbf{S}_t} \mathbf{H}^T(x - x_i, y - y_i) \mathbf{b}(x, y) N_i(x, y) \mathbf{H}(x - x_i, y - y_i) = \mathbf{H}(0), \quad (3.21)$$

and

$$\mathbf{bM} = \mathbf{H}(0), \quad (3.22)$$

where

$$\mathbf{M} = \sum_{i \in \mathbf{S}_t} \mathbf{H}^T(x - x_i, y - y_i) N_i(x, y) \mathbf{H}(x - x_i, y - y_i). \quad (3.23)$$

By solving this linear matrix system, we get

$$\mathbf{b} = \mathbf{H}(0) \mathbf{M}^{-1}. \quad (3.24)$$

Finally, we substitute Equation 3.24 into Equation 3.14 and add the correction part, then we obtain

$$\Phi_i(x, y) = \mathbf{H}^T(x - x_i, y - y_i) \mathbf{H}(0) \mathbf{M}^{-1} N_i(x, y) \quad (3.25)$$

for our newly improved triangular B-spline basis.

On the other hand, it is also important to know the derivative of the basis function  $\Phi_i$ , since it is used to calculate the stiffness matrix. We follow the proposed method by [88,89,115](#), continuing to use the small basis set  $\mathbf{S}_t$  to find the coefficients vector  $\mathbf{d}(x, y)$  in the following equation,

$$\nabla \Phi_i(x, y) = \mathbf{H}^T(x - x_i, y - y_i) \mathbf{d}(x, y) N_i(x, y). \quad (3.26)$$

After solving the coefficient vector  $\mathbf{d}(x, y)$  which are similar to Equations 3.21 and

### 3.3 Numerical Examples

---

3.24, we get

$$\sum_{i \in \mathcal{S}_t} \nabla \Phi_i(x, y) \mathbf{H}(x - x_i, y - y_i) = (-1) \nabla \mathbf{H}(0). \quad (3.27)$$

Plugging the results  $\mathbf{d}(x, y)$  into Equation 3.27, we finally have the derivative of the improved triangular B-spline basis function

$$\nabla \Phi_i(x, y) = \mathbf{H}^T(x - x_i, y - y_i) (-1) \nabla \mathbf{H}(0) \mathbf{M}^{-1}(x, y) N_i(x, y). \quad (3.28)$$

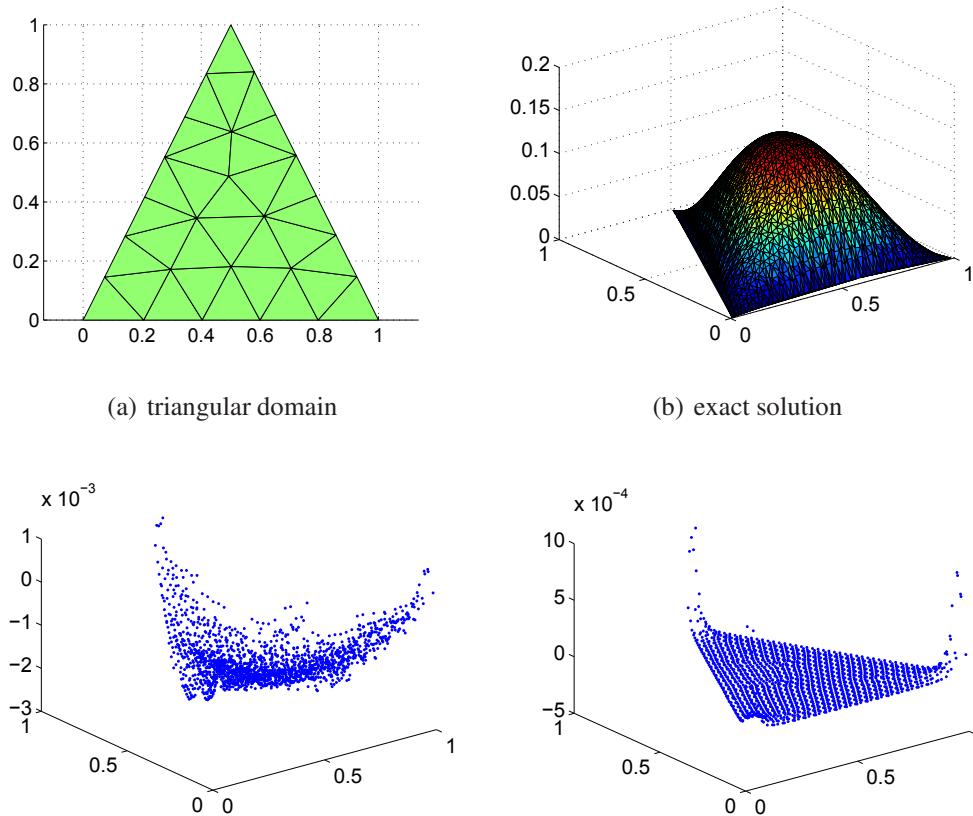
**Remark:** We have studied in detail about how to construct our reproducing kernel triangular B-spline basis function and its derivative. We combine the reproducing kernel approximation technique with the definition of the triangular B-spline, defining different variables for the kernel correction part and the triangular B-spline. In addition, we locally calculate the kernel correction term with a small set of the triangular B-spline according to its compact support property, and thus to improve the computational efficiency. The basis function has been improved. Meanwhile, the derivative of the basis function has also been changed without considering the derivative of the triangular B-spline itself, which is different from the study in subsection 3.2.2.

### 3.3 Numerical Examples

In the previous section, we have introduced the reproducing kernel triangular B-spline-based FEM method. In this section, this method is applied to solve three Poisson's problems.

### 3.3 Numerical Examples

#### 3.3.1 Example I (Triangular Domain)



(c) the distance error produced by the triangular B-spline

(d) the distance error produced by the reproducing kernel triangular B-spline

Figure 3.7: The numerical example I with a triangular domain. (a) is a triangulation of the triangular domain defined by Equation 3.29; (b) is the exact solution defined by Equation 3.31 in the triangular domain; (c) is the distance error produced by the triangular B-spline-based FEM without the reproducing kernel approximation improvement; and (d) is the distance error from the reproducing kernel triangular B-spline-based FEM.

### 3.3 Numerical Examples

---

The first example is a triangular domain as shown in Figure 3.7, and the open set domain  $\Omega$  is defined as

$$\Omega := \{(x, y) \mid ((0 < x \leq \frac{1}{2}) \& (0 < y < 2x)) \cup ((\frac{1}{2} < x < 1) \& (0 < y < 2 - 2x))\}. \quad (3.29)$$

The problem has homogeneous boundary condition. The body force is

$$f(x, y) = 2y + 4, \quad (3.30)$$

and the exact solution is given by

$$u(x, y) = -y(2x - y)(2x + y - 2). \quad (3.31)$$

We compare the  $L^2$  norm errors from the triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM, see the third column in Tables 3.1 and 3.2. It is obvious that the latter is much more accurate numerically. We also take the uniform discretizations with quadratic elements to study the convergence rate, and compare our numerical results with the theoretical ones. In Tables 3.1 and 3.2, the first two columns show the same uniform discretizing conditions for the two methods.  $h$  is called the characteristic length scale. Here we choose the longest edge length of all the triangle elements in the whole problem domain. DOFs stand for the degrees of freedom. As we can see, after each refinement step, the characteristic length scale decreases around by half and the DOFs are increased by around four times. The theoretical optimal convergence rate for  $L^2$  norm errors is 3. For our reproducing kernel triangular B-spline-based FEM, the convergence rate is 2.86, see the last column in Table 3.2. The convergence rate of numerical solutions almost matches with the theoretical result. While for the triangular B-spline-based FEM, the numerical convergence rate is not that good, see in Table 3.1. All the comparison results show that the effect of the reproducing kernel technique applying to the triangular B-spline is tremendous, see Figure 3.8.

### 3.3 Numerical Examples

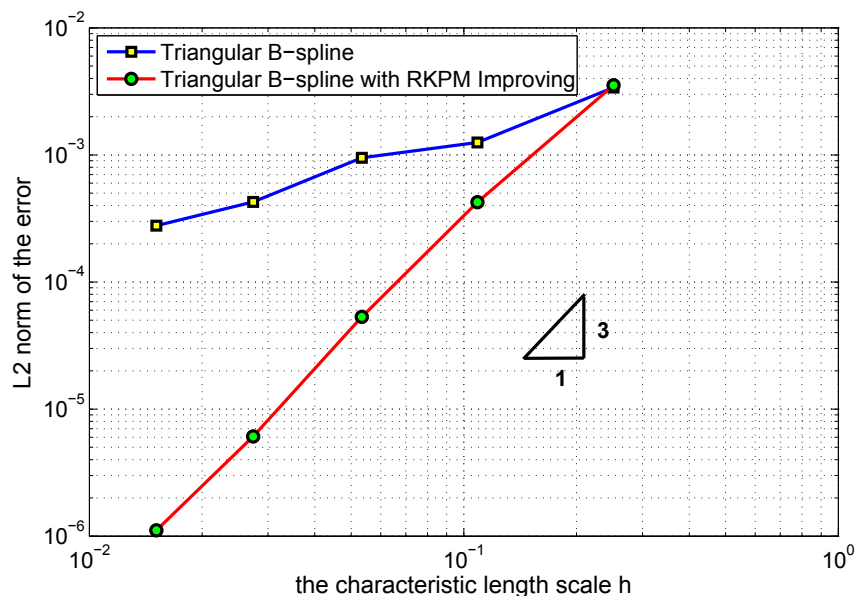


Figure 3.8: Comparison of the convergence rates of the  $L^2$  norms between the traditional triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM.

Table 3.1:  $L^2$  norm of the error by the triangular B-spline

h	DOFs	error( $L^2$ )	convergence rate $L^2$
0.251010779343374	82	0.003395238639438	
0.108696480900126	323	0.001254946562305	1.189195476850933
0.053424097825847	1283	9.499643140926e-04	0.391982020410419
0.027371900088744	5138	4.266453566335e-04	1.196975427770624
0.015102025840206	20533	2.775213782497e-04	0.723160916313839

### 3.3 Numerical Examples

---

Table 3.2:  $L^2$  norm of the error by the reproducing kernel triangular B-spline

h	DOFs	error( $L^2$ )	convergence rate $L^2$
0.251010779343374	82	0.003531520139462	
0.108696480900126	323	4.246381997186e-04	2.530952192422002
0.053424097825847	1283	5.315703381318e-05	2.925516508831231
0.027371900088744	5138	6.090211957827e-06	3.239751538120817
0.015102025840206	20533	1.109591893481e-06	2.863166009635878

#### 3.3.2 Example II (L-shaped Domain)

The second example is an  $L$ -shaped domain as shown in Figure 3.9, and the open set domain  $\Omega$  is defined as

$$\Omega := \{(x, y) \mid ((-1 < x < 1) \& (-1 < y < 1)) \setminus ((0 \leq x \leq 1) \& (-1 \leq y \leq 0))\}. \quad (3.32)$$

In the Example II, we consider homogeneous boundary condition. The body force is

$$f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y), \quad (3.33)$$

and the exact solution is given by

$$u(x, y) = \sin(\pi x) \sin(\pi y). \quad (3.34)$$

### 3.3 Numerical Examples

---

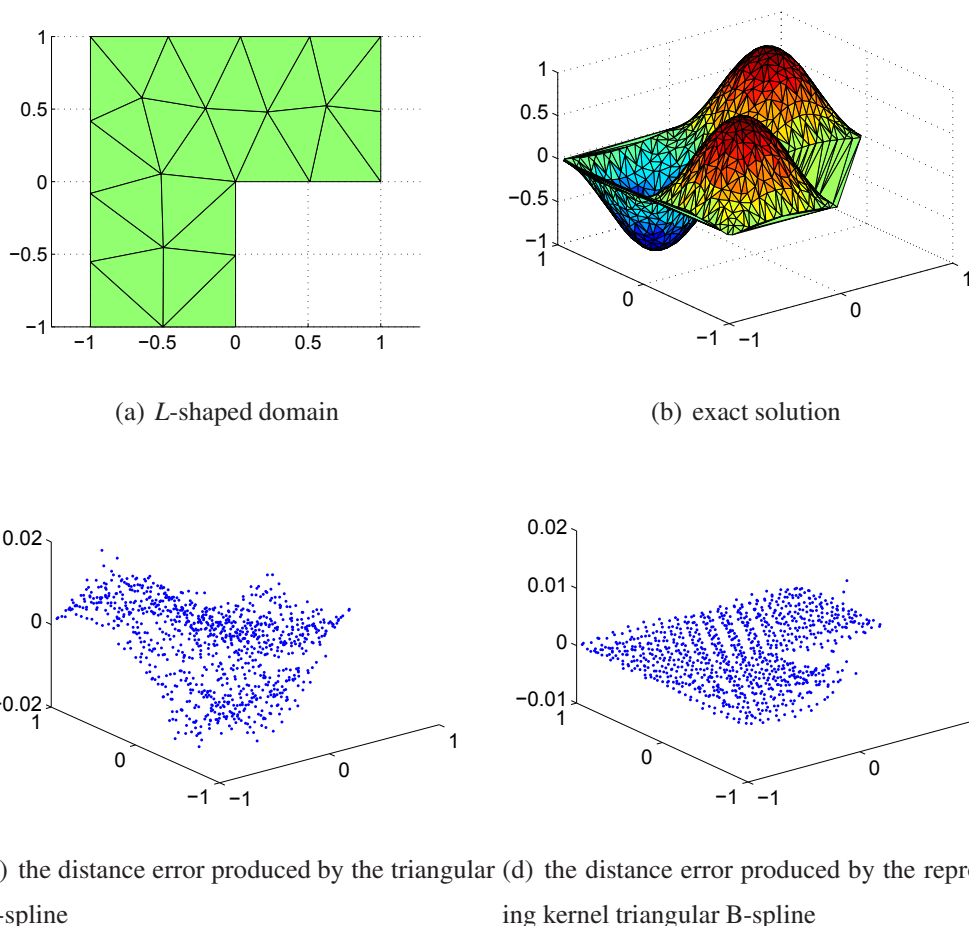


Figure 3.9: The numerical example II with an  $L$ -shaped domain. (a) is a triangulation of the  $L$ -shaped define by Equation 3.32; (b) is the exact solution defined by Equation 3.34 bounded by this  $L$ -shaped domain; (c) is the distance error from the triangular B-spline based-FEM without the reproducing kernel approximation improvement; and (d) is the distance error from the reproducing kernel triangular B-spline-based FEM.

For comparison, the numerical results of the triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM are shown in Tables 3.3 and 3.4 (third column). After each refinement procedure, the  $L^2$  norm errors from the former method are not reduced much. Compared to the efforts that we put into each refinement work, the triangular B-spline-based FEM does not perform well in solving

### 3.3 Numerical Examples

PDEs. However, for the reproducing kernel approximation technique, the errors are reduced significantly by each refinement step and it works in a very good situation. In addition, from the distance error plot in Figures 3.9(c) and 3.9(d), the reproducing kernel approximation technique adjusts the triangular B-spline basis, reducing many unexpected error from its instability. It is also interesting to point out that the  $L^2$  numerical convergence rate of our improved method is 2.77, see the fourth column in Table 3.4. Compared to the triangular B-spline-based FEM (the fourth column in Table 3.3), our improved numerical convergence rate almost recovers the theoretical optimal convergence rate. Based on the numerical error comparison, the approximation of the reproducing kernel triangular B-splined-based FEM is much more accurate and efficient than that of the triangular B-spline-based FEM, see Figure 3.10.

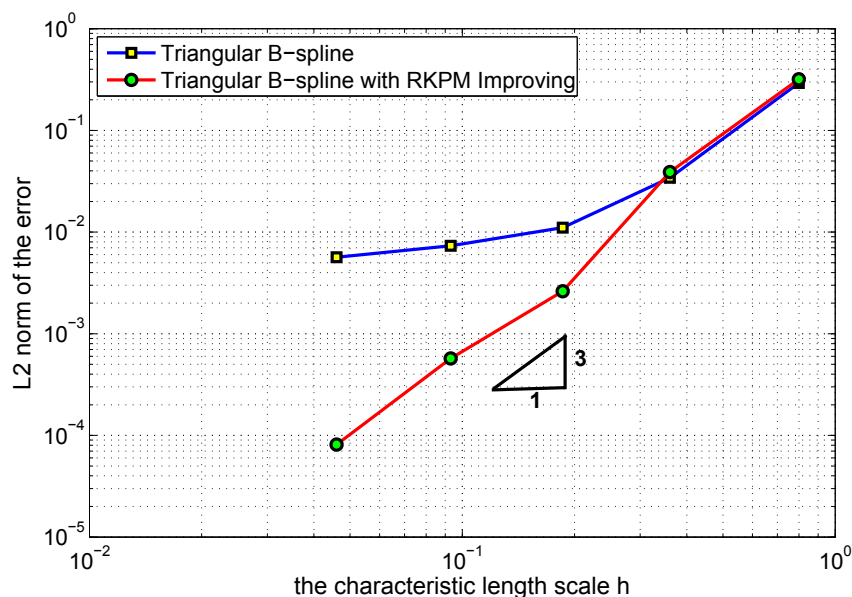


Figure 3.10: Comparison of the convergence rates of the  $L^2$  norms between the traditional triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM.



### 3.3 Numerical Examples

Table 3.3:  $L^2$  norm of the error by the triangular B-spline

h	DOFs	error( $L^2$ )	convergence rate $L^2$
0.799159180421800	61	0.293033363538699	
0.360513549037688	199	0.034229492552127	2.697382255541688
0.185904692351987	712	0.011073158023197	1.704018534183580
0.093089256218647	2860	0.007337114913702	0.595045459934525
0.045997931931117	11387	0.005649054917630	0.370881431738639

Table 3.4:  $L^2$  norm of the error by the reproducing kernel triangular B-spline

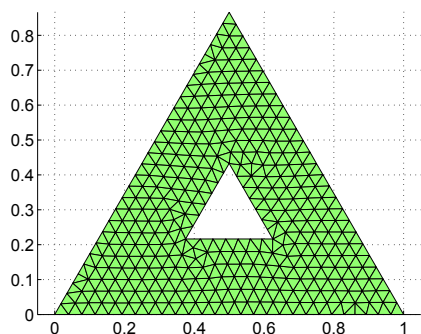
h	DOFs	error( $L^2$ )	convergence rate $L^2$
0.799159180421800	61	0.318970987027548	
0.360513549037688	199	0.038910165397480	2.642919233912530
0.185904692351987	712	0.002621118697456	4.073188957089076
0.093089256218647	2860	5.710786472836e-04	2.203099379560828
0.045997931931117	11387	8.113501389344e-05	2.768108622317826

#### 3.3.3 Example III (Triangle with A Hole)

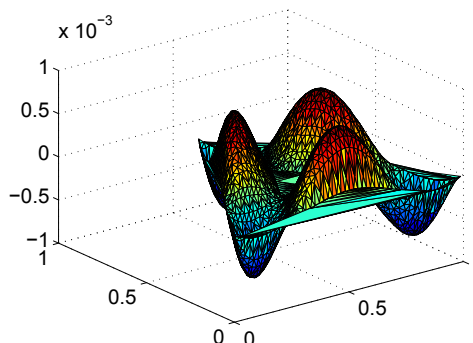
The last example is a triangle with a hole shaped domain as shown in Figure 3.11, and the open set domain  $\Omega$  is defined as

$$\begin{aligned}
 \Omega := \{ & (x, y) \mid [(0 < x \leq \frac{1}{2}) \& (0 < y < \sqrt{3}x) \cup \\
 & (\frac{1}{2} < x < 1) \& (0 < y < \sqrt{3} - \sqrt{3}x)] \setminus \\
 & [(\frac{3}{8} \leq x \leq \frac{1}{2}) \& (\frac{\sqrt{3}}{8} \leq y \leq \sqrt{3}x - \frac{\sqrt{3}}{4}) \cup \\
 & (\frac{1}{2} < x \leq \frac{5}{8}) \& (\frac{\sqrt{3}}{8} \leq y \leq \frac{3\sqrt{3}}{4} - \sqrt{3}x)] \}. \tag{3.35}
 \end{aligned}$$

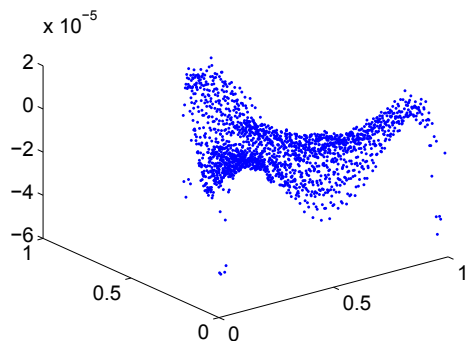
### 3.3 Numerical Examples



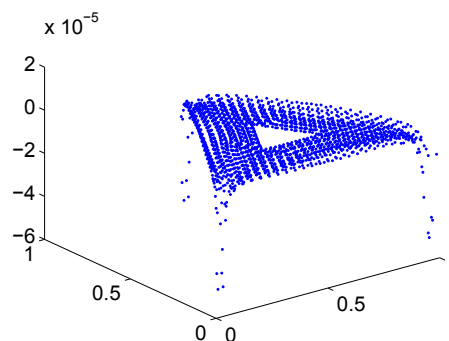
(a) triangle with a hole



(b) exact solution



(c) the distance error produced by the Triangular B-spline



(d) the distance error produced by the reproducing kernel Triangular B-spline

Figure 3.11: The numerical example III governed by a triangle with a hole shaped domain. (a) is a triangulation of the triangle with a hole shaped domain defined in Equation 3.35; (b) is the exact solution defined by Equation 3.37 in this triangle with a hole shaped domain; (c) is the distance error produced by the triangular B-spline-based FEM without the reproducing kernel approximation improvement; and (d) is the distance error from the reproducing kernel triangular B-spline-based FEM.

In Example III, we consider homogeneous boundary condition. The body force is

$$\begin{aligned}
 f(x,y) = & 18x^4 + 36x^3 + 36x^2y^2 + 18\sqrt{3}x^2y - 18x^2 - 36xy^2 \\
 & - 18\sqrt{3}xy + 18y^4 - 22\sqrt{3}y^3 + 36y^2 - \frac{27}{64},
 \end{aligned} \tag{3.36}$$

### 3.3 Numerical Examples

---

and the exact solution is given by

$$u(x,y) = -y(\sqrt{3}x - y)(\sqrt{3}x + y - \sqrt{3})(y - \frac{\sqrt{3}}{8})$$

$$(\sqrt{3}x - \frac{\sqrt{3}}{4} - y)(y + \sqrt{3}x - \frac{3\sqrt{3}}{4}). \quad (3.37)$$

As the refinement procedure goes on, the reproducing kernel triangular B-spline-based FEM becomes more accurate. In addition, the numerical convergence rate of the reproducing kernel B-spline-based FEM is 2.78 (see the last column in Table 3.6), which is close to the theoretical optimal convergence rate 3. Similarly to the previous two examples, Figure 3.12 shows the log-log plot, comparing the convergence rates between the triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM. The slopes of the two segments represent the convergence rates of these two methods. The comparison verifies that the reproducing kernel approximation technique significantly improves the triangular B-spline in solving PDEs.

Table 3.5:  $L^2$  norm of the error by the triangular B-spline

h	DOFs	error( $L^2$ )	convergence rate $L^2$
0.051514399020636	1350	6.668310558137732e-06	
0.025080509709024	5235	3.596384903833970e-06	0.857825575052465
0.012808319353125	20904	2.173145634612668e-06	0.749637395571607
0.006213076467145	83809	1.546039483984449e-06	0.470640035412379

Table 3.6:  $L^2$  norm of the error by the reproducing kernel triangular B-spline

h	DOFs	error( $L^2$ )	convergence rate $L^2$
0.051514399020636	1350	2.741116610044200e-06	
0.025080509709024	5235	3.874866803318393e-07	2.718143694859875
0.012808319353125	20904	5.213907020582597e-08	2.984789362245809
0.006213076467145	83809	6.969382234436867e-09	2.781697884926696

### 3.3 Numerical Examples

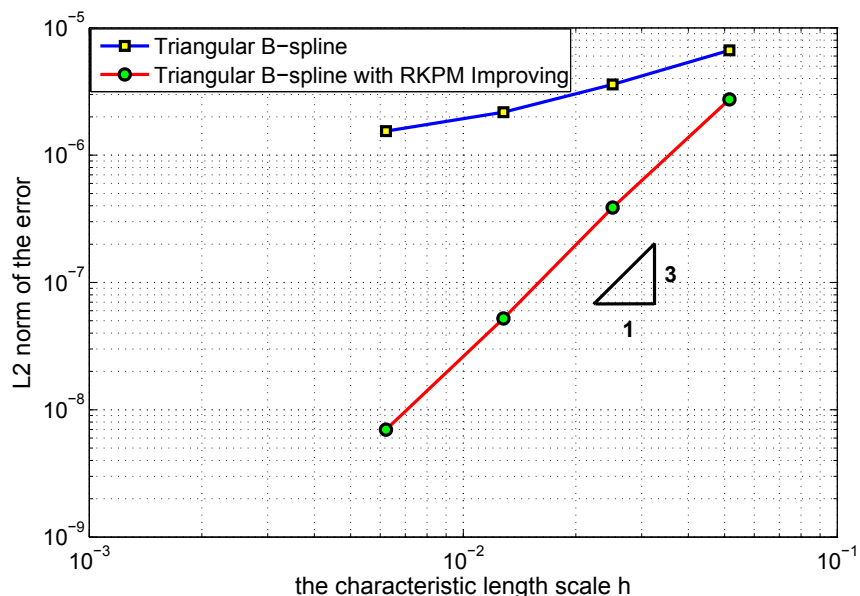


Figure 3.12: Comparison of the convergence rates of the  $L^2$  norms between the traditional triangular B-spline-based FEM and the reproducing kernel triangular B-spline-based FEM.

**Discussion:** In this section, we have numerically studied the reproducing kernel triangular B-spline-based FEM, solving three different problems based on the Poisson's equation over a triangular domain, an  $L$ -shaped domain and a triangle with a hole shaped domain, respectively. We also applied the triangular B-spline-based FEM to solve the same problems, and compared their results. We found that the two methods have quite different effects in terms of the  $L^2$  norm errors and the numerical convergence rates. This is mainly due to the instability of the triangular B-spline functions. For our proposed method, the errors are tremendously reduced, and the numerical convergence rates are also very close to the optimal convergence rate. These comparisons verify that the reproducing kernel approximation technique plays an important role in controlling the performance of the triangular B-spline and preventing many unexpected errors occurring.

### 3.4 Conclusions

---

In addition, we also compared our work with NURBS-based methods. In general, the performance of NURBS outperforms the triangular B-splines. However, NURBS and T-splines need a lot of work on local refinement to handle complicated geometry with sharp feature along arbitrary directions. While due to their special triangular shape structure, triangular B-splines can easily preserve arbitrary features. This is a very important advantage for triangular B-splines compared to NURBS-based methods.

### 3.4 Conclusions

In summary, we developed an improved reproducing kernel triangular B-spline-based FEM to solve PDE problems. Nowadays applying the triangular B-spline to solve PDEs is still staying at the beginning stage. Besides, the performance of triangular B-splines is quite unstable which leads to many uncontrolled errors occurring almost everywhere. To overcome its instability, we take the reproducing kernel approximation technique and calculate the kernel correction term locally to construct general unity shape functions. The numerical results show the tremendous improvement.

As part of the future work in this part, we plan to apply our algorithm to complicated problems, study singular and locking problems. In addition, we would also like to extend our technique to 3D based on tetrahedral meshes and the trivariate triangular B-spline.

# Chapter 4

## Extended Isogeometric Analysis for Material Interface Problems

### 4.1 Introduction

We propose an approach to extend the IGA (Isogeometric Analysis) method to solve material interface problems. The development is carried out through incorporating the advantages of the extended finite element method into the standard IGA approach for solving problems with discontinuities. By applying both the XIGA and IGA methods to solve the Poisson's equation problem containing weak discontinuities, we demonstrate that the XIGA achieves the optimal convergence rate while the IGA only converges suboptimally. The proposed method is then successfully applied to solve bimaterial and curved material interface problems.

### 4.2 Extended Isogeometric Analysis

#### 4.2.1 Enriched Basis Functions Selection

The enrichment means to expand the traditional continuous approximation space of IGA by adding particular functions which have non-smooth characteristics. There are

## 4.2 Extended Isogeometric Analysis

---

two types of enrichments. The first one contains the discontinuities in the displacement field, which is suitable for modeling strong discontinuous problems. The second type includes the discontinuities in the gradient field, which is used for modeling weak discontinuous problems. Our study will focus on this latter category.

In XIGA, only a subset of the original NURBS basis functions needs to be enriched. The first step is thus to distinguish which functions should be enriched. We only enrich the basis functions that have support in the elements containing the discontinuity.

The coordinates of all nodes of the element are found and compared with the material interface, by using the Signed-Distance Function as follows

$$\xi(\mathbf{x}) = \min_{\mathbf{x}_\Gamma \in \Gamma} \|\mathbf{x} - \mathbf{x}_\Gamma\| \text{sign}(\mathbf{n} \cdot (\mathbf{x} - \mathbf{x}_\Gamma)) , \quad (4.1)$$

where  $\mathbf{x}$  can be any point in the domain. Here  $\mathbf{x}$  stands for one node of an element.  $\mathbf{x}_\Gamma$  is the normal projection of  $\mathbf{x}$  onto the material interface  $\Gamma$ .  $\mathbf{n}$  is a normal unit vector. If all nodes of one element are on one side of the material interface, they are given the same sign. Otherwise some nodes will have the different signs with the others in the same element.

After applying the signed-distance function (4.1), there will be three possible situations as shown in Figure 4.1. It shows that these points can have zero distance (o), positive distance (+) or negative distance (-). In subsection 4.3.1 Example I and subsection 4.3.2 Example II, we will apply Signed-Distance Function (4.1) to distinguish the types of elements.

**Remark:** The signed-distance function can be applied to both parametric and physical domains. In our study, we define the enriched basis functions in the parametric domain, because all the basis functions of IGA are originally defined in the parametric domain. The supports of each basis function are cut by straight lines within a square. The parametric space is also used for integration, where normally each integration cell is a quadrilateral. It is easier to map just the discontinuities in the parametric space and construct enrichment based on the image of the interface segment than to construct

## 4.2 Extended Isogeometric Analysis

the enrichment functions in the physical space and map them back to the parametric space. The method needed to find the parametric coordinates of the discontinuities is discussed in the Section 4.2.5.

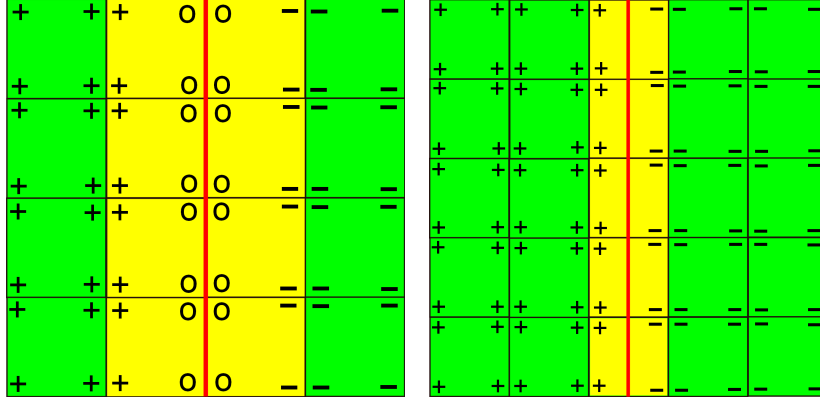


Figure 4.1: Applying the signed-distance function to distinguish the types of elements.

The XIGA method<sup>47,48</sup> follows the PU framework of<sup>55,118</sup>. The main idea is to extend the classical solution space through multiplying the enrichment functions by the subset of these same basis functions which can ensure a conforming approximation. We define one set  $\mathbf{N} = \{ N_1, N_2, \dots, N_n \}$ ,  $N_i$  are the original IGA basis functions. The basis set of XIGA (represented by  $\mathbf{N}^*$ ) is the union of two sets,  $\mathbf{N}^* = \mathbf{N} \cup \mathbf{M}$ , where  $\mathbf{M}$  is the new enriched basis functions set,  $\mathbf{M} = \{ M_1, M_2, \dots, M_m \}$ .  $M_j$  are constructed by multiplying  $N_i$  with a certain enrichment function  $\psi$ , i.e  $M_1 = N_{\hat{1}} \cdot \psi$ ,  $M_2 = N_{\hat{2}} \cdot \psi, \dots, M_m = N_{\hat{m}} \cdot \psi$ . We use a different subscript notation of  $N_{\hat{i}}$  from that of  $N_i$ , since  $N_{\hat{i}}$  are the basis functions needed to be enriched. The approximation of XIGA can be expressed by:

$$u_h = \underbrace{\sum_{i \in \mathbf{N}} N_i u_i}_{\mathbf{u}^{st}} + \underbrace{\sum_{j \in \mathbf{M}} N_j \psi a_j}_{\mathbf{u}^{enr}} . \quad (4.2)$$

In the above, we have introduced the framework of IGA and XIGA. In the rest of Section 4.2, we will present some details and techniques applied in our XIGA method.



### 4.2.2 Enrichment Functions

#### 4.2.2.1 Ramp Enrichment Function

One kind of the enrichment functions is called ramp enrichment functions. More details about the ramp enrichment function can be found in <sup>119</sup>. We will apply it in our numerical Example I. The ramp enrichment functions are formed by two independent parts  $\phi_1$  and  $\phi_2$ :

$$\phi_1 = \begin{cases} x - 1/2 & \text{if } x \geq 1/2 \\ 0 & \text{otherwise ,} \end{cases} \quad (4.3)$$

$$\phi_2 = \begin{cases} -x + 1/2 & \text{if } x < 1/2 \\ 0 & \text{otherwise .} \end{cases} \quad (4.4)$$

If  $N_i$  stand for the standard basis functions, then  $N_i \cdot \phi_1$  and  $N_i \cdot \phi_2$  are two new enriched basis functions. The extended approximation can be written as:

$$u_h = \underbrace{\sum_{i \in \mathbf{S}} N_i u_i}_{\mathbf{u}^{st}} + \underbrace{\sum_{j \in \mathbf{S}_{e1}} N_j \phi_1 a_j}_{\mathbf{u}^{enr1}} + \underbrace{\sum_{k \in \mathbf{S}_{e2}} N_k \phi_2 b_k}_{\mathbf{u}^{enr2}} ,$$

where  $\mathbf{S}$  contains the indices of all global basis functions.  $\mathbf{S}_{e1}$  and  $\mathbf{S}_{e2}$  contain the indices of the enriched basis functions. We define another two ramp enrichment functions  $\varphi_1$  and  $\varphi_2$  for Example III:

$$\varphi_1(x,y) = \begin{cases} \sqrt{x^2 + y^2} - \frac{1}{2} & \text{if } \frac{1}{4} \leq x^2 + y^2 \\ 0 & \text{otherwise ,} \end{cases} \quad (4.5)$$

$$\varphi_2(x,y) = \begin{cases} -\sqrt{x^2 + y^2} + \frac{1}{2} & \text{if } x^2 + y^2 \leq \frac{1}{4} \\ 0 & \text{otherwise .} \end{cases} \quad (4.6)$$

## 4.2 Extended Isogeometric Analysis

---

### 4.2.2.2 Moës Enrichment Function

Another type of enrichment function is called the Moës enrichment function. The detailed discussion about Moës enrichment functions can be seen in<sup>120</sup>. We will apply it for solving numerical Example II. It is defined by:

$$\psi_1(x,y) = \begin{cases} \frac{\sqrt{x^2+y^2} - \sqrt{x_1^2+y_1^2}}{1/2 - \sqrt{x_1^2+y_1^2}} & \text{if } \sqrt{x_1^2+y_1^2} \leq \sqrt{x^2+y^2} \leq \frac{1}{2} \\ \frac{\sqrt{x_2^2+y_2^2} - \sqrt{x^2+y^2}}{\sqrt{x_2^2+y_2^2} - 1/2} & \text{if } \frac{1}{2} < \sqrt{x^2+y^2} \leq \sqrt{x_2^2+y_2^2} \end{cases} \quad (4.7)$$

where  $(x_1, y_1)$  and  $(x_2, y_2)$  are two neighbor points of material interface shown in Figure 4.2.

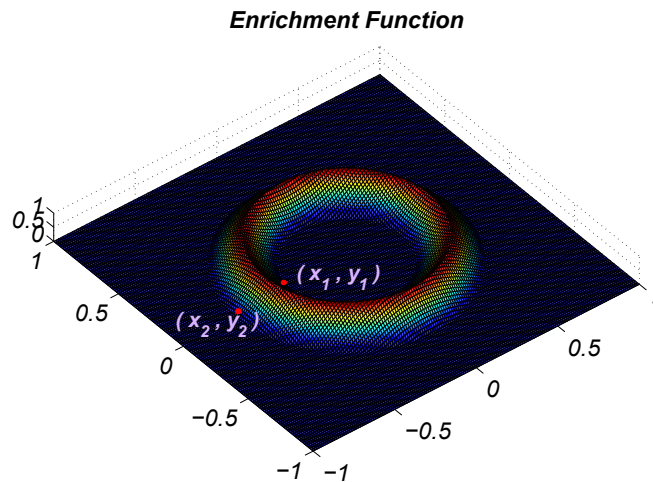


Figure 4.2: the enrichment function  $\psi_1(x,y)$

### 4.2.3 Greville Abscissae

In this subsection, we will introduce a way of selecting control points using Greville abscissae. This technique will be applied in our following numerical examples, its

## 4.2 Extended Isogeometric Analysis

---

effects can be seen clearly in Figure 4.7. For quadratic and higher orders, if control points are uniformly spaced, the mapping from the parameter space to the physical space is no longer identity. However, the Greville Abscissae can ensure this mapping to be an identity mapping. For more details, readers are referred to<sup>93</sup>.

In the Greville Abscissae, the control points  $\mathbf{C} = (C_x, C_y)$  are calculated by:

$$C_x(i) = \frac{\sum_{j=i+1}^{i+p} \xi_j^u}{p} \quad C_y(i) = \frac{\sum_{j=i+1}^{i+q} \xi_j^v}{q}, \quad (4.8)$$

where  $\{ \xi_k^u \mid k = 1, \dots, n_u \}$  and  $\{ \xi_k^v \mid k = 1, \dots, n_v \}$  are the knots in  $u$  and  $v$  directions respectively.  $n_u$  and  $n_v$  represent the numbers of NURBS basis functions in each direction.

### 4.2.4 Repeating Middle Neighbour Knots

Repeating middle neighbour knots uses a unique property of NURBS functions. Recall that the  $p^{\text{th}}$  order NURBS function has support over  $p + 1$  elements. We can repeat the neighbor knots of the material interface so that we can reduce the support of some NURBS functions.

Figure 4.3 illustrates how the repeating neighbor knots works. On the left plot of Figure 4.3, we want to enrich the elements in yellow color. All the basis functions that have supports in these elements will be enriched. A basis function has support in several elements, since each element is not isolated from others. As a result, some extra elements may be enriched, when these yellow elements are enriched. These elements are called blending elements in XFEM. In these blending elements, only part of the global basis functions will be enriched. The accuracy of the approximation is lower in these blending elements, as is discussed in<sup>42</sup>.

## 4.2 Extended Isogeometric Analysis

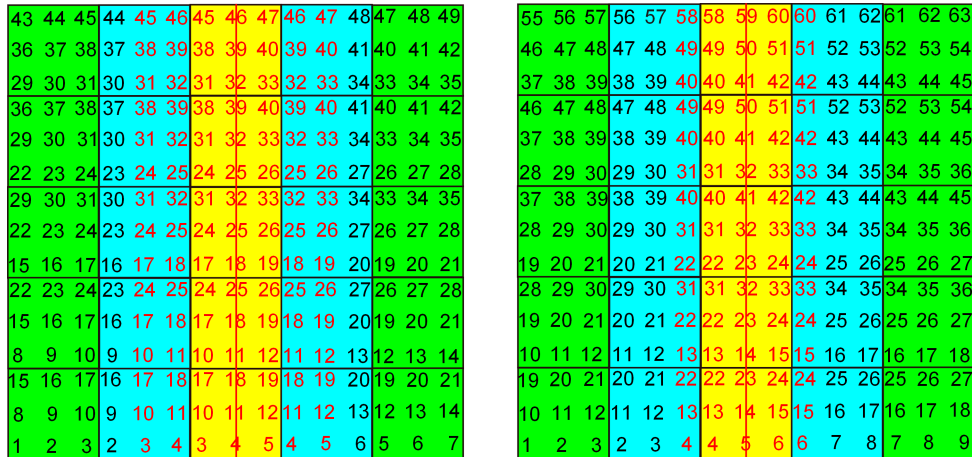


Figure 4.3: the basis functions distribution before repeating middle neighbour knots (left) after repeating middle neighbour knots (right)

The distribution of the global basis functions is shown in the right panel of Figure 4.3 after applying the repeating middle neighbor knots. The repeating middle neighbor knots decreases the spans for the basis functions in blending elements. However an obvious disadvantage is that a larger number of degrees-of-freedom is needed, which leads to more expensive calculation cost.

### 4.2.5 Inverse Mapping

The basis functions are defined in the parametric space, and the numerical integration is also done in the parametric space. However usually the problems are described in the physical space. The material interface can have different shapes in the parametric and physical spaces. Hence, an inverse mapping needs to relate these two different shapes. The inverse mapping is set up by finding a pair of parametric coordinates  $(u, v)$  and satisfying the distance equation:

$$\mathbf{r}(u, v) = \mathbf{S}(u, v) - \mathbf{P} = \mathbf{0} , \quad (4.9)$$

## 4.2 Extended Isogeometric Analysis

---

where  $\mathbf{P}$  stands for the coordinates of the material interface.  $\mathbf{S}$  stands for the coordinates of the NURBS surface point. Both points are defined in the physical space.

The Newton-Raphson method is applied to solve the nonlinear system (4.9). Here the function  $\mathbf{r}(u, v)$  is continuous and has a continuous derivative  $\nabla\mathbf{r}(u, v)$ . (4.9) can be expanded in the Taylor series:

$$\begin{aligned}\mathbf{r}(u, v) + \delta\mathbf{u}\nabla\mathbf{r}(u, v) &= \mathbf{0} \ , \\ \delta\mathbf{u}\nabla\mathbf{S}(u, v) &= -(\mathbf{S}(u, v) - \mathbf{P}) \ .\end{aligned}\tag{4.10}$$

The solution of this linear system is of the form:

$$\delta\mathbf{u} = \begin{pmatrix} u_{n+1} - u_n \\ v_{n+1} - v_n \end{pmatrix}^T, \tag{4.11}$$

$$\begin{aligned}u_{n+1} &= u_n + \delta u, \\ v_{n+1} &= v_n + \delta v.\end{aligned}$$

**Remark:** We note that it is important to choose proper initial values  $u_0$  and  $v_0$ , the tolerance  $\varepsilon > 0$ , and the maximum number of iterations  $N$ . The inverse mapping is done iteratively using the gradient values which gives good results in many applications. Finding good initial values usually depends on the scale and the shape of the problem domain studied. For example, in our third numerical example, we initially choose  $u_0 = 0$  and  $v_0 = 0$ . Then in each iterative step, we check the stopping criteria  $\|\mathbf{S}(u, v) - \mathbf{P}\| \leq \varepsilon$  with the tolerance  $\varepsilon = 10^{-7}$  in our implementation.

### 4.2.6 Curve Fitting

The inverse mapping is needed for finding the points in the parametric space corresponding to the material interface. These points are discrete, so they need to be re-

## 4.2 Extended Isogeometric Analysis

---

computed for each curved material interface problem. In Example III, we introduce normal triangular elements and curved triangular elements for solving the numerical integration properly. Finding two intersection points in each enriched elements is very important, so it is better computing the path of this curve in the parametric space in the pre-processing stage. The reference<sup>92</sup> introduces several methods for curve fitting, here we apply the least squares curve approximation method to calculate the control points of the B-spline curve of the material interface in the parametric space.

All weights of the control points are set to 1.  $p$  stands for the polynomial order of the basis functions.  $n$  is the number of the control points.  $Q_0, Q_1, \dots, Q_{m-1}, Q_m$  are data points from the inverse mapping. We look for a  $p$ th degree non-rational curve:

$$\mathbf{C}(\lambda) = \sum_{i=1}^n N_{i,p}(\lambda) \mathbf{P}_i \quad \lambda \in [0, 1] . \quad (4.12)$$

These data points  $\mathbf{Q}_k$ , with  $k = 1, 2, \dots, m-1, m$ , are approximated in the least squares sense :

$$\begin{aligned} f &= \sum_{k=1}^m |\mathbf{Q}_k - \mathbf{C}(\hat{u}_k)|^2 \\ &= \sum_{k=1}^m (\mathbf{Q}_k - \mathbf{C}(\hat{u}_k))(\mathbf{Q}_k - \mathbf{C}(\hat{u}_k)) \\ &= \sum_{k=1}^m [\mathbf{Q}_k \mathbf{Q}_k - 2\mathbf{Q}_k \mathbf{C}(\hat{u}_k) + \mathbf{C}(\hat{u}_k) \mathbf{C}(\hat{u}_k)] \\ &= \sum_{k=1}^m [\mathbf{Q}_k \mathbf{Q}_k - 2 \sum_{i=1}^n N_{i,p}(\hat{u}_k) (\mathbf{Q}_k \cdot \mathbf{P}_i) + (\sum_{i=1}^n N_{i,p}(\hat{u}_k) \mathbf{P}_i) (\sum_{i=1}^n N_{i,p}(\hat{u}_k) \mathbf{P}_i)] \end{aligned} \quad (4.13)$$

The standard technique for least squares curve fitting is to minimize  $f$  by setting the derivatives of  $f$  with respect to the  $n$  points equal to zeros :

$$\frac{\partial f}{\partial \mathbf{P}_l} = \sum_{k=1}^m [-2N_{l,p}(\hat{u}_k) \mathbf{Q}_k + 2N_{l,p}(\hat{u}_k) \sum_{i=1}^n N_{i,p}(\hat{u}_k) \mathbf{P}_i] , \quad (4.14)$$

## 4.2 Extended Isogeometric Analysis

---

which yields

$$\sum_{i=1}^n \left( \sum_{k=1}^m N_{l,p}(\hat{u}_k) N_{i,p}(\hat{u}_k) \right) \mathbf{P}_i = \sum_{k=1}^m N_{l,p}(\hat{u}_k) \mathbf{Q}_k , \quad (4.15)$$

$$(\mathbf{N}^T \mathbf{N}) \mathbf{P} = \mathbf{N}^T \mathbf{Q} , \quad (4.16)$$

$\mathbf{N}$  is the  $m \times n$  matrix

$$\mathbf{N} = \begin{pmatrix} N_{1,p}(\hat{u}_1) & N_{2,p}(\hat{u}_1) & \cdots & N_{n,p}(\hat{u}_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{1,p}(\hat{u}_m) & N_{2,p}(\hat{u}_m) & \cdots & N_{n,p}(\hat{u}_m) \end{pmatrix} , \quad (4.17)$$

$$\begin{aligned} \mathbf{Q} &= [\mathbf{Q}_1 \ \mathbf{Q}_2 \ \cdots \ \mathbf{Q}_m] \\ \mathbf{P} &= [\mathbf{P}_1 \ \mathbf{P}_2 \ \cdots \ \mathbf{P}_n] , \end{aligned} \quad (4.18)$$

with  $\mathbf{P} = (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \mathbf{Q}$ .

Before solving the least square system, we should build a parametric coordinates consequence  $\{\hat{u}_1, \hat{u}_2, \dots, \hat{u}_m\}$  corresponding to the data points  $\mathbf{Q}_k, k = 1, 2, \dots, m$ . There are several ways to get this set. A good way is to calculate the chord length.  $d$  is the total chord length defined by

$$d = \sum_{k=2}^m | \mathbf{Q}_k - \mathbf{Q}_{k-1} | . \quad (4.19)$$

We have  $\hat{u}_1 = 0, \hat{u}_m = 1$

$$\hat{u}_k = \hat{u}_{k-1} + \frac{| \mathbf{Q}_k - \mathbf{Q}_{k-1} |}{d} , \quad k = 1, 2, \dots, m-1 \quad (4.20)$$

We now find the knots vector, which can be obtained by the equally spaced method or the averaging technique. We will use the averaging technique, which is recommended

## 4.2 Extended Isogeometric Analysis

---

by reference<sup>92</sup>. Following this technique, we state by calculating,

$$u_1 = \cdots = u_{p+1} = 0 ,$$

$$u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p} \hat{u}_i \quad j = 1, 2, \cdots, n-p+1 ,$$

$$u_{m+1-p} = \cdots = u_{m+2} = 1 .$$

### 4.2.7 Intersection Points

Now we can rebuild the material interface in the parametric space by using the results of the curve fitting. The curved material interface has its own knots vector and the control points, so the parametric space mesh is independent of the material interface. In this section, we are going to apply the Newton-Raphson method to find the intersection points between the material interface and the parametric mesh. We have,

$$r(\lambda) = C(\lambda) - \xi , \tag{4.21}$$

An illustration is shown in Figure 4.4. We loop the edges of the element for each quadrilateral enriched element. For a horizontal edge, we take  $\xi$  to be the  $v$  coordinate value of the edge. We take  $\xi$  to be the  $u$  coordinate value of the edge for a vertical edge.



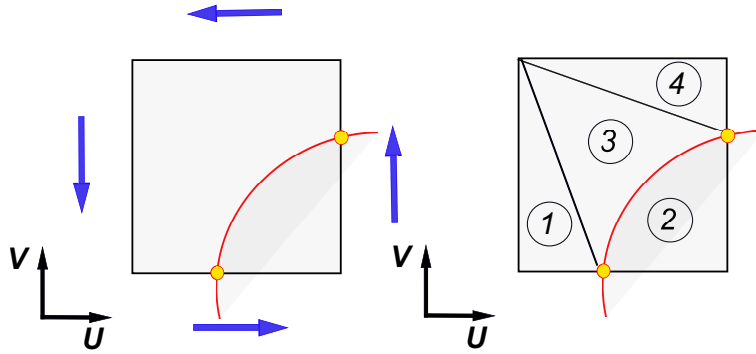


Figure 4.4: The element is crossed by the curved material interface (left). The enriched element is divided by triangulation (right).

As mentioned in the inverse mapping, we always have to consider and pick some proper initial values for starting the iteration procedure. Here, we use the previous calculation of the fitted curve. We pick the first knot of the curve as the initial value of  $\lambda_0$ . We could also reinitialize the value of  $\lambda_0$  by replacing with sequences of known knots inside the knots vector of the curve. After picking the values for  $\lambda_0$ , we iterate using the standard Newton - Raphson iteration:

$$\lambda_{n+1} = \lambda_n - \frac{r(\lambda_n)}{r'(\lambda_n)}. \quad (4.22)$$

Finally we find the parametric value corresponding to the intersection points on the material interface. In our case, after the loop for each enriched element, we should get two parametric value  $\lambda_1$  and  $\lambda_2$ . We do not need to calculate  $\lambda$  for the special case, where the curve has only one intersection point in the corner of the element.

### 4.2.8 Triangular Integration

In the Example III, we need to consider more details for the numerical integration. Because of the curved shape material interface, curved integration elements are needed to improve the quality of the domain discretization. The curved element development is

## 4.2 Extended Isogeometric Analysis

---

derived from introducing isoparametric elements for handling curved boundaries in <sup>121</sup>. The triangular element with one curved edge is analyzed mathematically in <sup>122,123,124</sup>, etc. <sup>125</sup> also introduced transfinite elements which consist of a reference square mapped to a subdomain with curved boundaries. Our curved triangular elements construction is based on the study in <sup>126,127,128</sup>. We calculate the coordinates of the intersection points for each enriched element that is crossed by the material interface. We use the two intersection points together with the four nodes of the quadrilateral element for constructing a Delaunay triangulation. Since the material interface is curved in the parametric space, we have two triangles containing curved edges for matching it. For each quadrilateral element, there are two kinds of triangles, i.e normal straight side triangles and curved edge triangles. For the normal triangle, we do the standard triangle transformation  $T_1$  which is shown in Figure 4.5.

$$T_1 : \xi \rightarrow (1 - \hat{\xi}^1 - \hat{\xi}^2)\xi_1 + \hat{\xi}^1\xi_2 + \hat{\xi}^2\xi_3 , \quad (4.23)$$

where  $\xi_1$ ,  $\xi_2$  and  $\xi_3$  are the coordinates of the standard parent triangular element.  $(\hat{\xi}^1, \hat{\xi}^2)$  are the coordinates of the Gaussian point.  $\xi$  is the new coordinates of the Gaussian point after transformation. For a curved triangle, we apply the transformation  $T_2$  for constructing a curved edge triangle. This transformation is also shown in Figure 4.5.

$$T_2 : \xi \rightarrow \frac{1 - \hat{\xi}^1 - \hat{\xi}^2}{1 - \hat{\xi}^1} C(\lambda(\hat{\xi}^1)) + \frac{\hat{\xi}^1 \hat{\xi}^2}{1 - \hat{\xi}^1} \xi_2 + \hat{\xi}^2 \xi_3 , \quad (4.24)$$

where  $C(\lambda(\hat{\xi}^1))$  is the B-spline curve at point  $\lambda(\hat{\xi}^1)$  and  $\lambda(\hat{\xi}^1) = \lambda_1 + (\lambda_2 - \lambda_1)\hat{\xi}^1$ ,  $\lambda_1$  and  $\lambda_2$  are the parametric coordinates of the two intersection points  $\xi_1$  and  $\xi_2$ . Using  $T_1$  and  $T_2$  allows us to consider the exact geometry in the integration process.

### 4.3 Numerical Examples

---

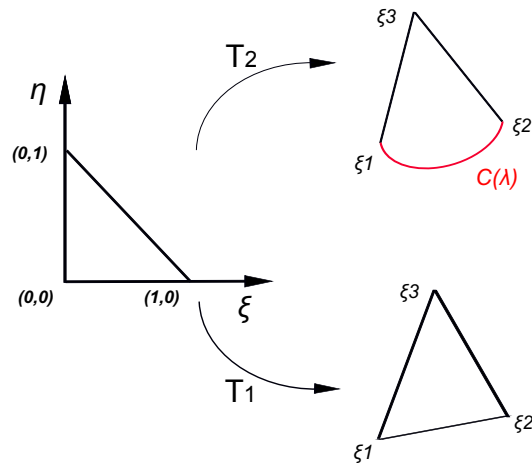


Figure 4.5: Triangle transformation

**Remark:** Carrying out the integration efficiently and accurately needs particular attention, especially for the basis functions which contain discontinuities in the approximation space, since the Gaussian quadrature is not exact there. For example, for model III in section 3.3 with a curved material interface in parametric space, the curved triangles are really necessary for obtaining accurate approximation. The first reason is that the curved triangle are needed to maintain all the Gaussian points which are inside the triangle on one side of the discontinuity. Once the curved edge matches the discontinuity, accurate integration is guaranteed. On the other side, if we do not use curved triangles, the integration is not accurate anymore, and the optimal convergence rate will be lost. When the discontinuity passes through the integration triangle, the accuracy is completely lost. For more systematical curved elements studies with a plenty of numerical examples, we refer to the works of<sup>128,129</sup> for the interested readers.

### 4.3 Numerical Examples

In previous section, we have introduced the XIGA method. In this section, the XIGA method will be applied to solve three material interface problems. There are mainly

### 4.3 Numerical Examples

---

two kinds of discontinuities, i.e the strong and weak discontinuities. The strong discontinuities are jumps in the displacement field; while the weak discontinuities are for problems with a jump in the gradient field. The following discussion is restricted to the weakly discontinuous problems. In XFEM, special enrichment functions are added into the FEM approximation space, which play an important role in describing these discontinuities. The enrichment functions should be continuous themselves, but discontinuous in their derivatives<sup>42,130</sup> for describing weak discontinuities. In the following, we are going to solve the bimaterial interface problem (subsection 4.3.1), the curved material interface problem (subsection 4.3.2), and one more general curved material interface problem (subsection 4.3.3).

#### 4.3.1 Example I (Bimaterial Interface on a Square)

The open set domain  $\Omega$  is chosen by:

$$\Omega := \{ (x,y) \mid 0 < x < 1, 0 < y < 1 \} ,$$

and the boundary  $\partial\Omega := \{ (x,y) \mid x = 0, x = 1, y = 0, y = 1 \}$ .

The body force and the exact solution are given by:

$$f(x,y) = 6x - \frac{9}{5} , \tag{4.25}$$

$$u = \begin{cases} u_1 = 2x^3 - \frac{9}{5}x^2 & \text{if } 0 \leq x \leq \frac{1}{2} \\ u_2 = x^3 - \frac{9}{10}x^2 - \frac{1}{10} & \text{if } \frac{1}{2} \leq x \leq 1 . \end{cases} \tag{4.26}$$

### 4.3 Numerical Examples

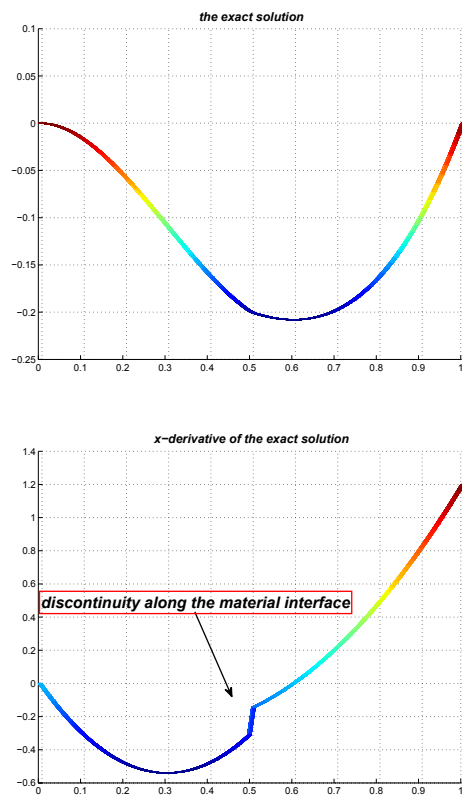


Figure 4.6: The function  $u$  is continuous, but  $\nabla_x u$  is discontinuous along the middle axis.

The problem has homogeneous boundary conditions:  $u = 0$  if  $x = 0$  or  $x = 1$ ;  $\partial u / \partial \mathbf{n} = 0$  if  $y = 0$  or  $y = 1$ . The material properties are given by:

$$\mathbf{K} = \begin{cases} \mathbf{K}_1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} & \text{if } 0 \leq x \leq \frac{1}{2}, 0 \leq y \leq 1 \\ \mathbf{K}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \text{if } \frac{1}{2} \leq x \leq 1, 0 \leq y \leq 1 . \end{cases} \quad (4.27)$$

### 4.3 Numerical Examples

It is easy to check that  $\nabla \cdot (\mathbf{K}_1 \nabla u_1) = f = \nabla \cdot (\mathbf{K}_2 \nabla u_2)$ . From Figure 4.6,  $u$  is continuous function inside  $\Omega$ , including the interface  $x = \frac{1}{2}$ . However,  $\nabla u$  has discontinuity along the interface:

$$\nabla_x u_1\left(\frac{1}{2}\right) = -\frac{3}{10} \neq -\frac{21}{20} = \nabla_x u_2\left(\frac{1}{2}\right) .$$

A comparison of error plots is shown in Figure 4.7. First, Figure 4.7(a) shows the  $x$ -derivative approximation errors from the traditional IGA. Then after applying ramp enrichment function  $\phi_1$  and  $\phi_2$  defined in Sec. 4.2.2.1, the errors that occur in the area of the discontinuity part are reduced significantly as shown in Figure 4.7(b). Another improvement is for the accuracy near the boundaries by using control points located at the Greville abscissae, which is shown in Figure 4.7(c). The last step is to reduce the error in the so-called ‘‘blending elements’’ by repeating middle neighbor knots strategy. It is obvious that the final results in Figure 4.7(d) are much more accurate numerically.

Table 4.1: Energy Norm of the Errors (Ramp Enrichment + Repeating Middle Neighbour Knots + Greville Abscissae)

element mesh	h	DOFs	errors(energy)	convergence rate repect to DOFs	convergence rate repect to h
5 × 5	1/5	105	8.152394645841e-003		
7 × 7	1/7	153	4.628136404195e-003	-1.503827761202	1.682627454730
9 × 9	1/9	209	2.939321629442e-003	-1.455533206509	1.806404308904
11 × 11	1/11	273	2.024635329808e-003	-1.395495412781	1.857716271361
21 × 21	1/21	713	5.876083130807e-004	-1.288616546179	1.913133803627
31 × 31	1/31	1353	2.746882059737e-004	-1.187052858616	1.952484534261
41 × 41	1/41	2193	1.584892270868e-004	-1.138739937679	1.967023645371
51 × 51	1/51	3233	1.029971849244e-004	-1.110385256692	1.974698392950
61 × 61	1/61	4473	7.226087499876e-005	-1.091699073833	1.979460056531

We also collect all the detailed final results of the XIGA in Table 4.1. Inside the table,  $h$  denotes the characteristic length scale and DOFs stand for the degrees of freedom. As it can be seen, after each refinement step, the characteristic length scale

### 4.3 Numerical Examples

decreases and the DOFs are increased. The theoretical optimal convergence rate for energy norm errors is 2. For our XIGA, the convergence rate is 1.979, see the last column in Table 4.1. The convergence rate of numerical solutions matches with the theoretical result. Figure 4.8 compares the energy norm convergence rates from IGA and XIGA with several different techniques. All the comparison results show that the effect of the XIGA is obvious compared to the standard IGA.

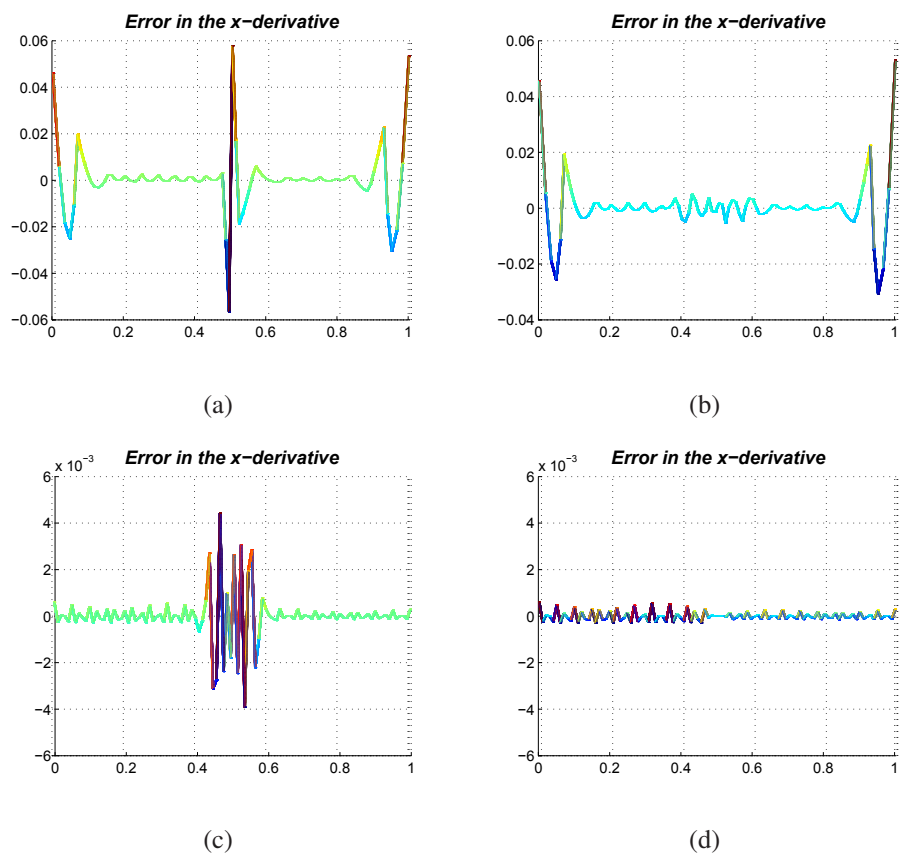


Figure 4.7: Comparison of the  $x$ -derivative approximation between two methods.  $p = 2$  and  $q = 2$ . (a) is the error producing by IGA. (b) is the error producing by XIGA with Ramp enrichment functions  $\phi_1$  and  $\phi_2$ . (c) is considered Ramp enrichment function and Greville Abscissae but without repeating middle neighbor knots. (d) is considered Ramp enrichment function, Greville Abscissae and also repeating middle neighbor knots.

### 4.3 Numerical Examples

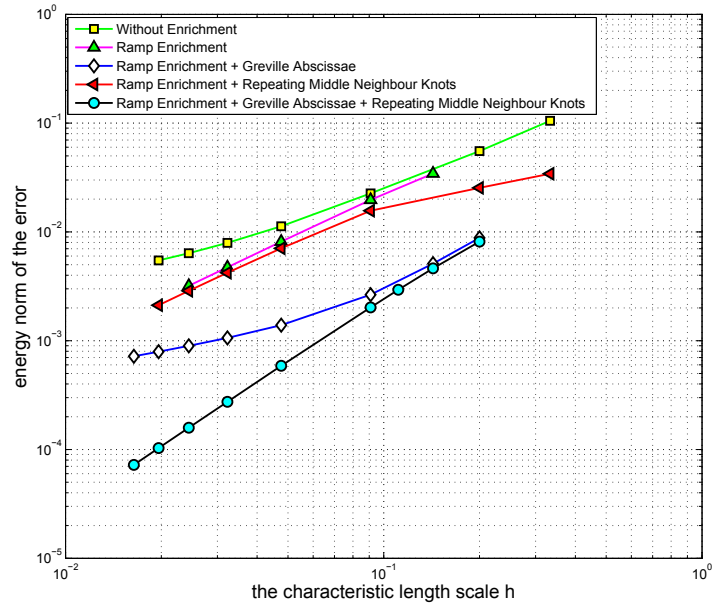


Figure 4.8: Comparison of the convergence rates of the energy norms between IGA and XIGA with different techniques.

#### 4.3.2 Example II (Bimaterial Interface on a Disc)

The open set domain  $\Omega$  is chosen by:

$$\Omega := \{ (x, y) \mid x^2 + y^2 < 1 \} ,$$

and the boundary  $\partial\Omega := \{ (x, y) \mid x^2 + y^2 = 1 \}$ .

The body force and the exact solution are given by:

$$f(x, y) = 16x^2 + 16y^2 - 9 , \tag{4.28}$$



### 4.3 Numerical Examples

---

$$u(x, y) = \begin{cases} u_1 = 2(-x^2 - y^2 + \frac{1}{4})(-x^2 - y^2 + 1) - 2x^2 - 2y^2 + \frac{5}{4} & \text{if } x^2 + y^2 \leq \frac{1}{4} \\ u_2 = (-x^2 - y^2 + \frac{1}{4})(-x^2 - y^2 + 1) - x^2 - y^2 + 1 & \text{if } \frac{1}{4} \leq x^2 + y^2 \leq 1 . \end{cases} \quad (4.29)$$

In the Example II, we consider homogeneous boundary condition  $u|_{\partial\Omega} = 0$ . The material interface is defined by  $I := \{ (x, y) \mid x^2 + y^2 = \frac{1}{4} \}$ , and the material properties are given by:

$$\mathbf{K} = \begin{cases} \mathbf{K}_1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} & \text{if } x^2 + y^2 \leq \frac{1}{4} \\ \mathbf{K}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \text{if } \frac{1}{4} < x^2 + y^2 \leq 1 . \end{cases} \quad (4.30)$$

Inside  $\Omega$ ,  $u(x, y) \in C^0$ , but  $u(x, y) \notin C^1$ , since  $\nabla u_1(x, y)|_I \neq \nabla u_2(x, y)|_I$ . The discontinuity is illustrated in Figure 4.9.

In  $u$  and  $v$  directions, the order of basis functions are quadratic and the initial knots vectors are given by:

$$\begin{aligned} \mathfrak{E} &= \{ 0, 0, 0, 1/4, 1/4, 1/2, 1/2, 3/4, 3/4, 1, 1, 1 \} \\ \mathfrak{H} &= \{ 0, 0, 0, 1, 1, 1 \} \end{aligned}$$

Since the knots vectors are open knots vectors, the end knots are repeated for three times. According to the initial knots vectors, there are 9 basis functions in the  $u$ -direction and 3 basis functions in the  $v$ -direction. Table 4.2 shows the initial control points applied for constructing the circular physical domain.

### 4.3 Numerical Examples

Table 4.2: The Initial Control Points (the third coordinates are weights)

$(0,0,1)$	$(0, -\frac{1}{2}, 1)$	$(0, -1, 1)$
$(0,0, \frac{\sqrt{2}}{2})$	$(\frac{1}{2}, -\frac{1}{2}, \frac{\sqrt{2}}{2})$	$(1, -1, \frac{\sqrt{2}}{2})$
$(0,0,1)$	$(\frac{1}{2}, 0, 1)$	$(1, 0, 1)$
$(0,0, \frac{\sqrt{2}}{2})$	$(\frac{1}{2}, \frac{1}{2}, \frac{\sqrt{2}}{2})$	$(1, 1, \frac{\sqrt{2}}{2})$
$(0,0,1)$	$(0, \frac{1}{2}, 1)$	$(0, 1, 1)$
$(0,0, \frac{\sqrt{2}}{2})$	$(-\frac{1}{2}, \frac{1}{2}, \frac{\sqrt{2}}{2})$	$(-1, 1, \frac{\sqrt{2}}{2})$
$(0,0,1)$	$(-\frac{1}{2}, 0, 1)$	$(-1, 0, 1)$
$(0,0, \frac{\sqrt{2}}{2})$	$(-\frac{1}{2}, -\frac{1}{2}, \frac{\sqrt{2}}{2})$	$(-1, -1, \frac{\sqrt{2}}{2})$
$(0,0,1)$	$(0, -\frac{1}{2}, 1)$	$(0, -1, 1)$

Table 4.3 shows the final numerical results of the XIGA in terms of  $L^2$  norm errors. The  $L^2$  norm errors on the third column are reduced dramatically in each refinement step. In addition, from the  $x$ -derivative errors plot in Figure 4.10, the XIGA is further enhanced by the repeating middle neighbor knots technique, improving the approximation around the material interface. It is also interesting to point out that the  $L^2$  norm numerical convergence rate of our proposed method is 2.979, see the fifth column in Table 4.3, which is very close to the theoretical optimal convergence rate 3. Figure 4.11 compares the convergence rates of the  $L^2$  norm errors between IGA and XIGA, and it is obvious the performance of XIGA outperforms the traditional IGA in modeling discontinuous problems.

Table 4.3:  $L^2$  Norm of the Errors (Moës Enrichment + Repeating Middle Neighbor Knots)

DOFs	$h$	errors( $L^2$ )	convergence rate respect to DOFs	convergence rate respect to $h$
170	0.5305002849268	2.649329880e-003		
656	0.1887891241413	1.219080760e-004	-2.2799767199322	2.9798925443626
3842	0.0637964594038	4.642324369e-006	-1.8488770727607	3.0122157258419
29610	0.0213459577589	1.744186487e-007	-1.6069151209544	2.9972654453068

### 4.3 Numerical Examples

---

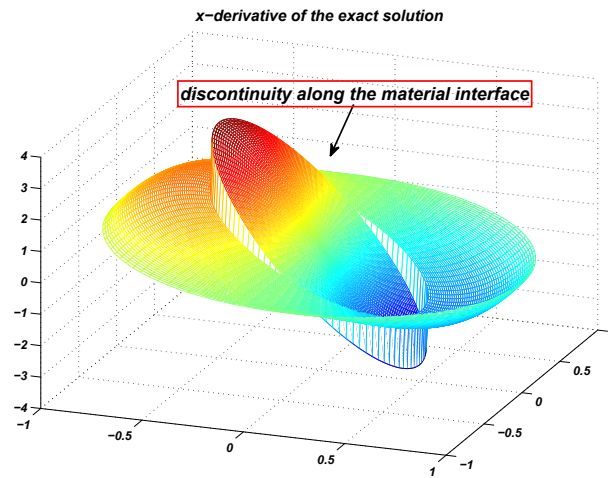


Figure 4.9: the curved material interface model

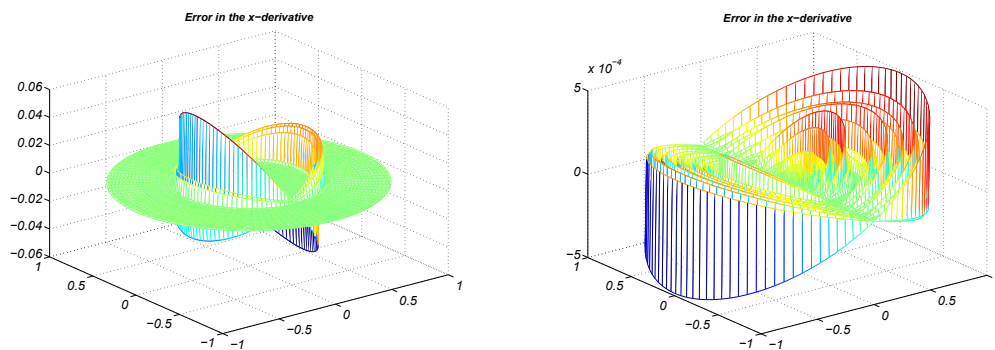


Figure 4.10: Comparison of the  $x$ -derivative approximation errors between the XIGA enriched by Moës enrichment function but without repeating the middle neighbor knots (left) and the XIGA enriched by Moës enrichment function with repeating the middle neighbor knots.

### 4.3 Numerical Examples

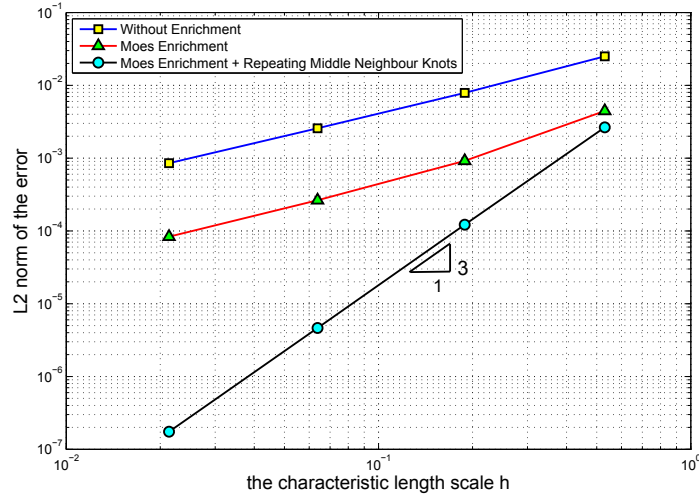


Figure 4.11: Compare the convergence rates of the  $L^2$  norm errors between IGA and XIGA.

#### 4.3.3 Example III (More General Interface Problem on a Disc )

The exact solution is given by:

$$f(x, y) = 16x^2 + 16y^2 - 4 - \frac{1}{16} , \quad (4.31)$$

$$u(x, y) = \begin{cases} u_1 = 2(-x^2 - y^2 + \frac{1}{64})(-x^2 - y^2 + 1) & \text{if } x^2 + y^2 \leq \frac{1}{64} \\ u_2 = (-x^2 - y^2 + \frac{1}{64})(-x^2 - y^2 + 1) & \text{if } \frac{1}{64} \leq x^2 + y^2 \leq 1 . \end{cases} \quad (4.32)$$

In Example III, we consider the same boundary condition as Example II. The material interface is defined by  $I := \{ (x, y) \mid x^2 + y^2 = \frac{1}{64} \}$ , and the material properties

### 4.3 Numerical Examples

are given by:

$$\mathbf{K} = \begin{cases} \mathbf{K}_1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} & \text{if } x^2 + y^2 \leq \frac{1}{64} \\ \mathbf{K}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \text{if } \frac{1}{64} < x^2 + y^2 \leq 1 . \end{cases} \quad (4.33)$$

Figure 4.12 shows the parametric domain and the physical domain. The parametric coordinates of the material interface are constructed by the inverse mapping in subsection 4.2.5 and the curve fitting in subsection 4.2.6.

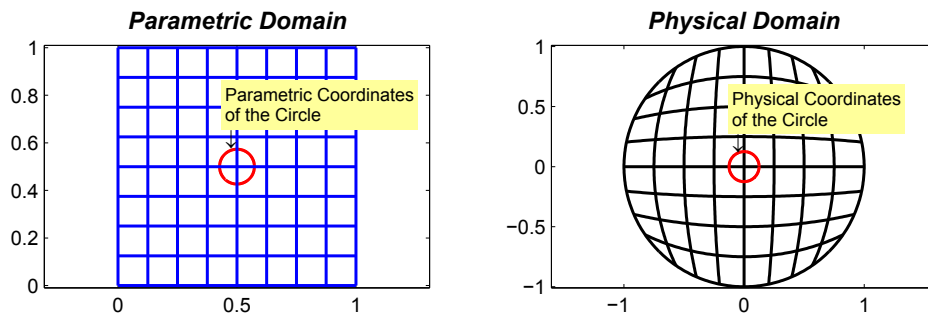


Figure 4.12: The effect of the inverse mapping for a circle

In this Example, we apply the ramp enrichment functions (4.5) and (4.6), and repeat the knots that surround the material interface similar to Example I and Example II. The enriched elements are divided into sub-triangular elements, which contain the curved triangular elements discussed in subsection 4.2.8. The results are shown in Table 4.4, 4.5, 4.6 and 4.7. Figure 4.13 and Figure 4.14 compare the convergence rates obtained by IGA and XIGA. It can be seen that as the mesh is refined, the XIGA becomes much more accurate. In addition, the numerical convergence rate of the XIGA in terms of the energy norm errors reaches to 2.004 (see the last column in Table 4.5),

### 4.3 Numerical Examples

---

which perfectly matches the theoretical optimal convergent rate 2. While, for the traditional IGA, the errors improvement gets slower and slower because of its limitation in modeling discontinuities. Likewise, for  $L^2$  norm errors case, the XIGA also makes an outstanding performance. The optimal convergence rate is clearly observed from the last column of the Table 4.7.

Table 4.4: Energy Norm of the Errors with IGA

h	DOFs	errors (Energy Norm)	convergence rate (Energy Norm)
0.368683	100	0.0646215809	
0.184825	324	0.0288270496	1.169007407172
0.092547	1156	0.0155824155	0.889377446160
0.046308	4356	0.0102319397	0.607493710512
0.023164	16900	0.0070860871	0.530353370040

Table 4.5: Energy Norm of the Errors with XIGA

h	DOFs	errors (Energy Norm)	convergence rate (Energy Norm)
0.368683	219	0.0527724692	
0.184825	579	0.0134102137	1.98394527752073
0.092547	1825	0.0033327138	2.01280210416680
0.046308	6265	0.0008353585	1.99838295205034
0.023164	22921	0.0002084374	2.00404591385051

Table 4.6:  $L^2$  Norm of the Errors with IGA

h	DOFs	error( $L^2$ )	convergence rate $L^2$
0.368683	100	1.49022952e-03	
0.184825	324	5.70841339e-04	1.3896204857
0.092547	1156	1.89682082e-04	1.5928611855
0.046308	4356	7.51777547e-05	1.3366457428
0.023164	16900	2.96590460e-05	1.3426812289

### 4.3 Numerical Examples

---

Table 4.7:  $L^2$  Norm of the Errors with XIGA

h	DOFs	error( $L^2$ )	convergence rate $L^2$
0.368683	219	1.09719621e-03	
0.184825	579	1.33239639e-04	3.0532563778
0.092547	1825	1.66963391e-05	3.0027438134
0.046308	6265	2.08023877e-06	3.0079514666
0.023164	22921	2.58861564e-07	3.0083955677

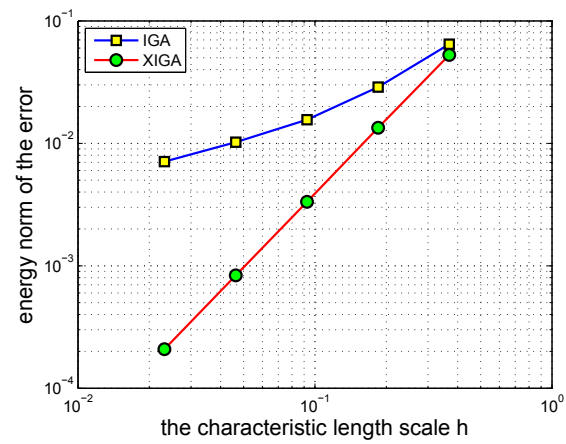


Figure 4.13: Compare the convergence rates of the energy norm of the errors between IGA and XIGA.

## 4.4 Conclusions

---

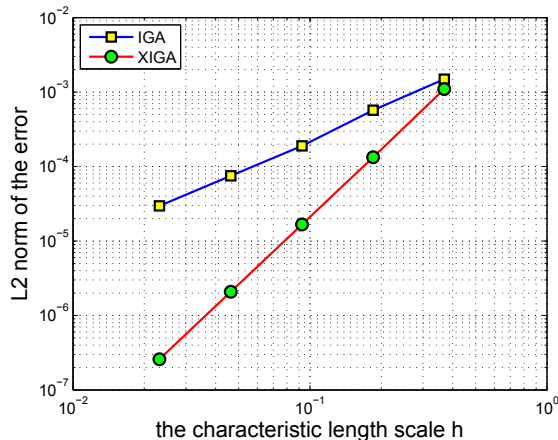


Figure 4.14: Compare the convergence rates of the  $L^2$  norm of the errors between IGA and XIGA.

**Discussion:** In this section, we have numerically studied XIGA, solving three different material interface problems based on the Poisson's equation with weak discontinuities. We have also used IGA as a control method, and compared their results. We found that the two methods have quite different effects in terms of the  $L^2$  norm of the errors, the energy norm of the errors and the numerical convergence rates. This is mainly due to the limitation of IGA in modeling problems with discontinuous solutions. In our proposed method, the precision is significantly improved, and the numerical convergence rates are also optimal. These comparisons verify that the XIGA has the ability to accurately model problems with weakly discontinuous solutions.

## 4.4 Conclusions

We have developed an extended IGA method to solve material interface problems by combining the features of XFEM and IGA. We show that the XIGA can achieve the optimal convergence rate while the IGA can only reach suboptimal convergence for the Poisson's equation with weakly discontinuous solutions. Several important techniques have been applied in our XIGA method. (1) The Moës enrichment function<sup>120</sup>



## 4.4 Conclusions

---

and ramp enrichment functions<sup>119</sup> are used during our approximation. (2) The control points are located at the Greville abscissae so that the mapping between the parametric and physical space is free from mesh size-related problems. (3) Repeating certain knots reduces the span of several basis functions and resolves the accuracy issues in the blending elements encountered in the standard XFEM. (4) For the real curved material interface problem, the curved triangular elements are applied and the material interface is approximated in the parametric space. We also applied our XIGA method to study the bimaterial, curved, and real curved material interface problems, and optimal convergence rates are obtained for all of the three examples.

As part of our future work, we plan to apply our algorithm to more complex material interface and fracture problems. In addition, we would also like to extend our technique to three dimensions.

# Chapter 5

## An Isogeometric Collocation Method Using Superconvergent Points

### 5.1 Introduction

We develop an IGA collocation method modified by collocating at points other than the standard Greville abscissae. The method is related to orthogonal collocation used for solving differential equations and to the superconvergence theory, therefore we refer to this method as “super-collocation” (IGA-SC). By carefully choosing the collocation points, it can be seen that the IGA-SC converges in the first derivative (energy) norms at rates similar to that of the Galerkin solution. This is different from the collocation at Greville abscissae (IGA-C), where the convergence in energy norm for odd polynomial degrees is typically suboptimal. The method is tested on 1D, 2D and 3D numerical examples, in which it is compared to IGA-C and Galerkin’s method (IGA-G). The comparison includes a detailed cost vs. accuracy analysis, which shows an improved efficiency of the proposed method in particular for odd polynomial degrees.

## 5.2 IGA Super Collocation Method

### 5.2.1 Collocation

We assume that the strong form of the boundary value problem is given by

$$\mathcal{L}u = f \quad \text{in } \Omega, \quad (5.1)$$

$$\mathcal{G}u = g \quad \text{on } \partial\Omega. \quad (5.2)$$

Here  $\Omega$  is an open set with Lipschitz continuous boundary  $\partial\Omega$ ,  $u$  is the unknown solution,  $\mathcal{L}$  is a differential operator,  $\mathcal{G}$  is a vector operator and  $f, g$  are given functions. As an example, one can take  $\mathcal{L}(u) := -\nabla \cdot \nabla u + au$ ,  $a \geq 0$ .

To solve this problem by the collocation method, two sets of sample (collocation) points need to be considered, a set  $\{x_i^{int}\}$ ,  $i = 1, \dots, N^{int}$  in the interior of  $\Omega$  and a set  $\{x_i^{bnd}\}$ ,  $i = 1, \dots, N^{bnd}$  on  $\partial\Omega$ , where  $N^{int}$  and  $N^{bnd}$  are the number of collocation points chosen in the interior and on the boundary, respectively. Then the collocation solution  $u_h^C$ , is required to satisfy:

$$\begin{aligned} \mathcal{L}u_h^C(x_i^{int}) &= f(x_i^{int}) \quad \text{in } \Omega, \quad i = 1, \dots, N^{int} \\ \mathcal{G}u_h^C(x_i^{bnd}) &= g(x_i^{bnd}) \quad \text{on } \partial\Omega, \quad i = 1, \dots, N^{bnd}. \end{aligned}$$

The traditional IGA collocation uses the so-called ‘‘Greville abscissae’’ to define the collocation points. For a given knot vector  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ , the associated Greville abscissae points  $\bar{\xi}_i$ ,  $i = 1, \dots, n$ , are calculated by

$$\bar{\xi}_i = \frac{\xi_{i+1} + \xi_{i+2} + \dots + \xi_{i+p}}{p}. \quad (5.3)$$

The collocation points  $\tau_{ijk} \in \Omega$  are defined by the tensor product structure:

$$\tau_{ijk} = \mathbf{F}(\hat{\tau}_{ijk}), \quad \hat{\tau}_{ijk} = (\bar{\xi}_i, \bar{\eta}_j, \bar{\zeta}_k) \in \hat{\Omega}, \quad (5.4)$$

## 5.2 IGA Super Collocation Method

---

for  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ , and  $k = 1, \dots, l$ .  $\mathbf{F}$  is the parametric mapping from (2.9).  $\hat{\tau}_{ijk}$  are the parametric coordinates of the collocation points, and  $\tau_{ijk}$  represent their corresponding physical coordinates. We note that the  $\hat{\tau}_{ijk}$  (and also  $\tau_{ijk}$ ) need to be separated into interior and boundary collocation points. When  $u_h^C$  is a linear combination of the B-spline (or NURBS) basis functions, this gives rise to a non-symmetric linear system. We mention that there are other ways of choosing the collocation points, such as at the so-called Demko's abscissae. These have been shown, for example in<sup>37</sup>, to lead to more stable collocation schemes, but are more difficult to compute and only result in very slight differences in accuracy compared to Greville abscissae for most problems of interest. In the next subsection, we will propose another choice of collocation points which is easy to implement and leads to considerably better approximation properties.

In our numerical examples, the linear elasticity problem is evaluated by the  $L^2$  and energy norms. In others, we use the  $L^\infty$  norm,  $W^{1,\infty}$  and  $W^{2,\infty}$  seminorms, and also the  $L^2$  norm,  $H^1$  and  $H^2$  seminorms, for which analogous results have been observed. The equations of these norms are referred to the Section 2.3 of Chapter 2. We note that collocation in general converges optimally in the 2<sup>nd</sup> derivative norms ( $W^{2,\infty}$  and  $H^2$ ), but not always so for 0<sup>th</sup> and 1<sup>st</sup> derivative norms. The latter in particular are of interest in engineering applications.

### 5.2.2 Superconvergent Points for Collocation

In this subsection, we discuss a choice of collocation points for which the collocation solution behaves similarly to the standard Galerkin approximation. The main idea of this method is that we seek the collocation points for which the error in the second derivatives of the Galerkin approximation is also small. This is based on the observation that the error in the second derivative for the collocation approximation is smallest at the collocation points.

The phenomenon of superconvergence is based on the fact that there are often points  $x^*$  in the domain  $\Omega$  for which the computed solution  $u_h$  is more accurate at  $x^*$

## 5.2 IGA Super Collocation Method

---

compared to other points in  $\Omega$ . These points are, of course, only useful if they can be calculated based on the mesh discretization and do not depend on the unknown solution  $u$ . For example, in the standard 1D finite elements it is well known that the approximation  $u_h$  is exact at the mesh nodes. It is also known for linear finite elements, under some regularity assumptions, that the derivative of the approximate solution is more accurate at the centroid of the elements. This has been used for error estimation purposes, such as in ZZ-patch recovery techniques<sup>131</sup>.

Here we will look for superconvergence points in the second derivative, not for error estimation purposes, but to investigate their use in collocation. We consider an approximation space  $S^h(\Omega)$  which reproduces polynomials of degree  $p$ , and an exact solution  $u$ . Then, the approximation properties of the computed solution  $u_h$  around a point  $x_0$  in the domain can be investigated by considering the Taylor expansion of degree  $p + 1$  at  $x_0$ .

Specifically, suppose that the Galerkin formulation of the problem (5.1) can be written as:

$$\text{Find } u_h \in S^h(\Omega) \text{ such that } B(u - u_h, \chi) = 0, \forall \chi \in S^h(\Omega),$$

where  $B(\cdot, \cdot)$  is the bilinear form corresponding to the weak formulation of the problem. Let  $\Omega_0$  be a subdomain of  $\Omega$  containing a point  $x_0$ , let  $Q$  be the  $p + 1$  degree Taylor expansion of  $u$  centered at  $x_0$ , and let  $Q_h$  be the projection  $B(Q - Q_h, \chi) = 0, \forall \chi \in S^h(\Omega_0)$  with  $\int_{\Omega_0} (Q - Q_h) = 0$ . Then we can write, by using the triangle inequality:

$$\|u - u_h\|_{W^{i,\infty}(\Omega_0)} \leq \|Q - Q_h\|_{W^{i,\infty}(\Omega_0)} + \|(u - Q) - (u_h - Q_h)\|_{W^{i,\infty}(\Omega_0)}, \quad i = 0, 1, 2.$$

It has been shown for the  $p$ -version of the Finite Elements<sup>132</sup> and also for Generalized Finite Elements<sup>133</sup>, that for  $i = 1$  and an interior subset  $\Omega_1 \subset \Omega_0$ , the approximation is dominated by the first term on  $\Omega_1$ , i.e.

$$\|u - u_h\|_{W^{i,\infty}(\Omega_1)} \leq \|Q - Q_h\|_{W^{i,\infty}(\Omega_1)} + O(h^{p-i+1+\alpha}), \text{ with } \alpha > 0. \quad (5.5)$$

## 5.2 IGA Super Collocation Method

---

Moreover, for an uniform discretization (the shape functions in  $\Omega_0$  are translation invariant and the mesh is uniform), it has been shown that  $Q - Q_h$  can be approximated by a periodic function related to a suitably chosen interpolant  $I^h[Q]$  of  $Q$  in  $S^h(\Omega_0)$ :

$$\|Q - Q_h\|_{W^{i,\infty}(\Omega_1)} \leq \|(Q - I^h[Q]) - P_{per}(Q - I^h[Q])\|_{W^{i,\infty}(\Omega_1)} + O(h^{p-i+1+\alpha}),$$

where  $P_{per}(Q - I^h[Q])$  satisfies  $B(P_{per}(Q - I^h[Q]), \chi) = B(Q - I^h[Q], \chi), \forall \chi \in S_h^{per}(\Omega_0)$ , and  $S_h^{per}(\Omega_0)$  is a subset of  $S^h(\Omega_0)$  that is periodic with period  $\bar{h}$  related to the mesh size parameter  $h$ . Then by the linearity of the interpolation and projection operators, to find the superconvergent points it suffices to find the roots of

$$\psi := (Q - I^h[Q]) - P_{per}(Q - I^h[Q]) \quad (5.6)$$

and its derivatives, where  $Q$  is any monomial of degree  $p + 1$ . Because  $\psi$  is periodic, this requires only finitely many operations, i.e. this can be solved on a reference element independently of  $h$  and the results can be scaled and translated to any particular mesh. Moreover, if  $S^h$  is a space of piecewise polynomials, then  $\psi$  is also a piecewise polynomial of degree  $p + 1$ . The interior estimate (5.5) has not been proven for B-splines, however we can follow the steps similar to finite elements to calculate  $\psi$ .

For example to find the superconvergent points in the parametric space, on the reference interval  $[-1, 1]$  for a B-spline basis of degree  $p$ , the procedure is:

1. Consider a non-open knotvector  $[-2p + 1, -2p + 3, \dots, -1, 1, \dots, 2p - 3, 2p - 1]$ , where the knots are uniform with nodal distance 2 and calculate the associated B-spline basis of degree  $p$ .
2. Construct an interpolant  $I^h[Q]$  for  $Q = x^{p+1}$  and the subspace  $S_h^{per}$  of B-splines with period 2.
3. Calculate  $\psi$  according to (5.6).
4. Calculate the roots of  $\psi$ ,  $\psi'$  or  $\psi''$ . We present the superconvergent points for B-splines with polynomial degree  $p = 1, \dots, 7$  in Table 5.1.

## 5.2 IGA Super Collocation Method

Table 5.1: The superconvergent points for B-spline basis of degree  $p$  on interval  $[-1,1]$

	Roots of $\psi$	Roots of $\psi'$	Roots of $\psi''$
$p = 1$	$-1, 1$	$0$	$-$
$p = 2$	$-1, 0, 1$	$\pm 1/\sqrt{3}$	$0$
$p = 3$	$\pm\sqrt{225 - 30\sqrt{30}}/15$	$-1, 0, 1$	$\pm 1/\sqrt{3}$
$p = 4$	$-1, 0, 1$	$\pm\sqrt{225 - 30\sqrt{30}}/15$	$-1, 0, 1$
$p = 5$	$\pm 0.5049185675126533$	$-1, 0, 1$	$\pm\sqrt{225 - 30\sqrt{30}}/15$
$p = 6$	$-1, 0, 1$	$\pm 0.5049185675126533$	$-1, 0, 1$
$p = 7$	$\pm 0.503221894597504$	$-1, 0, 1$	$\pm 0.5049185675126533$

Somewhat surprisingly, the same points are also obtained if each knot is repeated twice, resulting in a different basis of reduced continuity. Of interest for our method are the superconvergent points for the second derivative (i.e. the roots of  $\psi''$ ) as we will collocate at these points. For higher dimensions, we take the tensor product of the coordinates of the points for one dimension. Due to superconvergence, the error in the Galerkin approximation at these points is also smallest in the second derivative (i.e. up to  $O(h^p)$ , instead of the overall rate of  $O(h^{p-1})$ ). Therefore with this choice, the Galerkin and collocation schemes agree with each other up to at least terms of order  $O(h^p)$ .

We note that when  $p = 2$ , the collocation points in the interior of  $\Omega$  are the same as the Greville abscissae, which are just the knot averages. For  $p$  even,  $p > 2$  and uniform meshes, the Greville abscissae in the interior are also located at the midpoint of each knotspan. However, this method also suggests collocating at the knot coordinates. For  $p$  odd and uniform meshes, the Greville abscissae are located at the knots (mesh nodes in the parametric space), while the superconvergent points are in the interior of each knot-span.

**Remark 2.3.1:** We observe that for all values of  $p > 2$  there are  $2^d$  superconvergent

## 5.2 IGA Super Collocation Method

---

points per knotspan, where  $d$  is the dimension of the domain. When the smoothest B-spline basis is used, this collocation method gives rise to an over-determined system of equations. A way to avoid this is to double the knots in each direction, which however results in reduced continuity for the basis of  $C^{p-2}$  instead of  $C^{p-1}$  and an increase in the number of degrees of freedom. For  $p = 3$ , this method becomes the same as the orthogonal collocation studied in <sup>90,91</sup>, since the superconvergent points in this case are the Gauss-Legendre point and each knot is doubled. In the numerical examples below we choose instead to keep the  $C^{p-1}$  basis and solve the resulting linear system in a least-squares sense. The least squares solution can be written as

$$\mathbf{x} = (\mathbf{K}^T \mathbf{K})^{-1} (\mathbf{K}^T \mathbf{F}). \quad (5.7)$$

We note that  $\mathbf{K}^T$ ,  $\mathbf{K}$ , and most importantly  $\mathbf{K}^T \mathbf{K}$  are sparse matrices. There are also iterative methods for solving over-determined linear systems <sup>134,135</sup>.

### **Remark 2.3.2:** *Imposing boundary conditions*

It is important that the boundary conditions are imposed exactly and not in a least-squares sense. This was also done in <sup>136</sup> for an overdetermined collocation method. For the homogeneous Dirichlet boundary, this can be accomplished by eliminating the corresponding columns from the matrix system, as shown below:

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & a_{N3} & a_{N4} & \cdots & a_{Nn} \end{pmatrix}}_{N \times n} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{pmatrix}}_{n \times 1} = \underbrace{\begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \end{pmatrix}}_{N \times 1}, \quad (5.8)$$

Here  $N$  and  $n$  are equal to the number of the collocation points and the number of global basis functions respectively, and we assume there are  $b$  prescribed coefficients



### 5.3 Numerical Examples

---

for the boundary basis functions which correspond to the boxed columns in the matrix. The prescribed components of the solution are also boxed. The boxed columns and the known coefficients of the solution can be eliminated from the linear system. Specifically, we solve

$$\underbrace{\begin{pmatrix} a_{11} & a_{13} & a_{15} & \cdots & a_{1n} \\ a_{21} & a_{23} & a_{25} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N3} & a_{N5} & \cdots & a_{Nn} \end{pmatrix}}_{N \times (n-b)} \underbrace{\begin{pmatrix} x_1 \\ x_3 \\ x_5 \\ \vdots \\ x_n \end{pmatrix}}_{(n-b) \times 1} = \underbrace{\begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \end{pmatrix}}_{N \times 1} - \underbrace{\begin{pmatrix} a_{12} & a_{14} & \cdots \\ a_{22} & a_{24} & \cdots \\ \cdots & \cdots & \cdots \\ a_{N2} & a_{N4} & \cdots \end{pmatrix}}_{N \times b} \underbrace{\begin{pmatrix} x_2 \\ x_4 \\ \vdots \end{pmatrix}}_{b \times 1} \quad (5.9)$$

and the final solution is:  $\left( x_1 \boxed{x_2} x_3 \boxed{x_4} \cdots x_n \right)$ .

For Neumann boundary conditions, which are commonly considered in many problems, there are only a few investigations within the framework of collocation. A natural approach, proposed in<sup>38</sup> is called the “pure collocation” approach. In a recent study<sup>137</sup>, it has been shown that the pure collocation approach does not give accurate results in certain situations. Two alternative methods are proposed to improve the imposition of Neumann boundary conditions with good accuracy of the results: hybrid collocation and the enhanced collocation, respectively. Regardless of the method used, the boundary conditions give rise to additional constraints on the linear system of equations which should be satisfied exactly, i.e. not in the least-squares sense.

### 5.3 Numerical Examples

In this section, we present several numerical examples, ranging from one dimensional to three dimensional problems for which an analytic solution is available. We show the error in various norms for IGA-C, IGA-G and the proposed IGA-SC. The focus of the analysis is on the convergence rates obtained by each method.

## 5.3 Numerical Examples

---

### 5.3.1 1D Problem

We begin with a 1D problem, similar to the one studied in<sup>37</sup>. The strong form of the differential equation is given by:

$$\begin{cases} -u''(x) + u'(x) + u(x) = (1 + 4\pi^2) \sin(2\pi x) + 2\pi \cos(2\pi x), \forall x \in (0, 1) \\ u(0) = u(1) = 0, \end{cases} \quad (5.10)$$

and the corresponding exact solution is:

$$u(x) = \sin(2\pi x). \quad (5.11)$$

We consider a linear parametrization of the domain and we apply the three numerical methods to solve this problem, and the results are compared in Figure 5.1 for the  $L^\infty$ -norm, Figures 5.2 and 5.3 for  $W^{1,\infty}$  and  $W^{2,\infty}$  semi-norms from  $p = 2$  up to  $p = 7$  where  $p$  represents the polynomial degree of the B-spline basis. In the  $L^\infty$ -norm comparison through Figures 5.1(a) and 5.1(b), the errors corresponding to IGA-SC are smaller than those of IGA-G. For  $p = 3, 5$  and  $7$ , IGA-SC obtains the optimal convergence rates shown in Figure 5.1(b) in the sense that they are the same as the ones for the IGA-G (see Figure 5.1(c)). The convergence rates are calculated with respect to  $h$  which is the size of the elements in the mesh. We mention that for  $p = 2$ , the IGA-SC is the same as the IGA-C, since the superconvergent points are the same as the Greville abscissae. For  $p = 4$  and  $p = 6$ , IGA-SC is still not optimal in the  $L^\infty$ -norm as compared to IGA-G, however the error is generally much smaller than IGA-C, especially as the mesh is refined. It can be seen that IGA-SC performs better than IGA-C and for  $p = 3, 5$  and  $7$  the IGA-SC solutions approach asymptotically the IGA-G results.

Figure 5.2 also displays the errors in  $W^{1,\infty}$ -seminorm obtained by the three methods. Here, for  $p = 2, 4$  and  $6$ , the IGA-C and IGA-SC are optimal (same as IGA-G, see Figure 5.2 (a) and (b)). But the errors of IGA-SC are smaller than those of IGA-C when  $p = 4$ , however for  $p = 6$  this only holds for relatively fine meshes. For  $p = 3, 5$

### 5.3 Numerical Examples

and 7, the IGA-SC results are very close to the IGA-G results which are one order of convergence better than IGA-C. For  $W^{2,\infty}$ -seminorm, all 3 methods have the expected convergence rate of  $O(h^{p-1})$ .

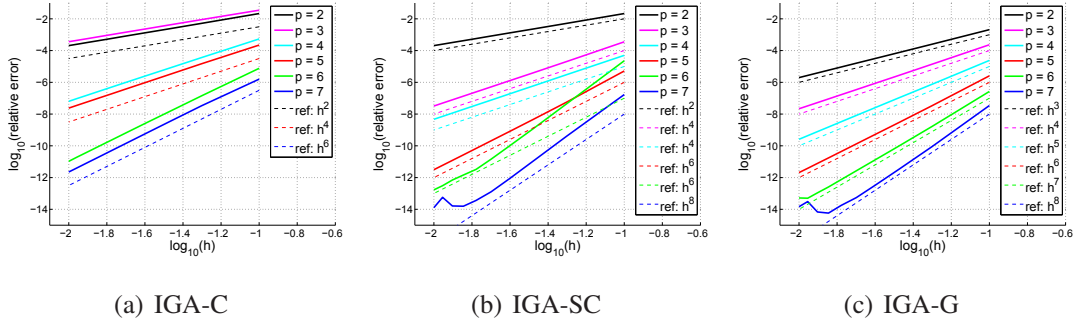


Figure 5.1: The 1D problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.  $\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{L^\infty} / \|u_{ex}\|_{L^\infty})$

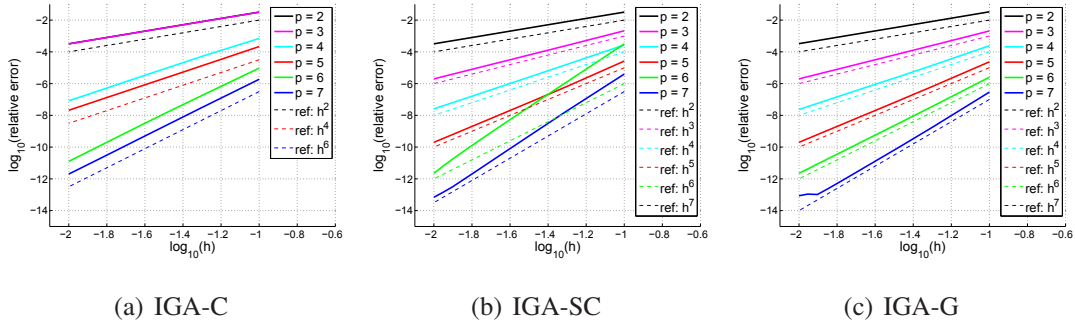


Figure 5.2: The 1D problem solved by (a) IGA-C, (b) IGA-SC, and (c) IGA-G.  $\log_{10}(\text{relative error}) = \log_{10}(|u_{ex} - u_h|_{W^{1,\infty}} / |u_{ex}|_{W^{1,\infty}})$

## 5.3 Numerical Examples

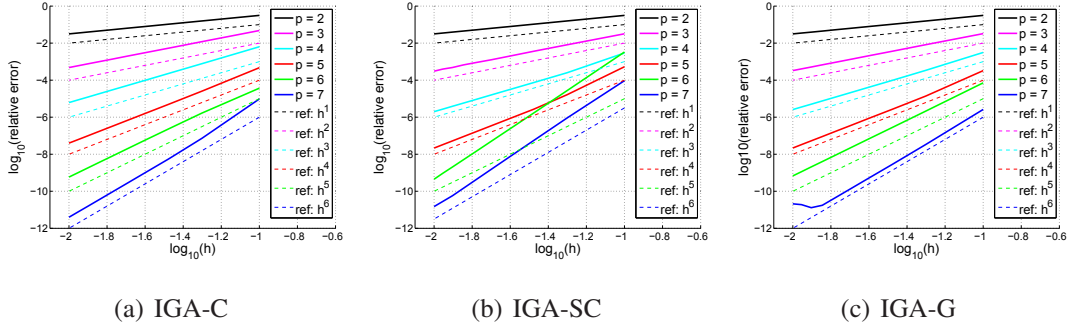


Figure 5.3: The 1D problem solved by (a) IGA-C, (b) IGA-SC, and (c) IGA-G.  
 $\log_{10}(\text{relative error}) = \log_{10}(|u_{ex} - u_h|_{W^{2,\infty}} / |u_{ex}|_{W^{2,\infty}})$

### 5.3.2 Annulus Problem

The second example is the 2D annulus problem whose domain is shown in Figure 5.4. The strong form of the partial differential equation is:

$$\begin{cases} -\Delta u + u = f, & \forall x \in \Omega \\ u|_{\partial\Omega} = 0, \end{cases} \quad (5.12)$$

with

$$\begin{aligned} f(x,y) &= (3x^4 - 67x^2 - 67y^2 + 3y^4 + 6x^2y^2 + 116) \sin(x) \sin(y) \\ &+ (68x - 8x^3 - 8xy^2) \cos(x) \sin(y) \\ &+ (68y - 8y^3 - 8yx^2) \cos(y) \sin(x) \end{aligned} \quad (5.13)$$

which is chosen so that

$$u(x,y) = (x^2 + y^2 - 1)(x^2 + y^2 - 16) \sin(x) \sin(y). \quad (5.14)$$

### 5.3 Numerical Examples

---

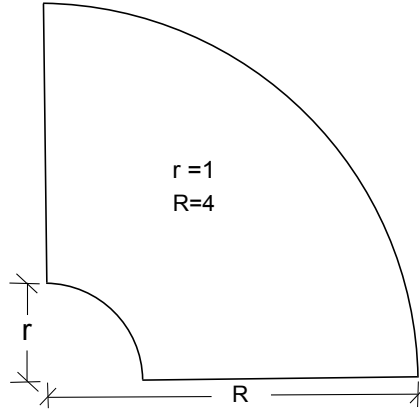


Figure 5.4: Domain  $\Omega$  for the quarter of an annulus problem

This model is used as a “patch test” in many IGA studies, including<sup>37</sup>. As in the previous example, collocation points of the classical IGA-C are given by the Greville abscissae as defined in (5.4). Figure 5.5 (a-b) shows the distribution of the IGA-C collocation points. The coordinates of the collocation points for the IGA-SC are the tensor product of the 1D superconvergent points (in the parameter space). Figure 5.5 (c-d) and (e-f) show the IGA-SC collocation points. We distinguish the odd and the even cases for  $p = 4$  (c-d) and  $p = 5$  (e-f). For even degree, the collocation points include the middle points between each two neighbor knots and also the interior knots. For odd degree, the positions of the super collocation points are in the interior of each knot span.

For this problem, we also present the  $L^\infty$ -norm,  $W^{1,\infty}$ -seminorm and  $W^{2,\infty}$ -seminorm errors of the solution obtained by the three methods. The detailed results are shown in Figures 6.1, 5.7 and 6.3. For the  $L^\infty$ -norm, it can be seen that for  $p = 3, 5$  and  $7$ , IGA-SC has the same convergence rate as IGA-G, but one power of  $h$  slower convergence for  $p = 4$  and  $p = 6$ . For these polynomial degrees of the NURBS basis, IGA-C and IGA-SC have the same convergence rate, but the error in IGA-SC is typically much smaller, by an order of magnitude or more when the mesh is sufficiently refined.

For  $W^{1,\infty}$ -seminorm, IGA-SC has the same convergence rate as the Galerkin method

### 5.3 Numerical Examples

---

for all  $p$ . This is different from IGA-C, which is suboptimal for odd  $p$ . In fact, it can be seen that for  $p = 3, 5$  and  $7$ , the IGA-SC solution converges again asymptotically to the one for IGA-G. However, for  $p = 4$  and  $p = 6$ , the difference between the three methods is much smaller. This indicates that IGA-C is also optimal in the first derivative norms for even polynomial degrees. For the  $W^{2,\infty}$  all 3 methods have the expected optimal convergence rate.

### 5.3 Numerical Examples

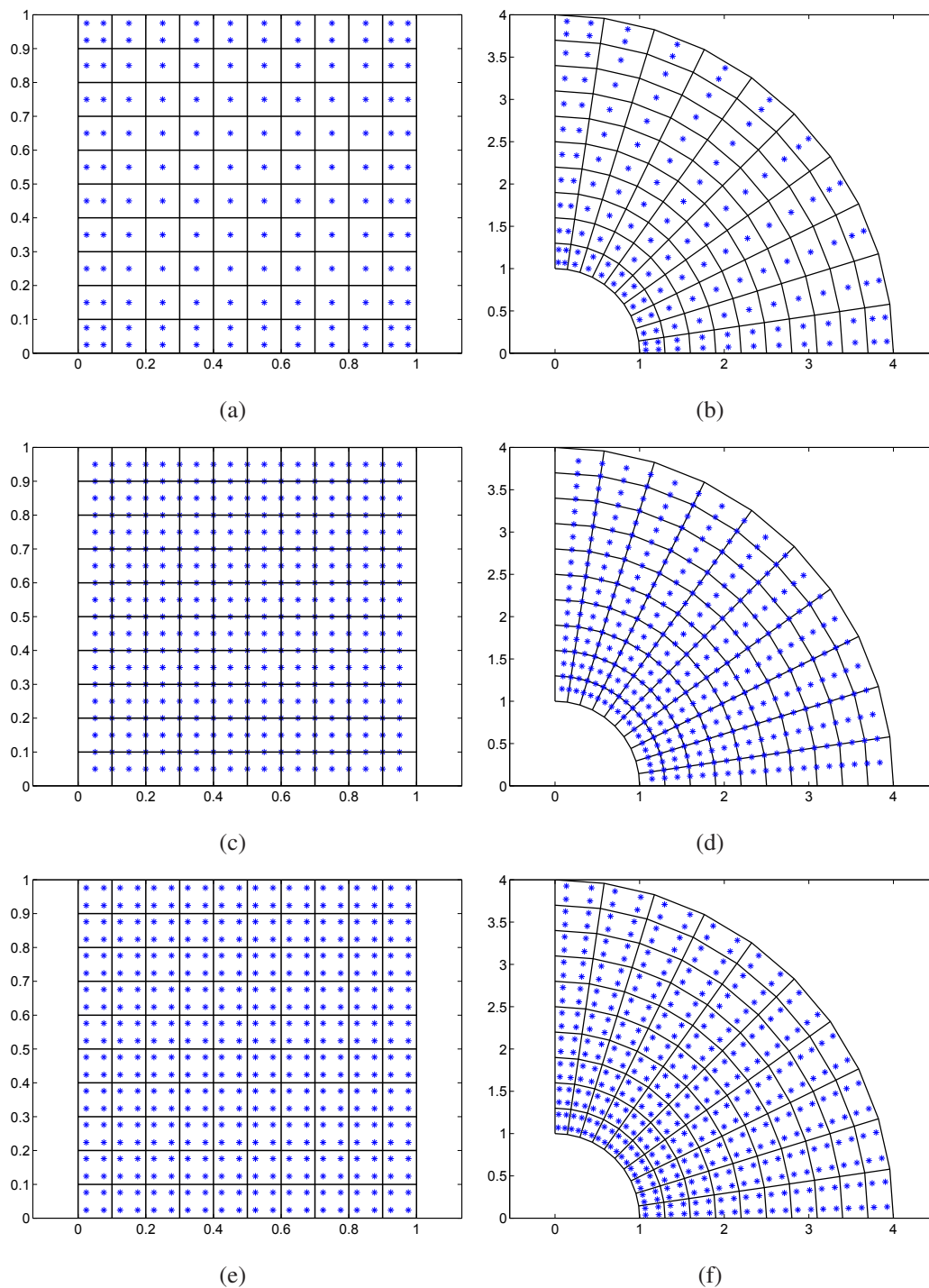


Figure 5.5: Greville abscissae points in (a) parametric domain and (b) physical domain for  $p = 4$ ; (c-d) IGA-SC collocation points in the parametric domain and the physical domain for  $p = 4$  and (e-f)  $p = 5$ . The black lines are the knot (mesh) lines and the blue asterisks represent the collocation points.

### 5.3 Numerical Examples

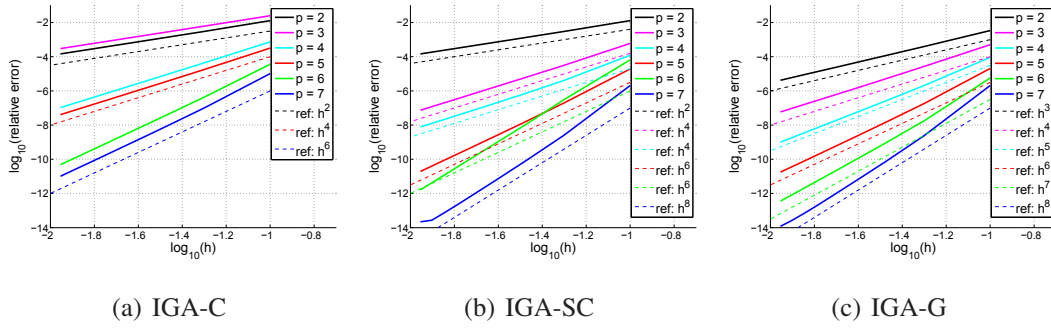


Figure 5.6: The quarter annulus problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.  $\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{L^\infty} / \|u_{ex}\|_{L^\infty})$

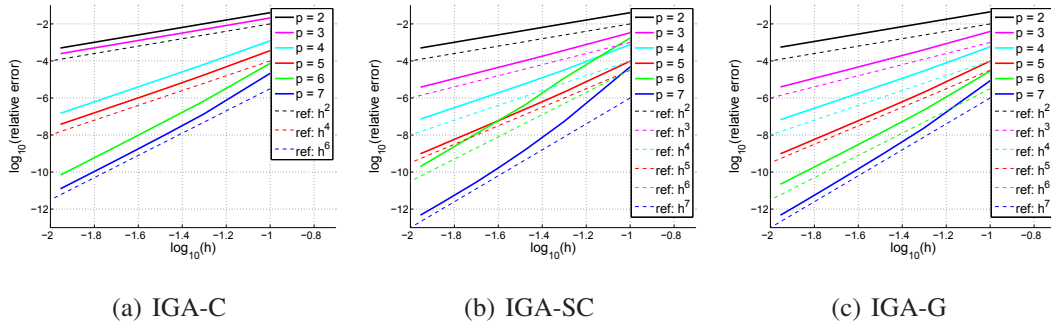


Figure 5.7: The quarter annulus problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.  $\log_{10}(\text{relative error}) = \log_{10}(|u_{ex} - u_h|_{W^{1,\infty}} / |u_{ex}|_{W^{1,\infty}})$

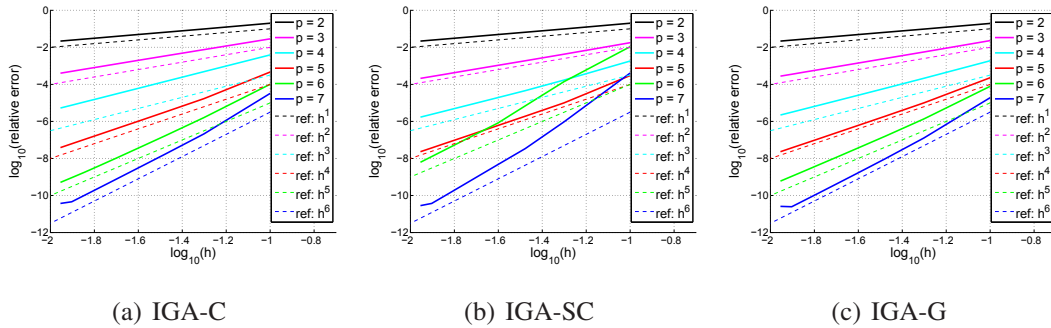


Figure 5.8: The quarter annulus problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.  $\log_{10}(\text{relative error}) = \log_{10}(|u_{ex} - u_h|_{W^{2,\infty}} / |u_{ex}|_{W^{2,\infty}})$



### 5.3.3 Infinite Plate with a Circular Hole with prescribed displacements

We use the three methods to solve a standard 2D linear elasticity benchmark problem. An infinite plate with a hole of radius  $R$  is subjected to a traction  $T_x$  in the  $x$ -direction. The exact solution<sup>138</sup> is given in terms of polar coordinates by:

$$\sigma_{rr}(r, \theta) = \frac{T_x}{2} \left(1 - \frac{R^2}{r^2}\right) + \frac{T_x}{2} \left(1 - 4\frac{R^2}{r^2} + 3\frac{R^4}{r^4}\right) \cos(2\theta), \quad (5.15)$$

$$\sigma_{\theta\theta}(r, \theta) = \frac{T_x}{2} \left(1 + \frac{R^2}{r^2}\right) - \frac{T_x}{2} \left(1 + 3\frac{R^4}{r^4}\right) \cos(2\theta), \quad (5.16)$$

$$\sigma_{r\theta}(r, \theta) = -\frac{T_x}{2} \left(1 + 2\frac{R^2}{r^2} - 3\frac{R^4}{r^4}\right) \sin(2\theta), \quad (5.17)$$

where  $r$  and  $\theta$  are radius and angle with respect to the origin which is located in the center of the hole. The stresses in the Cartesian coordinate system are given by

$$\begin{pmatrix} \sigma_{xx}(x, y) \\ \sigma_{yy}(x, y) \\ \sigma_{xy}(x, y) \end{pmatrix} = A^{-1} \begin{pmatrix} \sigma_{rr}(r, \theta) \\ \sigma_{\theta\theta}(r, \theta) \\ \sigma_{r\theta}(r, \theta) \end{pmatrix}, \quad (5.18)$$

where the transformation matrix  $A$  is:

$$A = \begin{pmatrix} \cos^2(\theta) & \sin^2(\theta) & 2\sin(\theta)\cos(\theta) \\ \sin^2(\theta) & \cos^2(\theta) & -2\sin(\theta)\cos(\theta) \\ -\sin(\theta)\cos(\theta) & \sin(\theta)\cos(\theta) & \cos^2(\theta) - \sin^2(\theta) \end{pmatrix}. \quad (5.19)$$

To solve the problem numerically, only a finite domain is considered, i.e. a square with side  $L = 4$  with one corner at the center of the hole. Exact displacements are imposed on the boundaries of this finite domain. The discretization on the coarsest mesh is the same as in<sup>2</sup>, using a repeated control point in the upper left corner. This parametrization will give rise to a singularity in the inverse mapping between the parametric space and the physical space. However, it has been observed that as long

### 5.3 Numerical Examples

---

as the Gauss points are chosen away from the singularity (which is at the corner of an element), the standard Galerkin method still gives good results. The analytic solution and the errors obtained in the  $x$ -direction stress ( $\sigma_{xx}$ ) are shown in Figure 5.9. We can see that for  $p = 5$  and a NURBS mesh with  $16 \times 32$  elements, the IGA-G and IGA-SC are relatively close, while the IGA-C solution is less accurate. For this problem, we calculate the errors in  $L^2$  norm and energy ( $H^1$ ) norm. The results are shown in Figures 5.10 and 5.11. Again, for odd degrees,  $p = 3$ ,  $p = 5$  and  $p = 7$  the IGA-SC errors are close to IGA-G and smaller than the ones obtained by IGA-C.

### 5.3 Numerical Examples

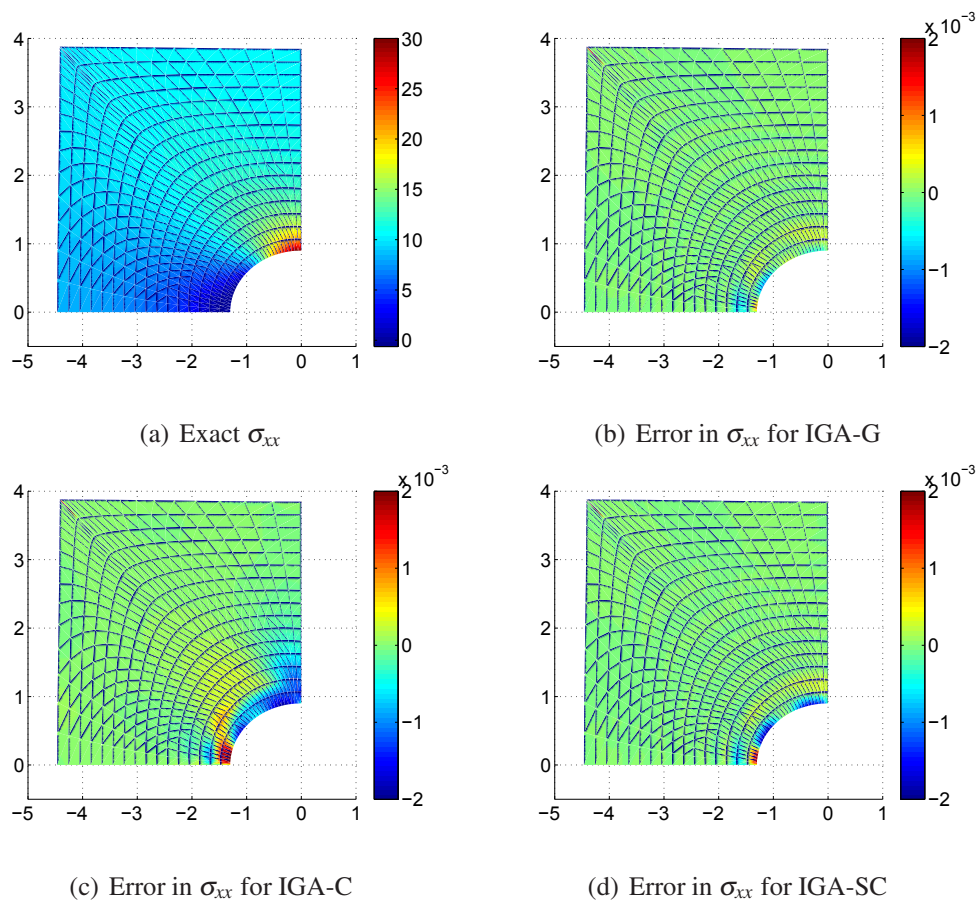


Figure 5.9:  $\sigma_{xx}$  plots of the 2D elastic plate with a circular hole (a) analytic solution, (b) Error of the IGA-G, (c) Error of the IGA-C and (d) Error of the IGA-SC . The applied force is  $T_x = 10$ . The polynomial degree of the basis is  $p = 5$  and there are  $16 \times 32$  elements ( $h = 1/16$  in the parametric space).

### 5.3 Numerical Examples

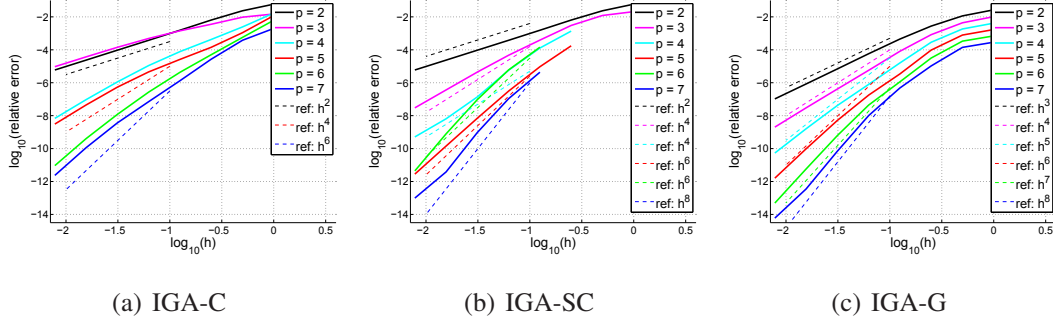


Figure 5.10: The elastic plate with a circular hole problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.  $\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{L^2} / \|u_{ex}\|_{L^2})$

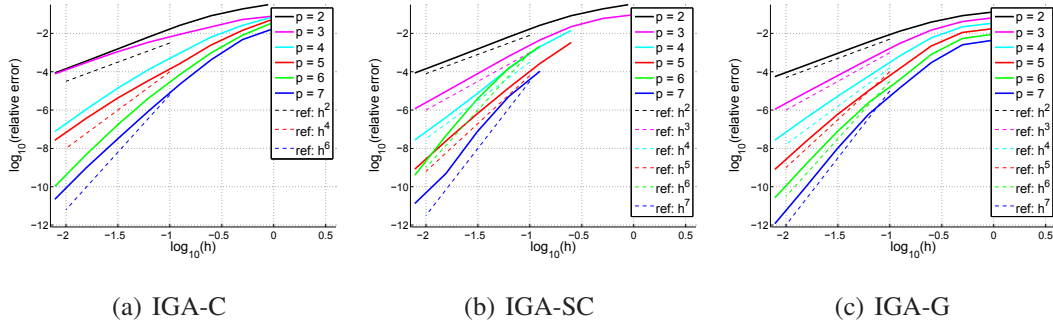


Figure 5.11: The elastic plate with a circular hole problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.  $\log_{10}(\text{relative error}) = \log_{10}(|u_{ex} - u_h|_{H^1} / |u_{ex}|_{H^1})$

**Remark:** In Figures 5.10 and 5.11, some of the results on coarse meshes for  $p = 4$  to  $p = 7$  are not shown. This is because the IGA-SC for those cases leads to an underdetermined system of equations, where there are more degrees of freedom than there are constraints. By comparing the number of basis functions and collocation points, it can be seen that a mesh with at least  $p - 2$  non-empty knots-spans in each direction are needed to obtain a unique solution when  $p$  is odd, and  $p - 1$  non-empty knot-spans are needed when  $p$  is even.

### 5.3.4 Infinite Plate with a Circular Hole under traction boundary conditions

For this benchmark problem, the usual boundary conditions considered are exact traction imposed on the top and left edges of the plate, and symmetry boundary conditions on the bottom and right edges of the modeled domain (i.e. prescribed zero displacement in the vertical direction on the bottom edge and prescribed zero displacement in the horizontal direction on the right edge). We note that in collocation, the zero traction conditions need to be explicitly added as additional constraints in the global linear system. This is different from Galerkin methods where no special treatment is needed on the free boundaries. For this problem, the traction-free boundaries are the bottom edge, right edge and the circular portion of the boundary.

The most straight-forward method of imposing Neumann boundary conditions in collocation is to interpolate the prescribed derivatives at collocation points on the boundary. For simplicity, we use this method and choose the collocation points on the traction boundary to be at the Greville abscissae for both IGA-C and IGA-SC. The results for the energy norm of the error are shown in Figure 5.12. A close examination of the convergence plots show similar patterns to those observed for the pure displacement boundary conditions. However, we note that the accuracy is somewhat decreased compared to the previous results, in particular for coarse meshes as more refinements are needed to approach the asymptotic convergence rates. It is likely that some of the recent developments with regard to imposing Neumann boundary conditions, such as those discussed in <sup>137</sup>, would be helpful in this regard.

## 5.3 Numerical Examples

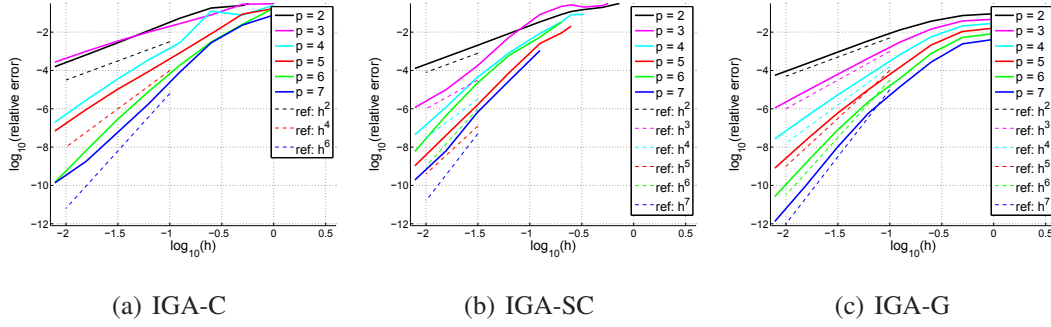


Figure 5.12: The elastic plate with a circular hole problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.  $\log_{10}(\text{relative error}) = \log_{10}(|u_{ex} - u_h|_{H^1} / |u_{ex}|_{H^1})$

### 5.3.5 3D Cube Problem

The next example considers a unit cube domain and the strong form of the problem considered is:

$$\begin{cases} -\Delta u + u = f, & \forall x \in [0, 1] \times [0, 1] \times [0, 1], \\ u|_{\partial\Omega} = 0, \end{cases} \quad (5.20)$$

with

$$f(x, y, z) = (1 + 12\pi^2) \sin(2\pi x) \sin(2\pi y) \sin(2\pi z), \quad (5.21)$$

where the exact solution is:

$$u(x, y, z) = \sin(2\pi x) \sin(2\pi y) \sin(2\pi z). \quad (5.22)$$

We compare IGA-SC with IGA-C and IGA-G solutions for this example, calculating the errors in the  $L^\infty$ -norm,  $W^{1,\infty}$  and  $W^{2,\infty}$ -seminorms. From Figure 6.6, for the  $L^\infty$  norm, we observe that the errors for IGA-C are greater than for IGA-SC, which are greater than the errors for IGA-G, except for the first two coarsest meshes and  $p = 6$ . A similar pattern holds for the other two norms considered, while noting that for  $p = 6$  the error for IGA-SC is initially greater before decreasing to the expected rates as the mesh is refined. This can be seen in the Figures 5.15 and 5.16.

### 5.3 Numerical Examples

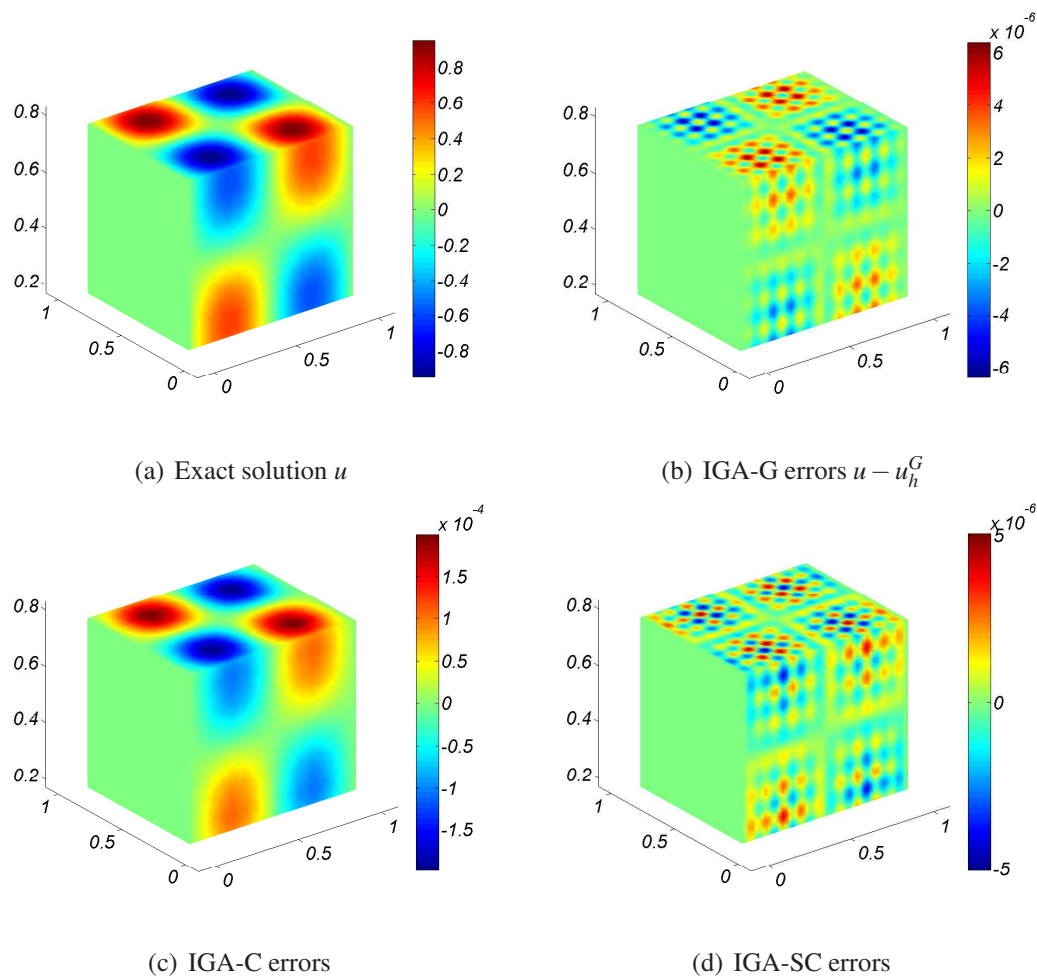


Figure 5.13: Unit cube model problem. (a) exact solution  $u$ , (b) error of the IGA-G  $u - u_h^G$ , (c) error of the IGA-C  $u - u_h^C$  and (d) error of the IGA-SC  $u - u_h^{SC}$ . The B-spline basis functions are of polynomial degree 5 in each direction. The domain is discretized by  $10 \times 10 \times 10$  elements. Shown is the section of the cube with the volume coordinates  $[0, 1] \times [0.1, 0.9] \times [0.2, 0.8]$ .

### 5.3 Numerical Examples

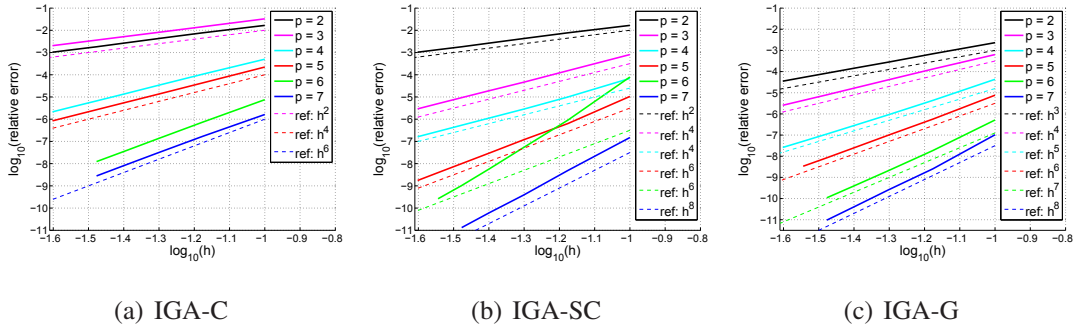


Figure 5.14: Unit cube problem solved by (a) IGA-C (b) IGA-SC and (c) IGA-G.

$$\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{L^\infty} / \|u_{ex}\|_{L^\infty})$$

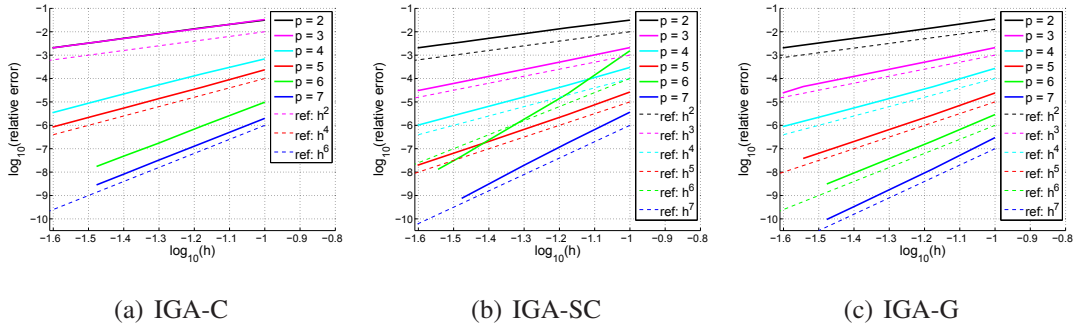


Figure 5.15: Unit cube problem solved by (a) IGA-C (b) IGA-SC and (c) IGA-G.

$$\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{W^{1,\infty}} / \|u_{ex}\|_{W^{1,\infty}})$$

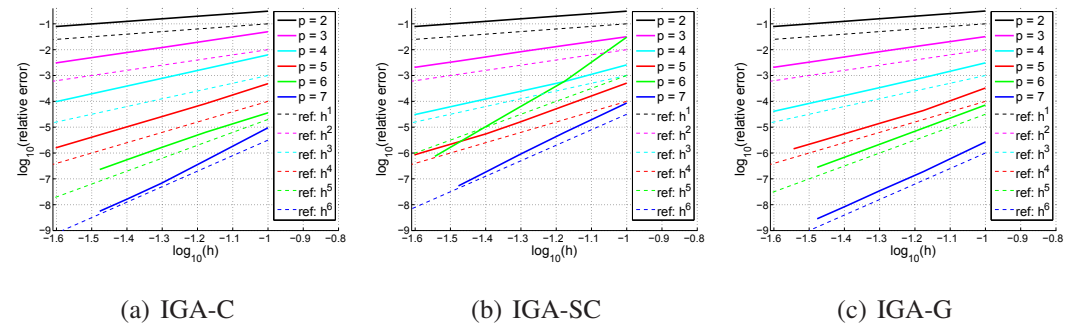


Figure 5.16: Unit cube problem solved by (a) IGA-C (b) IGA-SC and (c) IGA-G.

$$\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{W^{2,\infty}} / \|u_{ex}\|_{W^{2,\infty}})$$



### 5.3.6 Poisson equation on a hollow sphere

In this numerical example, we consider a 3D model of 1/8<sup>th</sup> of a hollow sphere, occupying the domain  $\Omega$  shown in Figure 5.17. The strong form of the model problem studied is:

$$\begin{cases} -\Delta u = f, & \forall x \in \Omega, \\ u|_{\partial\Omega} = 0, \end{cases} \quad (5.23)$$

where  $f$  is manufactured so that the exact solution is:

$$u(x, y, z) = \sin(x) \sin(y) \sin(z) (x^2 + y^2 + z^2 - (R + H)^2) (x^2 + y^2 + z^2 - (R - H)^2), \quad (5.24)$$

where we take  $R = 10$  and  $H = 1$ . We have used an IGA discretization given in<sup>2</sup> for a shell problem (there  $H = 0.02$ ). We note that this is a more severe stress of the IGA formulation, since the mapping from the physical space to the parametric space has a singularity along the  $y$  axis. In fact, the side of the cube corresponding to  $u = 0$  in the parametric space is mapped to an annulus with midsection radius  $R$  in the  $xy$ -plane, while the side corresponding to  $u = 1$  is mapped to the edge of the physical domain above the origin.

### 5.3 Numerical Examples

---

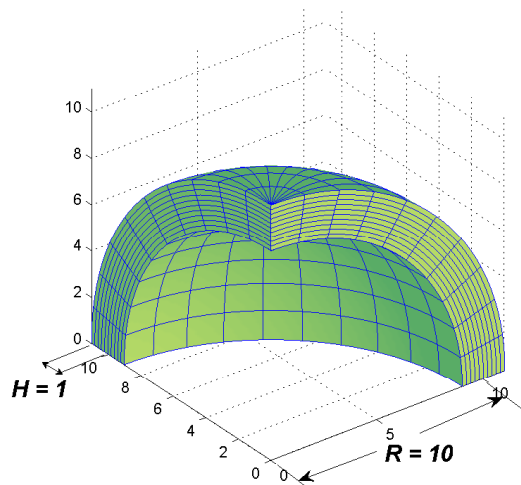


Figure 5.17: 8<sup>th</sup> of a hollow sphere

The results are shown in Figures 5.18 to 5.21. Here we use  $L^2$  norm and  $H^1$ ,  $H^2$  seminorms to report the errors. We note that for  $p = 6, 7$  and the coarsest meshes, the IGA-SC gives poorer results compared to the other two methods. This pattern was also seen in the previous examples, but the differences are more pronounced due to the oscillating character of the exact solution. Also, because superconvergence is not generally realized near the boundary, up to  $2p + 1$  knot-spans in each direction are needed for reliable results. Asymptotically, for odd  $p$ , IGA-G and IGA-SC are quite close, while IGA-C is suboptimal in the  $L^2$  norm and  $H^1$  semi-norm.

### 5.3 Numerical Examples

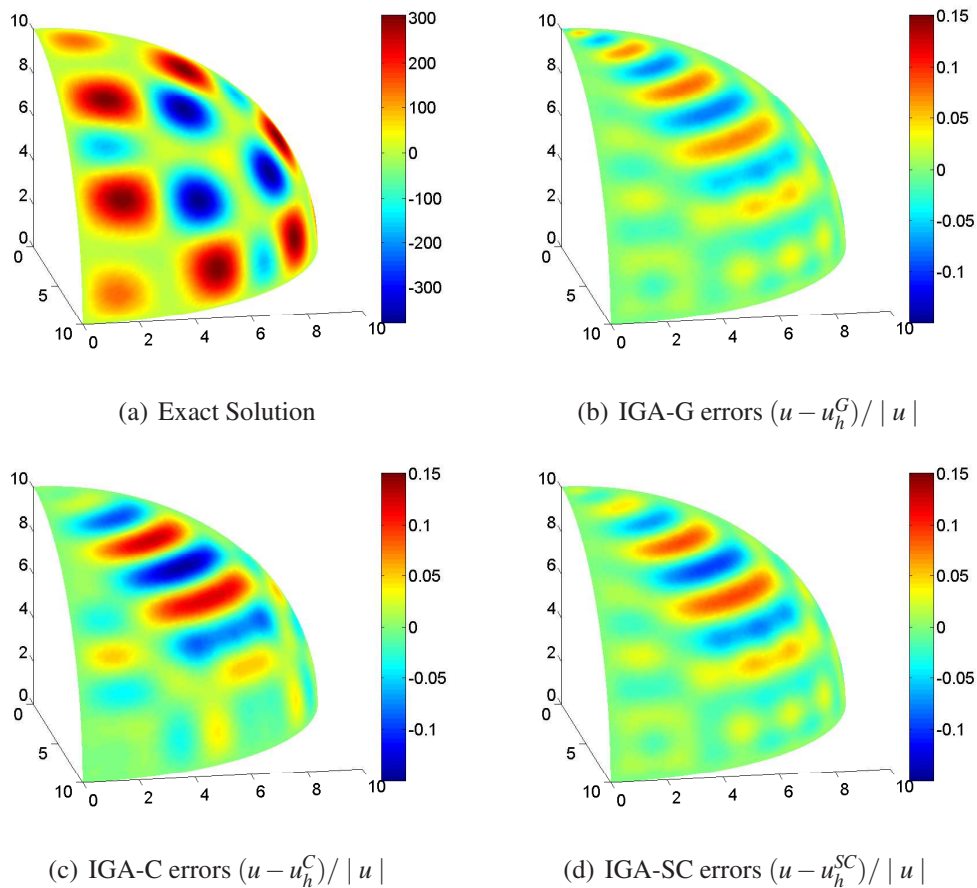


Figure 5.18: Hollow sphere problem with the solution and errors plotted on the mid-surface of the sphere (at radius  $r = R$ ). (a) exact solution  $u$ , (b) relative error of the IGA-G  $(u - u_h^G) / |u|$ , (c) relative error of the IGA-C  $(u - u_h^C) / |u|$  and (d) relative error of the IGA-SC  $(u - u_h^{SC}) / |u|$ . The B-spline basis functions are of polynomial degree 5 in each direction. The domain is discretized by  $10 \times 10 \times 10$  elements.

### 5.3 Numerical Examples

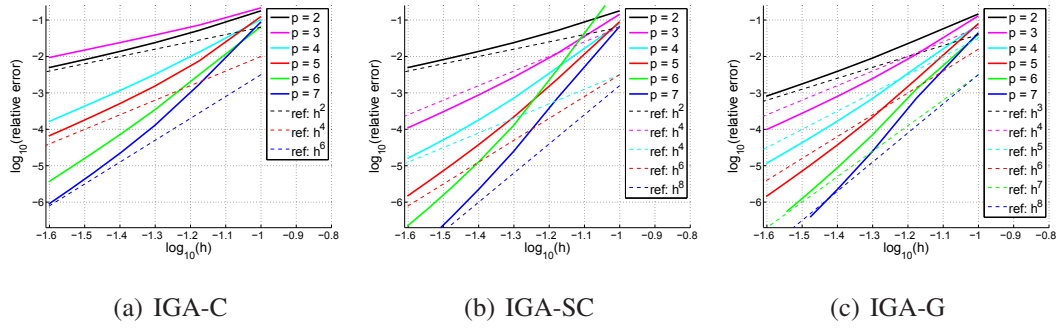


Figure 5.19: Hollow sphere problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.

$$\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{L^2} / \|u_{ex}\|_{L^2})$$

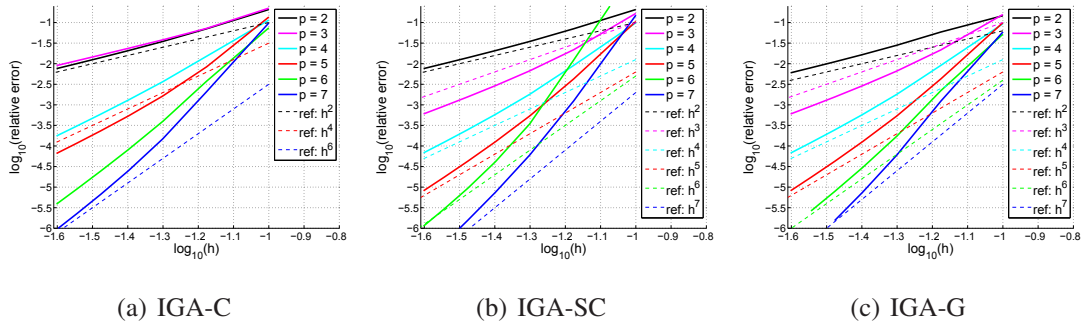


Figure 5.20: Hollow sphere problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.

$$\log_{10}(\text{relative error}) = \log_{10}(|u_{ex} - u_h|_{H^1} / |u_{ex}|_{H^1})$$

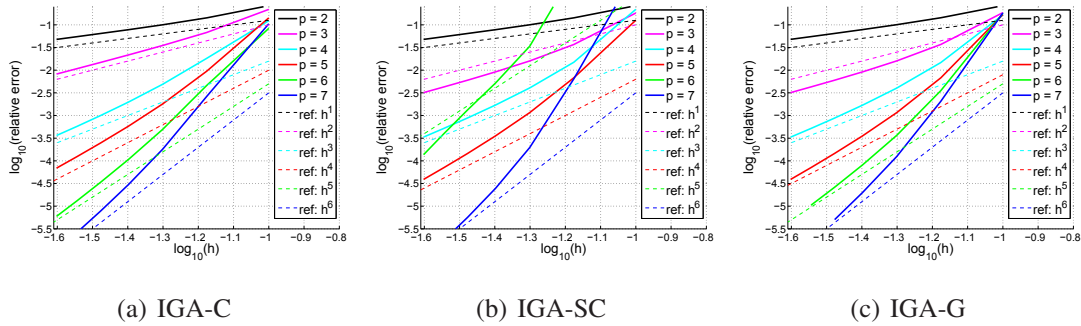


Figure 5.21: Hollow sphere problem solved by (a) IGA-C, (b) IGA-SC and (c) IGA-G.

$$\log_{10}(\text{relative error}) = \log_{10}(|u_{ex} - u_h|_{H^2} / |u_{ex}|_{H^2})$$

### 5.4 Analysis of algorithmic efficiency

In this section, we will calculate the number of floating point operations (*flops*) required for the methods discussed here. We note that a detailed comparison between IGA-C and IGA-G in terms of cost vs. accuracy has already been conducted in<sup>39</sup>. In the following, we will extend this analysis to include the properties and the results obtained for the numerical examples with IGA-SC.

A typical finite element or isogeometric analysis program has three components: pre-processing, analysis and post-processing. In the first part, the geometry of the domain is discretized and the connectivity between the quadrature or the collocation points and the shape functions is established. In the analysis component, a system of algebraic equations which incorporates the governing equations and the boundary conditions is assembled and then solved. Finally, the post-processing module is where the solution is displayed. Since the shape functions for a given discretization are the same among all 3 isogeometric methods considered, only the analysis component differs substantially between the three isogeometric methods studied here. Therefore, we will focus on this part, in particular on the costs involved in forming the left-hand side matrix (or stiffness matrix) and that of solving the linear system of equations, as those are the most expensive in terms of computational costs.

#### 5.4.1 Assembling the linear system

For simplicity, we will assume that the domain of the problem is discretized into  $n$  elements (or positive area knot-spans) in each spatial direction using NURBS of degree  $p$ . The number of dimensions of the physical space is denoted by  $d$ , and we denote by  $N$  the number of degrees of freedom in the discretization (which corresponds to the number of columns in the stiffness matrix). If we assume that NURBS of maximal continuity ( $C^{p-1}$ ) are used, then there are  $N = (n + p)^d$  degrees of freedom for a scalar problem (such as Poisson's equation) and  $d$  times more degrees of freedom for a vector problem (such as linear elasticity).

## 5.4 Analysis of algorithmic efficiency

---

The total number of flops for assembling the stiffness matrix (without taking into account the boundary conditions) is given by the flops needed for assembly the local collocation/stiffness matrix at one interior collocation/quadrature point multiplied by the number of evaluation points. Assembly of the local collocation matrix in IGA-SC is exactly the same as in IGA-C, however there are  $(n + p - 2)^d$  interior collocation points in IGA-C and  $(2n)^d$  or  $(2n - 1)^d$  evaluation points in IGA-SC for odd  $p$  or for even  $p$  respectively. The number of flops for evaluating the 2<sup>nd</sup> derivatives of the basis functions, as required for collocation, is more than that required for Galerkin method which needs only the first derivatives, although both require  $O(p^d)$  flops. However, for the scalar problem, the local collocation matrix has dimensions  $1 \times (p + 1)^d$ , while the local stiffness matrix has dimensions  $(p + 1)^d \times (p + 1)^d$ , requiring  $O(p^{2d})$  flops to compute. For a vector problem, the local collocation matrix has dimensions  $d \times d(p + 1)^d$ , while the local stiffness matrix has dimensions  $d(p + 1)^d \times d(p + 1)^d$ . For a detailed derivation of the flop counts we refer to<sup>39</sup>.

For the right-hand side (residual) vector, the cost for the collocation methods is lower since it requires just the evaluation of the volume load function at the given collocation point, while for the Galerkin method it requires the evaluation of a volume integral. Therefore for a scalar problem, the cost of IGA-C and IGA-SC is  $O(1)$  per collocation point and  $O(p^d)$  per quadrature point for IGA-G. For a vector problem, the cost for all the methods is increased by a factor of  $d$ . Since in all the examples studied, we consider  $d \leq 3$ , we neglect this additional cost.

Imposing the boundary conditions has a cost that depends on the type of boundary conditions and the method used. Regardless of this, since the the imposition of boundary conditions only affects the shape functions that have support on the domain boundaries, it requires  $O(n^{d-1})$  flops for a scalar problem (and again  $d$  times more for elasticity). In Table 5.2 we summarize the number of flops required by the three methods for the assembly of the linear system of equations. For computing the total time, we have considered only the highest exponents for  $n$  and  $p$  that appear in the flops calculations and neglected the lower order terms.

## 5.4 Analysis of algorithmic efficiency

Table 5.2: Number of flops for assembling the linear system in IGA-C, IGA-SC and IGA-G

	IGA-C	IGA-SC ( $p \geq 3$ )	IGA -G
Number of collocation/ quadrature points	$(n + p - 2)^d$	$(2n)^d$ odd $p$ $(2n - 1)^d$ even $p$	$n^d(p + 1)^d$
Cost of assembling the local matrix for scalar problem (per quadrature/collocation point)	$O(p^d)$	$O(p^d)$	$O(p^{2d})$
Cost of assembling the local matrix for vector problem (per quadrature/collocation point)	$O(d^2 p^d)$	$O(d^2 p^d)$	$O(d^2 p^{2d})$
Cost of assembling the RHS vector for scalar/vector problem (per quadrature/collocation point)	$O(1)$	$O(1)$	$O(p^d)$
Cost of imposing boundary conditions for scalar problem (for entire domain boundary)	$O(n^{d-1})$	$O(n^{d-1})$	$O(n^{d-1})$
Cost of imposing boundary conditions for vector problem (for entire domain boundary)	$O(dn^{d-1})$	$O(dn^{d-1})$	$O(n^{d-1})$
Total cost when $n \gg p$ (scalar problem)	$O((np)^d)$	$O((2np)^d)$	$O((p^3n)^d)$
Total cost when $n \gg p$ (vector problem)	$O(d^2(np)^d)$	$O(d^2(2np)^d)$	$O(d^2(p^3n)^d)$

### 5.4.2 Solving the linear system

The last part of the analysis is generally solving the linear system obtained from the problem discretization. There are two types of solvers that can be used: direct and iterative solvers. Direct solvers are typically more efficient for small linear systems, but require progressively more memory as the number of degrees of freedom is increased. Iterative solvers are more economical in terms of memory use but their performance depends on the condition number of the matrix and, implicitly, on the type of preconditioner being used. A detailed comparison between IGA-C and IGA-G in terms of the floating point operations required to solve the linear system has also been conducted in<sup>39</sup>. Therefore, in the following we will focus on calculating the cost of obtaining the least squares solution of an overdetermined, full rank system obtained from the IGA-SC method.

For obtaining a least-squares solution, one of two classes of algorithms are generally used: forming the normal equations or QR decomposition. The method of normal equations solves the system  $\mathbf{K}\mathbf{x} = \mathbf{b}$  by first multiplying both sides by  $\mathbf{K}^T$  to get:

$$(\mathbf{K}^T \mathbf{K})\mathbf{x} = \mathbf{K}^T \mathbf{b}. \quad (5.25)$$

It can easily be seen that the matrix  $\mathbf{K}^T \mathbf{K}$  is square, symmetric and positive-definite. Therefore the linear system in (5.25) can be solved by most standard direct or iterative solvers.

Similarly to IGA-C, the collocation matrix  $\mathbf{K}$  in IGA-SC is a sparse banded matrix, where each row corresponds to a collocation point and each column corresponds to a basis function. For a scalar problem, it has at most  $(p+1)^d$  non-zero entries for each row. Because each positive area knot-span contains up to  $2^d$  collocation points, there are at most  $(2(p+1))^d$  non-zero entries in each column of  $\mathbf{K}$ . We note that solving (5.25) is equivalent to an  $L^2$  projection over the set of interior collocation points  $\{x_k^{int}\}$ ,



## 5.4 Analysis of algorithmic efficiency

$k = 1, \dots, N^{int}$ , i.e. if  $\mathbf{A} = [a_{ij}] = \mathbf{K}^T \mathbf{K}$ , then

$$a_{ij} = \sum_{k=1}^{N^{int}} \mathcal{L} \phi_i(x_k^{int}) \mathcal{L} \phi_j(x_k^{int}),$$

where  $\mathcal{L}$  is the differential operator in the strong form of the problem as defined in (5.1). In practice, the sum is taken only over the points  $x_k^{int}$  which are in the intersection of the supports of the basis functions  $\phi_i$  and  $\phi_j$ . Therefore, forming the  $N \times N$  normal matrix requires  $O(Np^{2d})$  flops, where  $N$  is the number of degrees of freedom. Moreover the sparsity pattern and the number of non-zero entries for the normal matrix  $\mathbf{A}$  is the same as for the stiffness matrix in the Galerkin method. The collocation matrix for IGA-C, the normal matrix for IGA-SC and the stiffness matrix for IGA-G for a particular value of  $p$ ,  $n$  and  $d$  are shown in Figure 5.22.

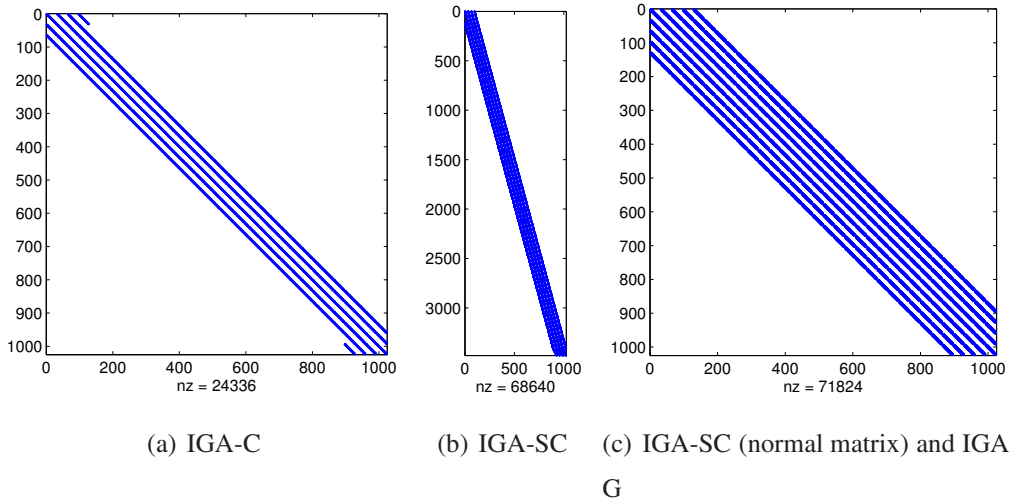


Figure 5.22: The sparsity pattern and number of non-zero entries of the left hand side matrix of the linear system for the three methods considered, with  $p = 4$ ,  $d = 2$ ,  $n = 30$ .

It is known that for sparse banded matrices, the cost of a direct solver using LU factorization or (Cholesky factorization for symmetric matrices) is proportional to the product of the upper and lower bandwidth of a matrix. We say that a matrix  $\mathbf{A} = [a_{ij}]$  has lower bandwidth  $k_1$  and upper bandwidth  $k_2$  if  $k_1$  and  $k_2$  are the smallest non-

## 5.4 Analysis of algorithmic efficiency

---

negative integers such that  $a_{ij} = 0$  for  $i > j + k_1$  and  $a_{ij} = 0$  for  $j > i + k_2$ . In particular, the number of flops required for the factorization by Gaussian elimination is  $O(Nk_1k_2)$ <sup>139</sup>. Moreover, the bandwidth of the matrices  $L$  and  $U$  is  $k_1$  and  $k_2$  respectively, which has implications for the memory requirements of the factorization.

The lower and upper bandwidths of the normal matrices resulting from IGA-SC are presented in Table 5.3. The values for IGA-C and IGA-G have also been calculated in<sup>39</sup>. We note that although the matrix arising from IGA-C is non-symmetric, the lower and upper bandwidths are equal when a sensible numbering of shape functions is used. The normal matrices from IGA-SC and the stiffness matrix in IGA-G are symmetric in the examples considered here.

Table 5.3: Lower/Upper bandwidth of the linear system matrices in IGA-C, IGA-SC and IGA-G for solving a scalar problem with  $n$  knot-spans in each direction and polynomial degree  $p$ , when  $n \gg p$

	IGA-C	IGA-SC and IGA -G
d=1	$p/2$ or $(p-1)/2$	$p$
d=2	$O(np/2)$	$O(np)$
d=3	$O(n^2p/2)$	$O(n^2p)$

We can expect that in terms of operation counts and memory requirements, the LU factorization of IGA-SC would be of similar cost as IGA-G and approximately 4 times more costly as IGA-C. However, one can exploit the symmetry of the matrices in IGA-G and IGA-SC to significantly reduce the memory and computation costs.

For ill-conditioned problems, it is desirable to use the QR factorization method instead of forming the normal equation, but this comes at the expense of increased memory requirements and computational costs. For dense matrices, QR factorization requires approximately twice the number of flops<sup>139</sup>. In our applications, we have observed a similar increase in computational times.

## 5.4 Analysis of algorithmic efficiency

---

For large problems, especially in 3 dimensions, the usage of direct solvers becomes problematic, particularly due to the increased memory requirements<sup>140</sup>. Although the matrices  $L$  and  $U$  inherit the bandwidth of the stiffness or collocation matrices, they typically have a much larger number of non-zero entries due to the “fill-in” that occurs during row reduction operations. A more efficient approach is the use of iterative solvers, coupled with an appropriate pre-conditioner.

We note that finding optimal pre-conditioners for IGA-G and IGA-C is still subject of ongoing research, see for example<sup>141</sup> or<sup>142</sup> for some recent developments. For our 3D examples using Poisson’s equation, we have obtained significant improvements by using incomplete LU (ILU) and incomplete Cholesky factorizations. The idea is to perform LU or Cholesky factorization with zero fill-in, i.e. any entry that is zero in the original matrix is also zero in the resulting decomposition. For IGA-C, which is non-symmetric, we have used a GMRES (Generalized Minimal Residual) solver. To solve the normal equations in IGA-SC and for IGA-G, the preconditioned conjugate gradient (PCG) solver was found to be more efficient, as it exploits the matrix symmetry. We give a more detailed discussion of the computing time needed for these solvers in the next subsection.

### 5.4.3 Computational cost vs. computational accuracy

The flip-side of measuring the efficiency of an algorithm is the question of how accurate it is. It is known that the Galerkin method for IGA converges at “optimal rates” for smooth problems, i.e. the same as the finite elements with the same polynomial degree  $p$ , while collocation methods have lower or equal rates of convergence.

Table 5.4 gives an overview of the convergence rates, in terms of a mesh size parameter  $h$ , obtained by the three methods currently studied. For the collocation methods, we also differentiate based on whether the polynomial degree  $p$  of the basis is odd or even. In particular, we note that for the  $L^\infty$  and  $L^2$  norms IGA-C is never optimal, while IGA-SC is optimal for  $p$  odd. For  $W^{1,\infty}$  and  $H^1$  seminorms, only IGA-C with  $p$  odd is not optimal. For the second derivative seminorms, all methods considered

## 5.4 Analysis of algorithmic efficiency

---

converge as  $O(h^{p-1})$ .

Table 5.4: The convergence rates of the current numerical methods

	$L^\infty, L^2$	$W^{1,\infty}, H^1$	$W^{2,\infty}, H^2$
IGA-C $p$ -even	$O(h^p)$	$O(h^p)$	$O(h^{p-1})$
IGA-C $p$ -odd	$O(h^{p-1})$	$O(h^{p-1})$	$O(h^{p-1})$
IGA-SC $p$ -even	$O(h^p)$	$O(h^p)$	$O(h^{p-1})$
IGA-SC $p$ -odd	$O(h^{p+1})$	$O(h^p)$	$O(h^{p-1})$
IGA-Galerkin	$O(h^{p+1})$	$O(h^p)$	$O(h^{p-1})$

While mathematically it is preferable to have a faster-converging method at the expense of an additional cost which is bounded by a fixed multiplicative factor, this is not always advantageous when this constant factor is very large. One way to better analyze the costs vs. benefits ratio is to plot the time to the solution on the horizontal axis vs. the error on the vertical axis as shown below.

The results shown in this section have been computed on a workstation with an Intel Xeon X5472 CPU running at 3.0 GHz and 32 GB of RAM. All codes have been written in MATLAB and run with the “-singleCompThread” option to restrict the program to a single core. The times shown are the “wall-clock times” (i.e. real times) measured from the start of the program until the full solution vector is obtained, including pre-processing, computation of the basis functions and assembly of the matrix system and the linear solver.

In Figure 5.23 we show the results for the 2D example in Section 5.3.2 (quarter of an annulus problem). The solving of the linear system in IGA-SC we used the direct solver based on sparse QR factorization provided by the MATLAB “mldivide” operator. For the other methods, direct solvers were also used. We note that for odd  $p$ , IGA-SC generally performs better than IGA-C or IGA-G, though the advantage diminishes as the polynomial degree increases because of the initial poorer performance

## 5.5 Conclusions

---

on coarse meshes. For even  $p$ , IGA-SC has the same order of convergence as IGA-C and it is typically not as efficient, however the difference is a fixed constant factor.

Next, we consider the hollow sphere example studied in Section 5.3.6. The time vs. relative error plots are shown in Figure 5.24. Because the linear systems arising from this example are quite large (up to 100,000 degrees of freedom for IGA-SC and IGA-G and up to 300,000 for IGA-C), we study the performance of both direct and iterative solvers for  $p = 3, 4$ , and 5. For IGA-SC, the normal equation is formed and then solved using either a direct solver or PCG with an incomplete Cholesky preconditioner. The same solvers are used for IGA-G. For IGA-C we used either a direct solver or GMRES coupled with an incomplete LU preconditioner.

We observe that the plotted curves descend at an initially faster rate as the linear systems are small and then the error decrease slows down as the number of degrees of freedom increases and a larger percentage of the time is spent in the linear solver. This slow-down is more evident for the direct solvers. Regardless of the solver used, we notice again that IGA-SC is more efficient for fine meshes and odd  $p$ , while for even  $p$  IGA-C is typically better by some fixed factor.

## 5.5 Conclusions

A noteworthy result is that the proposed collocation method has the same convergence rates as the Galerkin method for odd  $p$ . Moreover for these values of  $p$ , the two approximation schemes appear to converge asymptotically to the same relative error when the mesh is refined. This suggests that collocation methods and methods based on the weak form are more closely related than it would initially appear. This can also be seen from some of the reduced quadrature rules that have been proposed in<sup>50</sup> (such as the center-vertex rules and reduced Gauss quadratures), which correspond to the similar locations for the evaluation of the basis functions in these collocation methods.

A more thorough theoretical investigation of collocation methods, as well as applications to more interesting engineering problems that include nonlinearities remains the

## 5.5 Conclusions

subject of future work.

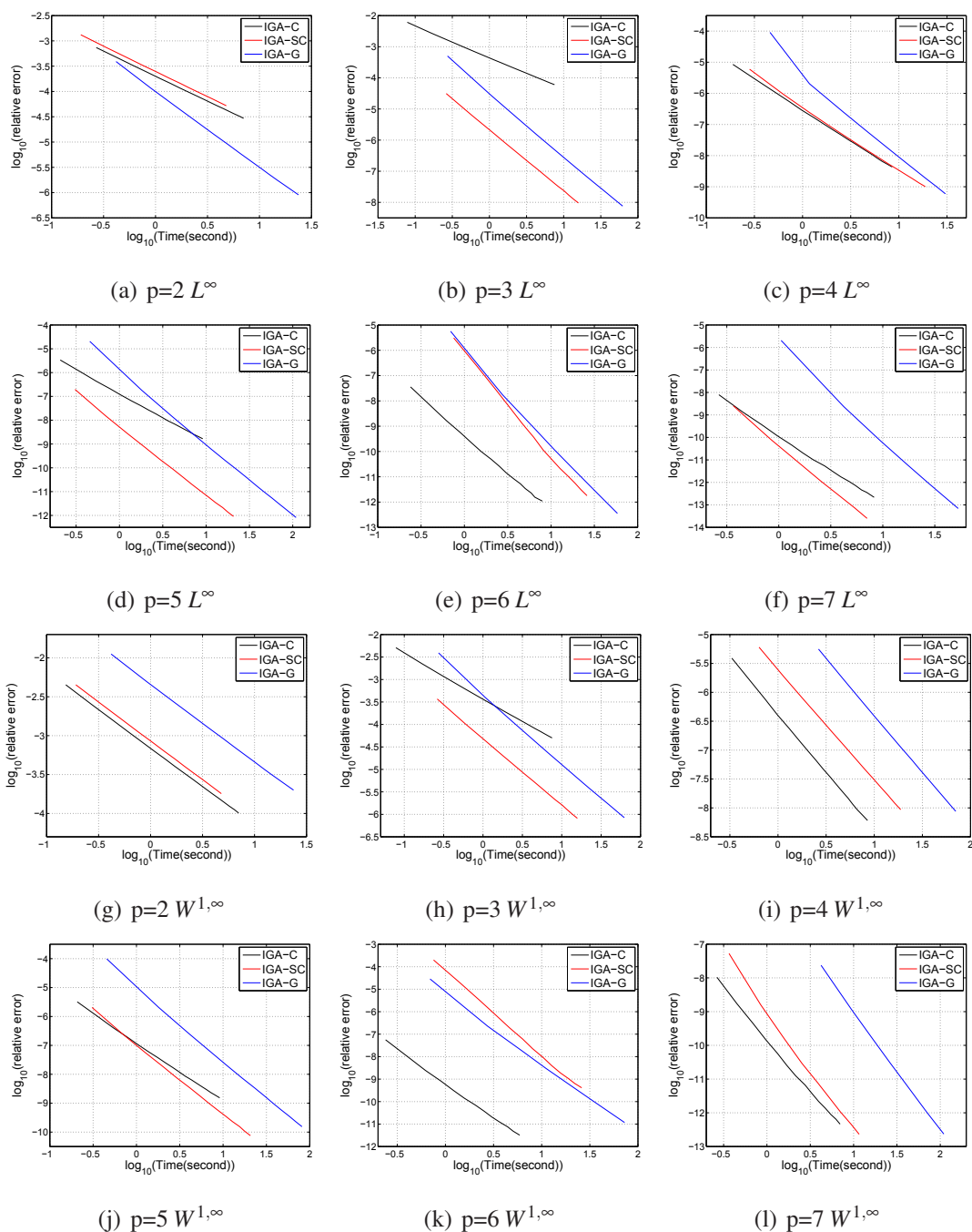


Figure 5.23: time vs. relative error of the 2D annulus example. (a-f) are the comparison of the  $L^\infty$ -norm results. (g-l) are the comparison of the  $W^{1,\infty}$ -norm results.

## 5.5 Conclusions

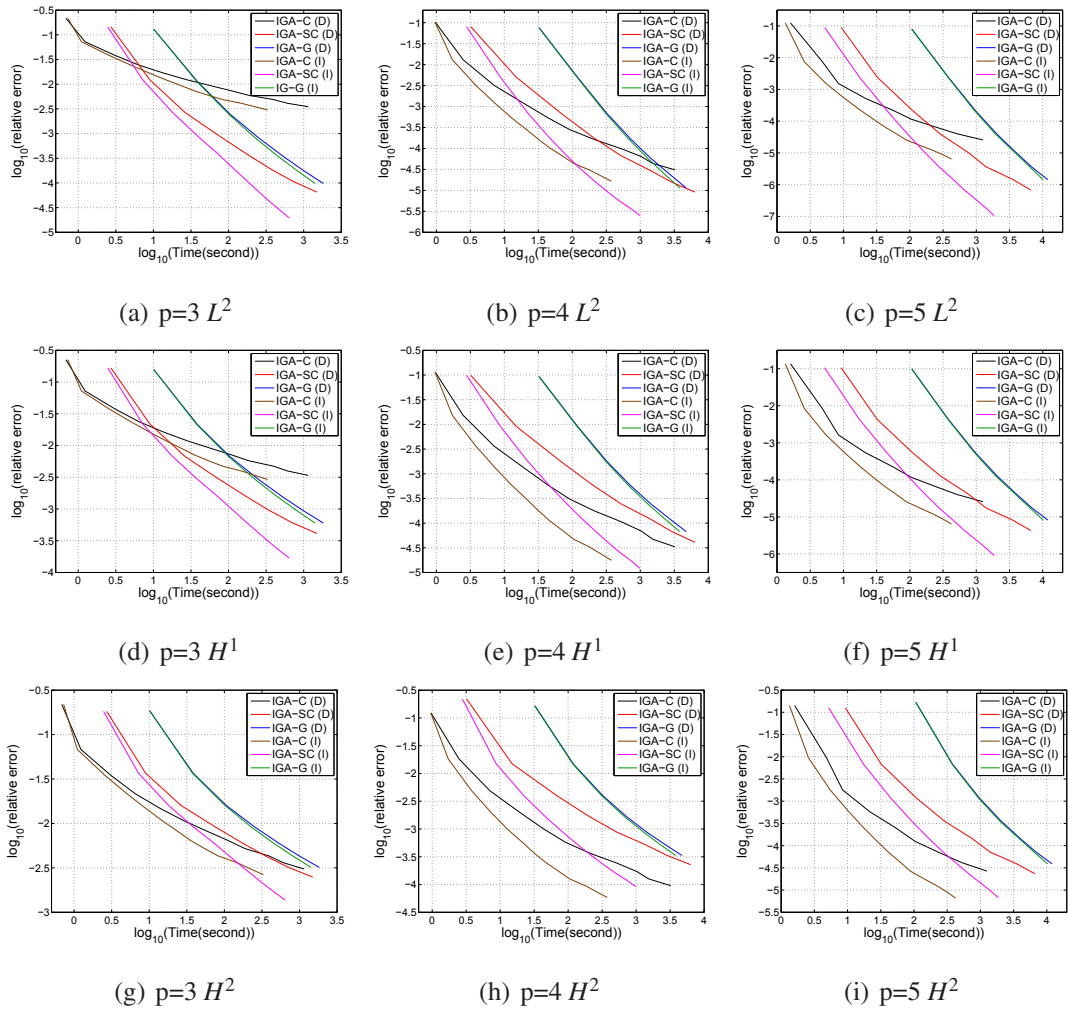


Figure 5.24: Accuracy-to-time comparison of the 3D sphere example. In (a-c)  $\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{L^2} / \|u_{ex}\|_{L^2})$ . In (d-f)  $\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{H^1} / \|u_{ex}\|_{H^1})$ . In (g-i)  $\log_{10}(\text{relative error}) = \log_{10}(\|u_{ex} - u_h\|_{H^2} / \|u_{ex}\|_{H^2})$ .

# Chapter 6

## A novel multilevel technique for solving image registration

### 6.1 Introduction

Image registration techniques, developed in recent years, aim to align two images by finding a spatial transformation<sup>70,71,72</sup>. These methods fall mainly into three basic categories, the landmark-based registration<sup>73,74,75</sup>, the segmentation-based registration<sup>76,77,78</sup> and the image intensity-based registration<sup>79,80</sup>. They have many applications such as medical imaging<sup>81,82</sup>, remoting sensing<sup>83,84</sup> and computer vision<sup>85</sup>. In addition to various types of spatial transformations, image registration approaches can also be divided into rigid registration and non-rigid (or deformable) registration. The landmark-based registration first identifies a limit set of landmark points on the target image, and also some corresponding salient points on the reference image. This step can be done manually or automatically. Then the spatial transformation can be defined based on the relationship among these points. The next step is to minimize the spatial transformation, which is also known as the orthogonal Procrustes problem<sup>143,144</sup>. The segmentation-based registration methods utilize segmented parts of the reference image and the target image and seeks to align them together. Apart from the



## 6.2 Dynamic Mathematical Modeling

---

landmark-based and the segmentation-based methods, the intensity-based registration methods directly operate on the image intensity values. The spatial transformations are based on the intensity information recorded in both images, combining the selected basis functions and transferring the discrete digital image data into a continuous expression. From a theoretical point of view, the intensity-based registration methods are the most flexible because they use all the available information throughout the registration process.

In this part, we present a novel dynamic multilevel technique for solving image registration problems. The development is carried out to construct a spatial transformation based on cubic B-spline basis functions and determine the control points dynamically. Unlike FEM-based image registration methods, we do not have the difficulty of solving a complicated matrix system. In addition, the presented method is enhanced by a multilevel technique, which makes it more efficient and flexible. The numerical results and several comparison studies on real bio-medical images show our technique is stable, accurate and fast, especially for large deformation registration problems.

## 6.2 Dynamic Mathematical Modeling

Suppose two images, the reference image  $I_1(\mathbf{x})$  and the target image  $I_0(\mathbf{x})$  are given, image registration aims to find a continuous spatial transformation  $f(\mathbf{x})$  such that  $I_1(f(\mathbf{x})) \approx I_0(\mathbf{x})$ . The spatial transformation is written as

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$f(\mathbf{x}) = \sum_{i=1}^N \mathbf{C}_i \phi_i(\mathbf{x}), \quad (6.1)$$

where  $\mathbf{C}_i$  are the control points,  $\phi_i(\mathbf{x})$  are the corresponding global basis functions, and  $N$  is the number of global basis functions applied to the registration model. The global basis functions are a combination of the cubic B-splines in two dimensions. Cubic B-

## 6.2 Dynamic Mathematical Modeling

---

splines are the most common basis functions used in image processing studies. They have many good properties, such as small overlap, local support and  $C^2$  smoothness. These advantages of cubic B-splines have been studied and compared with harmonic functions, radial basis functions and wavelets<sup>80</sup>. The recursive formula of B-splines can be written as,

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u \in [u_i, u_{i+1}) \\ 0 & \text{otherwise} \end{cases}, \quad (6.2)$$

and

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u), \quad (6.3)$$

where  $N_{i,p}$  represents the  $i^{\text{th}}$  B-spline function of polynomial order  $p + 1$ . It is defined on a knot vector  $\{u_1, \dots, u_m\}$  in the  $u$  direction. Likewise in 2D, there is a sequence of B-splines in the  $v$  direction associated with the knot vector  $\{v_1, \dots, v_n\}$ . When  $p = 0$ ,  $N_{i,0}$  becomes a step function and equals zero everywhere except on the half open interval  $[u_i, u_{i+1})$ .

In this paper, we focus on 2D image registration problems. The basis functions  $\phi_i$  in Equation 6.1 are a combination of cubic B-splines, and the cubic B-splines in each direction are defined on open knot vectors, which means the starting and ending knots are repeated by  $p + 1$  times. Thus  $N_{j,3}(u)$  are the cubic B-splines in the  $u$  direction with an open knot vector  $\xi^u = \{u_1, u_1, u_1, u_1, u_2, \dots, u_{m-1}, u_m, u_m, u_m, u_m\}$ , and  $N_{k,3}(v)$  are the cubic B-splines in the  $v$  direction with an open knot vector  $\xi^v = \{v_1, v_1, v_1, v_1, v_2, \dots, v_{n-1}, v_n, v_n, v_n, v_n\}$ . Then the global basis functions  $\phi_i(\mathbf{x})$  are expressed by

$$\phi_i(\mathbf{x}) = N_{j,3}(u)N_{k,3}(v), \quad (6.4)$$

where  $\mathbf{x} = (u, v)$ ,  $j = 1, \dots, m + p + 1$ ,  $k = 1, \dots, n + q + 1$ , and  $i = 1, \dots, (m + p + 1)(n + q + 1)$ . For details of the cubic B-spline, we refer interested readers to<sup>2,92,145</sup>. For the initial control points  $\mathbf{C} = (C_x, C_y)$ , we use the location of the Greville Abcissae

93,

$$C_x(i) = \frac{\sum_{j=i+1}^{i+p} \xi_j^u}{p} \quad \text{and} \quad C_y(i) = \frac{\sum_{j=i+1}^{i+q} \xi_j^v}{q},$$

where  $\xi_k^u$  and  $\xi_k^v$  are the  $k^{\text{th}}$  entries in  $\xi^u$  and  $\xi^v$ , respectively. Here, we consider equally spaced knots except for the ones in the beginning and at the end of the knotvectors which are repeated 4 times. Since our initial control points are defined by the Greville Abscissae, the initial control points are also equally spaced except near the domain boundaries.

The FEM-based registration methods<sup>1,146,147</sup> start with a pre-defined energy functional, which is used to measure the differences between  $I_0(\mathbf{x})$  and  $I_1(\mathbf{x})$ . It can be written as<sup>1</sup>,

$$\begin{aligned} E(\mathbf{x}) = & \int_{\Omega} (I_0(\mathbf{x}) - I_1(f(\mathbf{x})))^2 \mathbf{d}\Omega \\ & + \lambda_1 \int_{\Omega} (\|f_{,u}(\mathbf{x})\|^2 + \|f_{,v}(\mathbf{x})\|^2) \mathbf{d}\Omega \\ & + \lambda_2 \int_{\Omega} (\|f_{,u}(\mathbf{x})\|^2 \|f_{,v}(\mathbf{x})\|^2 - (\langle f_{,u}(\mathbf{x}), f_{,v}(\mathbf{x}) \rangle)^2) \mathbf{d}\Omega, \end{aligned} \quad (6.5)$$

where  $\lambda_1$  and  $\lambda_2$  are two weighting factors, and we set  $\lambda_1 = \lambda_2 = 0.0001$  in our numerical examples.  $f_{,u}(\mathbf{x})$  and  $f_{,v}(\mathbf{x})$  denote the derivatives of the spatial transformation  $f(\mathbf{x})$  with respect to  $u$  and  $v$ , respectively.  $\langle \cdot, \cdot \rangle$  stands for the inner product operator. The first term is called the fidelity term, which is the main component of this energy function used to minimize the differences between the reference and the target images. The second and third terms are two regularization terms. Following<sup>1</sup>, the first regularization term is called “the first-order regularization term”, which is used to keep the spacial transformation  $f(\mathbf{x})$  changing uniformly with respect to  $u$  and  $v$ . The second regularization term is called “the area regularization term”, which contains  $f(\mathbf{x})$  such that the area element remains consistent.

FEM-based image registration methods usually follow three main procedures: defining an energy equation, minimizing it, and finally constructing a matrix system. It is

## 6.2 Dynamic Mathematical Modeling

---

time consuming and it also influences the accuracy of the model. There are some new advances and improvements in recent years. For example in <sup>1</sup>, intermediate images are computed between two given ones, without distinguishing the reference image and the target image. A new energy function is considered and complicated variational formula are derived, which is different with the traditional FEM procedures<sup>66,117</sup>. The validation of the matrix system is based on the Fréchet differential of energy function and a metric function. However, these extended steps still lead to large matrix systems. Thus we endeavor to find another approach which is more computationally efficient. Different from the FEM-based methods, in the following we will handle the energy function directly and drive its differential formula. In addition, we also consider a new coefficient factor for the fidelity term.

We minimize the energy function (Equation 6.5) to dynamically solve the optimal control points for the spacial transformation. These procedures are written as

$$\frac{d\mathbf{C}}{dt} = -\delta E(\mathbf{x}), \quad (6.6)$$

and

$$\mathbf{C}^k = \mathbf{C}^{k-1} - \varepsilon \delta E(\mathbf{x}), \quad (6.7)$$

where  $\delta E(\mathbf{x})$  is the differential of the energy function,  $\mathbf{C}^k$  and  $\mathbf{C}^{k-1}$  are the control points calculated by the  $k^{th}$  and the  $(k-1)^{th}$  iteration steps, and  $\varepsilon$  is the time step chosen by the user. We calculate the differential of the energy function as follows

$$\begin{aligned} \delta_i E(\mathbf{x}) = & -2 \int_{\Omega} g(I_0(\mathbf{x}) - I_1(f(\mathbf{x}))) \nabla I_1(f(\mathbf{x})) \frac{df(\mathbf{x})}{d\mathbf{C}_i} d\Omega \\ & + \lambda_1 \int_{\Omega} (2f_{,u}(\mathbf{x}) \frac{df_{,u}(\mathbf{x})}{d\mathbf{C}_i} + 2f_{,v}(\mathbf{x}) \frac{df_{,v}(\mathbf{x})}{d\mathbf{C}_i}) d\Omega \\ & + \lambda_2 \int_{\Omega} (2f_{,u}(\mathbf{x}) \|f_{,v}(\mathbf{x})\|^2 \frac{df_{,u}(\mathbf{x})}{d\mathbf{C}_i} + 2\|f_{,u}(\mathbf{x})\|^2 f_{,v}(\mathbf{x}) \frac{df_{,v}(\mathbf{x})}{d\mathbf{C}_i} \\ & - 2\langle f_{,u}(\mathbf{x}), f_{,v}(\mathbf{x}) \rangle f_{,v}(\mathbf{x}) \frac{df_{,u}(\mathbf{x})}{d\mathbf{C}_i} \\ & - 2f_{,u}(\mathbf{x}) \langle f_{,v}(\mathbf{x}), f_{,u}(\mathbf{x}) \rangle \frac{df_{,v}(\mathbf{x})}{d\mathbf{C}_i}) d\Omega, \end{aligned} \quad (6.8)$$

## 6.2 Dynamic Mathematical Modeling

---

where

$$g = \frac{1}{\sqrt{\gamma + (\nabla_x I_1(f(\mathbf{x})))^2 + (\nabla_y I_1(f(\mathbf{x})))^2}},$$

$\gamma$  is a small number and chosen to be  $10^{-12}$  in all our case studies,  $g$  is a factor defined to accelerate the evolution process.  $g$  is treated as a constant number calculated from the previous iteration step. Different from the  $g$  factor defined in<sup>1</sup>, here we use the gradient of the deforming image, because it can speed up the evolution process in the homogeneous region, while slow down in the in-homogeneous region. By substituting Equation 6.1 into Equation 6.8, we can obtain the final calculated  $\delta_i E(\mathbf{x})$ ,

$$\begin{aligned} \delta_i E(\mathbf{x}) = & -2 \int_{\Omega} g(I_0(\mathbf{x}) - I_1(f(\mathbf{x}))) \nabla I_1(f(\mathbf{x})) \phi_i(\mathbf{x}) \mathbf{d}\Omega \\ & + \lambda_1 \int_{\Omega} (2f_{,u}(\mathbf{x}) \phi_{i,u}(\mathbf{x}) + 2f_{,v}(\mathbf{x}) \phi_{i,v}(\mathbf{x})) \mathbf{d}\Omega \\ & + \lambda_2 \int_{\Omega} (2f_{,u}(\mathbf{x}) \|f_{,v}(\mathbf{x})\|^2 \phi_{i,u}(\mathbf{x}) + 2\|f_{,u}(\mathbf{x})\|^2 f_{,v}(\mathbf{x}) \phi_{i,v}(\mathbf{x}) \\ & - 2\langle f_{,u}(\mathbf{x}), f_{,v}(\mathbf{x}) \rangle f_{,v}(\mathbf{x}) \phi_{i,u}(\mathbf{x}) \\ & - 2f_{,u}(\mathbf{x}) \langle f_{,v}(\mathbf{x}), f_{,u}(\mathbf{x}) \rangle \phi_{i,v}(\mathbf{x})) \mathbf{d}\Omega. \end{aligned} \quad (6.9)$$

**Remark 2.1.** From the differential energy formula derived in Equations 6.8 and 6.9, we can see that  $\phi_i(\mathbf{x})$ ,  $\phi_{i,u}(\mathbf{x})$  and  $\phi_{i,v}(\mathbf{x})$  come from the energy function itself. We do not introduce test functions by extra multiplications. In<sup>1</sup>, an additional test function  $\phi_i$  is introduced on the left-hand side, but not on the right hand-side. From the balance consideration, we decide to work on Equations 6.6 and 6.7 directly. In this way, we avoid solving a complicated FEM system. Since the energy functional is still in an integral formulation, we apply the most popular Gaussian quadrature rules to approximate it in each quadrilateral element. The numerical implementations of  $\delta E$  follow the FEM procedures<sup>66,117</sup>. In all our example studies, we apply 36 Gaussian points in each element.

**Remark 2.2.** In the presented work, we define a metric named the similarity ratio

### 6.3 Multilevel Technique

---

for our quantitative study,

$$Rs = 1 - \frac{\|I_0(\mathbf{x}) - I_1^k(f(\mathbf{x}))\|_{L^2}}{\|I_0(\mathbf{x}) - I_1(\mathbf{x})\|_{L^2}}, \quad (6.10)$$

where  $I_1^k(f(\mathbf{x}))$  is the deformed image after the  $k^{th}$  iteration step. The similarity ratio  $Rs$  is only applied for a healthy model, which means the differences between two images always decrease as the iteration steps go on. In other words, this number is bounded from 0% to 100%, and 100% is the ideal approximation. 0% is obtained when  $I_1^k(f(\mathbf{x})) \equiv I_1(\mathbf{x})$ , while 100% is obtained when  $I_1^k(f(\mathbf{x})) \equiv I_0(\mathbf{x})$ . However, in practice images usually contain a lot of complex information. In many cases, they cannot be represented mathematically. Therefore after some iteration steps, this ratio may not improve and remains stable. On one hand it reflects a limitation of image registration models, and on the other hand it is due to the complexities of image registration problems.

Another commonly used metric is called the mean square difference (MSD) measure, which is also used in<sup>1</sup>. The general form can be written as<sup>1</sup>,

$$MSD = \frac{1}{w \cdot h} \sum_{i=1}^w \sum_{j=1}^h (I_0(\mathbf{x}_{ij}) - I_1^k(f(\mathbf{x}_{ij})))^2,$$

where  $\{\mathbf{x}_{ij}\}_{i=1,j=1}^{w,h}$  are pixel points.

### 6.3 Multilevel Technique

In the previous section, we have introduced our dynamic mathematical model. In this section, this model will be enhanced by a multilevel technique. From the theoretical point of view, the more basis functions are applied, the better approximation is obtained. However, the large number of basis functions causes more time spent for computation. During our practical testing, we notice that generally large deformations happen at the initial iteration steps. Since our model has the dynamic property, we

### 6.3 Multilevel Technique

---

therefore are able to make it more flexible, stable and efficient. This development is carried out by enhancing the dynamic model with a multilevel technique. We use coarser levels in the first iteration steps. Despite the drawback of less accuracy, the coarser-level solving systems are able to handle large deformations fast and efficiently. However, the coarser levels can not be used for a long time. Errors will blow up after certain iteration steps. To prevent this problem, we switch to using finer levels. Continuing the previous iterations, the finer levels are able to catch more detailed information, remedying the accuracy problem and keeping the model in an active and efficient manner. We simply define an expression to illustrate our multilevel technique,

$$\underbrace{I_1^1(\mathbf{x}) \rightsquigarrow I_1^1(f_1^1(\mathbf{x})) \rightsquigarrow I_1^2(f_1^2(\mathbf{x})) \rightsquigarrow \dots \rightsquigarrow I_1^k(f_1^k(\mathbf{x})) \rightsquigarrow}_{\text{Level 1}} \underbrace{I_2^1(f_2^1(\mathbf{x})) \rightsquigarrow \dots \rightsquigarrow I_2^k(f_2^k(\mathbf{x})) \rightsquigarrow}_{\text{Level 2}} \underbrace{\dots \rightsquigarrow}_{\text{Level 3} \dots \text{Level } z-1} \underbrace{I_z^1(f_z^1(\mathbf{x})) \rightsquigarrow I_z^2(f_z^2(\mathbf{x})) \rightsquigarrow \dots \rightsquigarrow I_z^k(f_z^k(\mathbf{x})) \rightsquigarrow}_{\text{Level } z}, \quad (6.11)$$

where  $I_1^1(\mathbf{x}) = I_1(\mathbf{x})$  is initialized by the reference image,  $I_l^k(f_l^k(\mathbf{x}))$  stands for the registration results at the  $k^{\text{th}}$  iteration step at the  $l^{\text{th}}$  level based on the corresponding spatial transformation. The detailed algorithm is explained as follows.

#### Algorithm 3.1

The algorithm starts from two given images, the reference image  $I_1(\mathbf{x})$  and the target image  $I_0(\mathbf{x})$ . We want to find a spatial transformation  $f(\mathbf{x})$ , aligning  $I_1(\mathbf{x})$  with  $I_0(\mathbf{x})$  such that  $I_1(f(\mathbf{x})) \approx I_0(\mathbf{x})$ . We divide the registration process into a multilevel procedure, using coarser levels for large deformation and finer levels for more detailed deformation.

- 1) Based on the chosen initial control points, we build the initial spatial transformation  $f_1^1(\mathbf{x}) = \sum_{i=1}^{N_1} \mathbf{C}_{1i}^1 \phi_i(\mathbf{x})$ , then we perform an interpolation to obtain  $I_1^1(f_1^1(\mathbf{x}))$ .
- 2) For different levels  $l = 1, \dots, z$ , do the following
  - I) For different iterations  $s = 1, \dots, k$ , do the following

### 6.3 Multilevel Technique

---

- a) Substitute  $I_l^s(f_l^s(\mathbf{x}))$  into the differential energy Equation 6.9 to obtain  $\delta E_l^s$ .
- b) Substitute the control points  $\mathbf{C}_l^s$  and the calculated  $\delta E_l^s$  into the Equation 6.7, and then we obtain a new group of control points,  $\mathbf{C}_l^{s+1} = \mathbf{C}_l^s - \varepsilon \delta E_l^s$ .
- c) Using the newly calculated control points  $\mathbf{C}_l^{s+1}$ , we are able to construct the corresponding spatial transformation  $f_l^{s+1}(\mathbf{x}) = \sum_{i=1}^{N_l} \mathbf{C}_{li}^{s+1} \phi_i(\mathbf{x})$ . From the interpolation approximation, we obtain  $I_l^{s+1}(f_l^{s+1}(\mathbf{x}))$ .

II) Check the similarity ratio after each iteration. After certain iteration steps, if the similarity ratio increases slower or even decreasing. Then we increase the number of control points. Reset the spatial transformation, i.e.  $f_{l+1}^1(\mathbf{x}) = \sum_{i=1}^{N_{l+1}} \mathbf{C}_{(l+1)i}^{N_{l+1}} \phi_i(\mathbf{x})$ , and perform the interpolation operation to get  $I_{l+1}^1(f_{l+1}^1(\mathbf{x}))$ .

- 3) The multilevel procedure continues till the similarity ratio becomes stable without much improving.

**Remark 3.1:** If control grids overlap with each other and become too distorted, the registration will become less accurate and influence the further iterations. Such distortion problems commonly occur in coarse levels and also possibly occurs when applying a single fine level for the whole procedure. Our multilevel technique prevents this issue. From a practical point of view, the total iterations are simply equally divided between each level. This equal division may not be an optimal, but it is a straightforward choice which works reasonably well. The method will go to finer levels before distorted control grids occur. Since our algorithm is fast, one could find the time step  $\varepsilon$  through several tests for each problem.  $R_s$  directly reflects information during the running process, and can be used to choose a value for  $\varepsilon$  by observation.  $\varepsilon$  can also be obtained through direct calculations. In<sup>1</sup>, a method was proposed to approximate the varying  $\varepsilon_s$  where  $s$  stands for the  $s^{\text{th}}$  iteration by minimizing the energy functional. But this requires the calculation of the first and second differentials of the energy functional



### 6.3 Multilevel Technique

---

for each iteration.

In the following, we pick an example to illustrate this multilevel technique applied to our dynamic model. Figure 6.1(a-b) show the reference image and the target image. We want to register a rectangle from an ellipsoid. For this problem we apply four levels in total and each level contains 10 iteration steps. Figure 6.1(c-f) show the deformed mesh at each level. We can observe that large deformation happens at the beginning levels. The change at the finer levels becomes slower, and they are used to resolve more details. Figure 6.1(g-j) list the deformed images in the 10<sup>th</sup> iteration at the first level, the 20<sup>th</sup> iteration at the second level, the 30<sup>th</sup> iteration at the third level and the 40<sup>th</sup> iteration at the fourth level, respectively. We also give the inverse registration results for this example. Subfigures (k-n) collect the deformed meshes and the registration results. Furthermore, we compare several single level runs with the multilevel method, and the  $R_s$  for each iteration are shown in Figure 6.2. The plot lines in Figure 6.2 are the results of the registration from an ellipsoid to a rectangle as in Figure 6.1. The brown, gray, green and pink lines correspond to single level runs with  $40 \times 40$ ,  $50 \times 50$ ,  $60 \times 60$  and  $70 \times 70$  elements, respectively. As we can see, the brown line ( $40 \times 40$ ) decreases faster than the others. The gray line ( $50 \times 50$ ) decreases faster than the green and the pink lines. The pink line decreases at the slowest rate among the four single level cases. However, as the iterations increase, after around the 20<sup>th</sup> iteration, the evolution of the four methods shows an opposite pattern. The pink line is the most stable one among the four cases. And the brown line fluctuates and becomes unstable. In our proposed strategy, we combine the different levels. We simply divide the total iterations by four periods and each period is solved by one level and then move to the finer levels. The blue line in Figure 6.2 shows the corresponding results of the multiple level technique. As we can observe, the blue line decreases very fast at the beginning of the iteration and has stable properties in further iterations. The three solid green circles mark steps from one level to the next level.

**Remark 3.2:** The coarser level approximations are always fast and efficient, especially for large deformation at the beginning iteration steps. But they are not able to

### 6.3 Multilevel Technique

---

capture detailed deformation. This is why we move to finer levels gradually. During the calculation, the interpolation is necessarily used to obtain the information from the previous iteration. The interpolation procedures are also based on the cubic B-splines basis functions. In our study and testing, we usually do not have error accumulation issues because of using levels. In most cases, the error reduces as levels and iterations increase. However, after several levels, we notice the error decrease levels off. The similarity ratio depends on the complexity of the to-be-solved registration problems. In this paper, the number of levels and iterations are chosen empirically according to the given images. We will observe different final similarity ratios values in the numerical examples in section [6.4](#).

### 6.3 Multilevel Technique

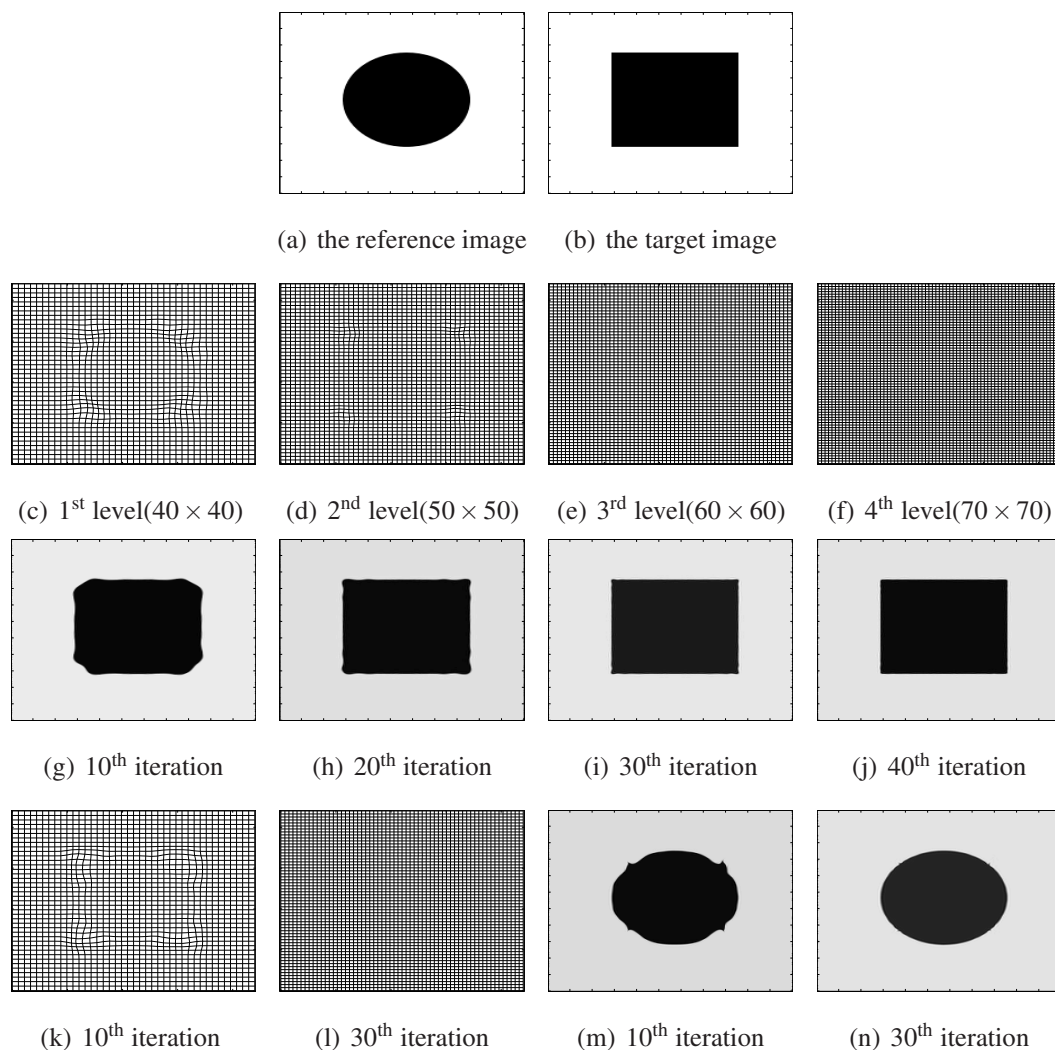


Figure 6.1: Registration results from an ellipsoid to a rectangle. The reference image and the target image are shown in (a) and (b). The deformed meshes are shown in (c-f). The corresponding deformed images are shown in (g-j). (k-n) are the inverse registration results from the rectangle to the ellipsoid.

## 6.4 Numerical Examples

---

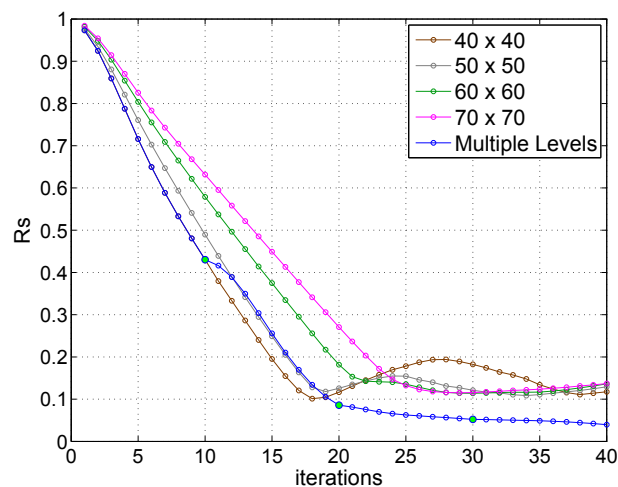


Figure 6.2: A comparison between the single level methods and the multilevel method.

## 6.4 Numerical Examples

In this section, we study more examples to test our dynamic multilevel registration method, including two pairs of artificial images (Figures 6.3 and 6.4), three pairs of MRI brain images (Figures 6.6, 6.7 and 6.8), and two pairs of CT cardiac images (Figures 6.10 and 6.11). The results shown in this section have been computed on a PC with an Intel Core i7-3740QM CPU running at 2.70 GHz and 8 GB of RAM. All the codes have been written in MATLAB. We show numerical registration results in different iteration steps at different levels. We also provide the similarity ratio values calculated for each problem. Table 6.1 shows the statistics data of all the tested examples.

## 6.4 Numerical Examples

Table 6.1: Similarity ratios ( $R_s$ ) and running time for all examples studied

Examples	Star-Gear	Disk-3Star	Brain 1	Brain 2	Brain 3	Heart 1	Heart 2
Image Size	$220 \times 220$	$220 \times 220$	$256 \times 256$	$256 \times 256$	$256 \times 256$	$512 \times 512$	$256 \times 256$
Total Levels	2	3	6	6	6	6	4
Iterations	5	20	15	10	10	10	10
Rs (%)							
Level 1	69.85	48.83	60.88	58.38	57.58	52.08	63.55
Level 2	90.00	79.71	71.93	73.35	68.64	65.86	71.51
Level 3	-	97.45	75.92	78.13	75.92	70.05	74.22
Level 4	-	-	78.89	80.03	78.99	74.11	76.15
Level 5	-	-	80.14	81.26	80.44	75.82	-
Level 6	-	-	81.31	81.71	81.12	77.05	-
Time (Seconds)							
Level 1	35.38	132.87	156.35	107.51	108.72	117.10	108.33
Level 2	58.20	210.74	230.31	160.25	157.97	169.39	160.82
Level 3	-	309.80	326.67	227.93	224.99	236.28	227.92
Level 4	-	-	446.89	314.18	309.82	322.05	312.96
Level 5	-	-	594.75	420.41	414.75	428.40	-
Level 6	-	-	790.97	549.05	545.35	557.61	-
Total	93.58	653.41	2545.90	1779.30	1761.60	1830.80	813.03

### 6.4.1 Artificial Images

Figure 6.3(a-b) shows the first pair of artificial images (star-gear). After applying our dynamic multilevel solving system as discussed in sections 6.2 and 6.3, we pick three sequential registration results and show them in (c-e). The total number of iterations is 10 for this example, which are equally divided into two levels, 5 steps for each level. (c) and (d) are results at the first level. In this level, we calculate  $44 \times 44$  control points. Then after the 5<sup>th</sup> iteration, we increase the number of elements and calculate  $54 \times 54$  control points at the second level approximation. The final numerical registration image is shown in (e). The image differences are shown in (f) and (g). In both level calculations, the time step  $\varepsilon$  is chosen to be 0.001. As we can observe, the starting star-shaped image is gradually registered into the gear-shaped image. The first level calculation is around 35.38s. The second level costs 58.20s.  $R_s$  is about 69.85%

## 6.4 Numerical Examples

---

after the first level calculation. With the second level,  $R_s$  is improved to be 90%.

Here we pick another artificial example (disk-3star) to show the large deformation study. Figures 6.4(a-b) show the reference image and the target image, they are very different from each other. We apply three levels to solve this large deformation registration with  $44 \times 44$ ,  $54 \times 54$  and  $64 \times 64$  control points, respectively. To accelerate the approximation at each level but prevent errors accumulation, we pick time step  $\varepsilon = 0.002$  for the first 10 iterations at a level, and  $\varepsilon = 0.001$  for the following 10 iterations at the same level. The three stars are shown clearly at the final iteration. From Table 6.1, we can observe that the third level promotes  $R_s$  more to achieve 97.45%, and our dynamic multilevel technique performs stably for large deformation cases. Due to its dynamic property, the registration process can be improved gradually. During the entire solving procedure, we are able to prevent inverted elements with the last two regularization terms in the energy function. Furthermore, the multilevel technique makes the dynamic model more efficient.

We also implemented the FEM-based method<sup>1</sup>, applied it to this large deformation problem, and compared with our method. Figure 6.4 shows the results using our dynamic multilevel method. (f) shows the initial differences between the reference image (a) and the target image (b). (g) shows the final differences between the registered result (e) and (b), it is obvious that the differences are significantly reduced as compared with (f). Figure 6.5 shows the registration results by the FEM-based method, which only works at a certain level during the whole registration process. Therefore, (a) is the same with our first level situation with applying  $44 \times 44$  control points. In this case,  $R_s$  becomes constant around the 90<sup>th</sup> iteration step with the value of 93.95%, but the time spent is already 583.26s. Then we calculated  $54 \times 54$  control points to solve the same problem as shown in (b).  $R_s$  levels off around the 90<sup>th</sup> iteration step again with an improved value of 97.22%. The total running time reaches 929.68s. (c) and (d) show the image differences. Comparing these two methods as shown in Table 6.2, our method is more efficient and accurate, yielding a  $R_s$  of 97.45% in 653.41s. In addition, we also compared these two methods in terms of the classical MSD measure. From

## 6.4 Numerical Examples

Table 6.2, we can observe that our method outperforms the method in <sup>1</sup> at the 3<sup>rd</sup> level for all the three testing examples.

Table 6.2: Comparison of Our Method with the Method in <sup>1</sup> in terms of similarity ratios ( $R_s$ ), mean square difference (MSD), and running time.

Example		Control Points	Iterations	$R_s$ (%)	MSD	Time (Seconds)
Disk-3Star	Method in <sup>1</sup>	44 × 44	total 90	93.95	7.8609e+02	583.26
		54 × 54	total 90	97.22	1.5792e+02	929.68
	Our Method	44 × 44 (Level 1)	0-20	48.83	5.3380e+04	132.87
		54 × 54 (Level 2)	20-40	79.71	8.7565e+03	210.74
64 × 64 (Level 3)		40-60	97.45	1.2974e+02	309.80	
						Total: 653.41
Brain 3	Method in <sup>1</sup>	54 × 54	stopped in 25 <sup>th</sup>	57.79	3.0676e+03	262.82
		64 × 64	stopped in 50 <sup>th</sup>	65.80	2.0004e+03	789.61
	Our Method	54 × 54 (Level 1)	0-10	57.58	3.0836e+03	108.72
		64 × 64 (Level 2)	10-20	68.64	1.6971e+03	157.97
		74 × 74 (Level 3)	20-30	75.92	1.0161e+03	224.99
		84 × 84 (Level 4)	30-40	78.99	7.7085e+02	309.82
		94 × 94 (Level 5)	40-50	80.44	6.5925e+02	414.75
104 × 104 (Level 6)	50-60	81.12	6.1496e+02	545.35		
						Total: 1761.60
Heart 2	Method in <sup>1</sup>	64 × 64	stopped in 80 <sup>th</sup>	73.11	1.9957e+07	1246.89
	Our Method	54 × 54 (Level 1)	0-10	63.55	3.6718e+07	108.33
		64 × 64 (Level 2)	10-20	71.51	2.2422e+07	160.82
		74 × 74 (Level 3)	20-30	74.22	1.8361e+07	227.92
84 × 84 (Level 4)	30-40	76.15	1.5711e+07	312.96		
						Total: 810.03

## 6.4 Numerical Examples

---

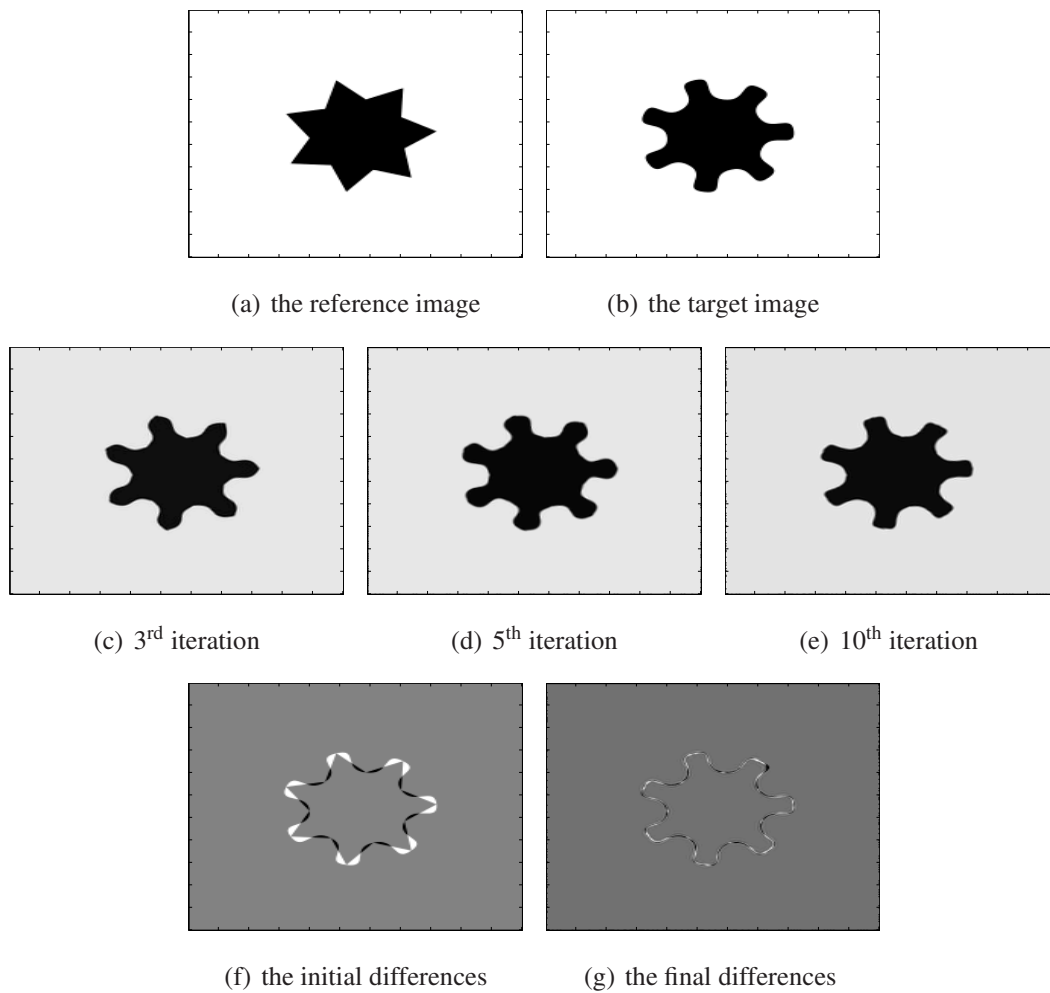


Figure 6.3: Register a gear-shaped object (b) from a star-shaped one (a). (c) and (d) are the first level registration results. (e) is the second level registration result. The initial differences between (a) and (b) and the final differences between (e) and (b) are shown in (f) and (g), respectively.



## 6.4 Numerical Examples

---

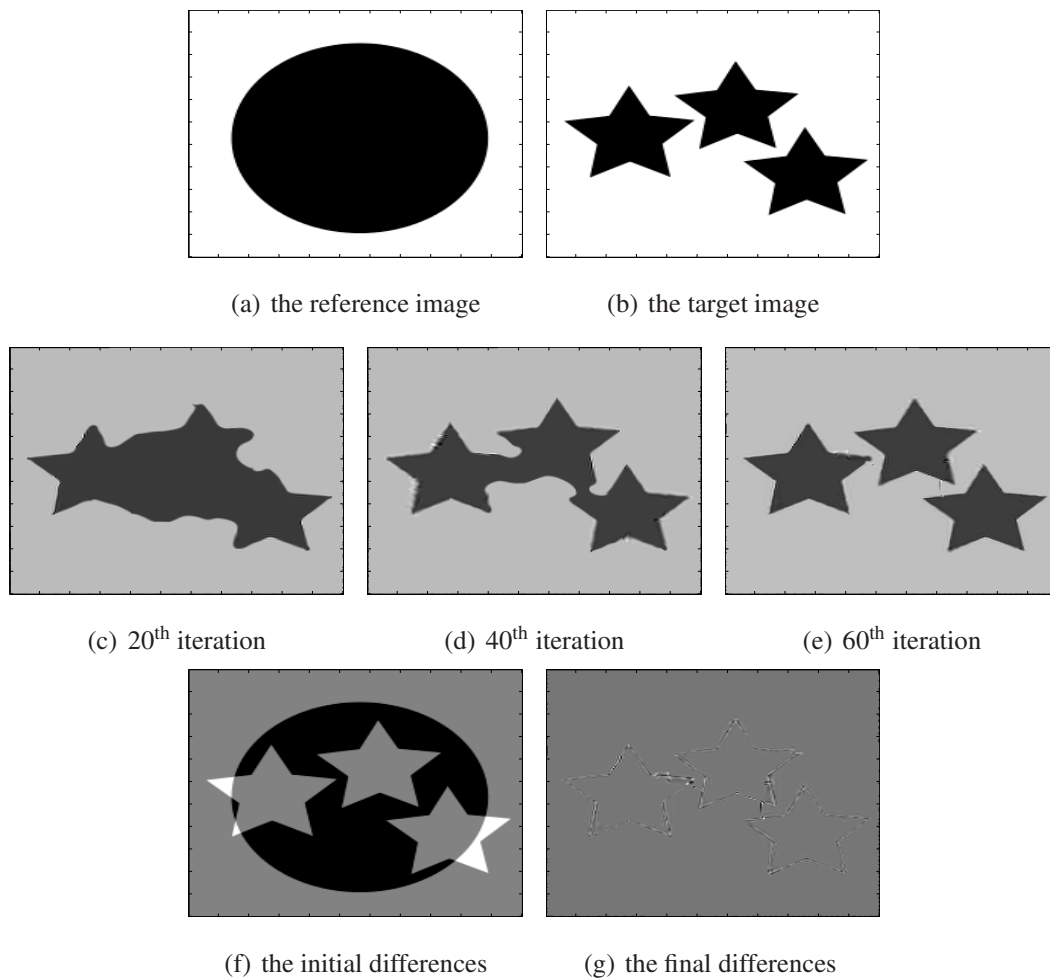


Figure 6.4: Register a three-star image (b) from a disk image (a). (c-e) are the registration results at the first level, the second level and the third level, respectively. (f) shows the initial differences between (a) and (b). (g) shows the final differences between our registration result (e) and the target image (b).

## 6.4 Numerical Examples

---

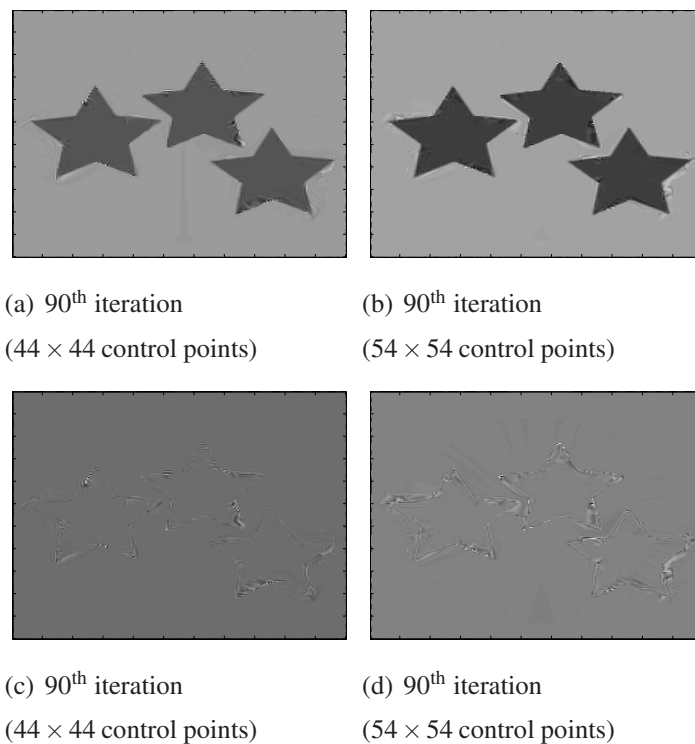


Figure 6.5: Registration results from method in <sup>1</sup>. (a) is the result after the 90<sup>th</sup> iteration steps with  $44 \times 44$  control points. (b) is the result after the 90<sup>th</sup> iteration step with  $54 \times 54$  control points. (c) shows the differences between (a) and the target image. (d) shows the differences between (b) and the target image.

### 6.4.2 MRI Brain Images

After two artificial images testing, we are ready to apply our model to deal with some real medical images. Usually, the medical images contain more complicated structures. Thus the registration from one medical image to another could be more challenging compared with the previous studies. Brain has the most complex structure in a human body, and MRI (magnetic resonance image) is the most common scanning technique for visualization. In our study, we choose three pairs of MRI images <sup>148</sup>, see Figures 6.6, 6.7 and 6.8.

## 6.4 Numerical Examples

---

The size of all brain images is  $256 \times 256$ , we use a total of six levels for each case. Except the first brain image example applying 15 iterations for each level, the other two examples consider 10 iterations for each level. For all the three examples, the number of control points for the six levels are  $54 \times 54$ ,  $64 \times 64$ ,  $74 \times 74$ ,  $84 \times 84$ ,  $94 \times 94$  and  $104 \times 104$ , respectively. As shown in Figures 6.6, 6.7 and 6.8, the reference images are gradually deformed in different iteration steps and levels, and the final results are very similar to their corresponding target images. In Figure 6.6, many complex cerebral cortex information are registered, see the comparison between (h) and (b) and also the image differences in (j). In Figure 6.7, the reference image (a) contains less tumor information than (b). After applying our registration process, in the final result (h), the tumor becomes more obvious and important. Meanwhile, all the important parts of the cerebral cortex of the target image (b) are registered clearly in our results. Figure 6.8(a-b) contain even more different information, such as the eyes, the nose region, the brain stem and many complex cortex. We apply our method to register (b) from (a). (c-h) are a sequence of numerical results. As we can observe, the final registration result (h) is substantially deformed, aligning with the target image (b). The similarity ratio  $R_s$  and the time spent of each example are collected in Table 6.1.

We also compare our method with the FEM-based method in<sup>1</sup>, see Table 6.2. Figures 6.8(i-j) show the initial differences and our final improved differences. Then we apply the model in<sup>1</sup> to deal with the same problem, see Figure 6.9. Considering the computational cost, we always first try less control points. The number of control points equals our first level case with  $54 \times 54$ . Because of the complication of this problem, the method gets stopped around the 25<sup>th</sup> iteration step with a  $R_s$  of only 57.79%. The time spent is around 262.82s. Thus we have to apply more control points to improve the approximation. By using  $64 \times 64$  control points,  $R_s$  stops improving in around the 50<sup>th</sup> iteration step with a  $R_s$  of 65.80%, but the time spent is already 789.61s. If we increase the number of control points to promote the model more,  $R_s$  will be improved but with even higher computational cost. From Table 6.2, we can observe that our method yields a much better  $R_s$  (81.12%) and MSD (614.9687) in a

## 6.4 Numerical Examples

---

shorter time. These comparisons show that our method is more suitable for dealing with complicated situations.

### 6.4.3 CT Heart Images

Heart is the center of the blood circulation system. It consists of four main chambers, connecting several major blood vessels. In our last subsection, we choose two pairs of CT heart images to test our model. In Figure 6.10(a-b), the main differences appear in the blood vessels surrounding the main heart body as well as the size and the shape of the chambers. Our registration results are shown in Figure 6.10(c-h). As we can observe, the 6<sup>th</sup> level result (h) is very similar to the target image (b), capturing well the structure of the chambers and the surrounding blood vessels. The image differences between (h) and (b) are shown in (j).

We perform the last comparison study between our method and the method in<sup>1</sup> only for the registration of the main heart structure region. The reference image and the target image are shown in Figure 6.11(a-b). Their differences are very obvious, see (c). For the method in<sup>1</sup>, we choose  $64 \times 64$  control points to solve this problem. The registration process gets stopped at around the 80<sup>th</sup> iteration step. The result and the improved differences are shown in (d) and (e). Then we carry out our method to deal with this problem. We use four levels with 10 iteration steps at each level. The final result of the 4<sup>th</sup> level is shown in (f) and its improved differences are shown in (g). From the results, we can observe that both methods can capture most features of the target image, but our result looks more smooth. From the comparison results in Table 6.2, we can also conclude that our method can save a lot of time (810.03s vs 1246.89s) with a better similarity ratio value (76.15% vs 73.11%).

## 6.4 Numerical Examples

---

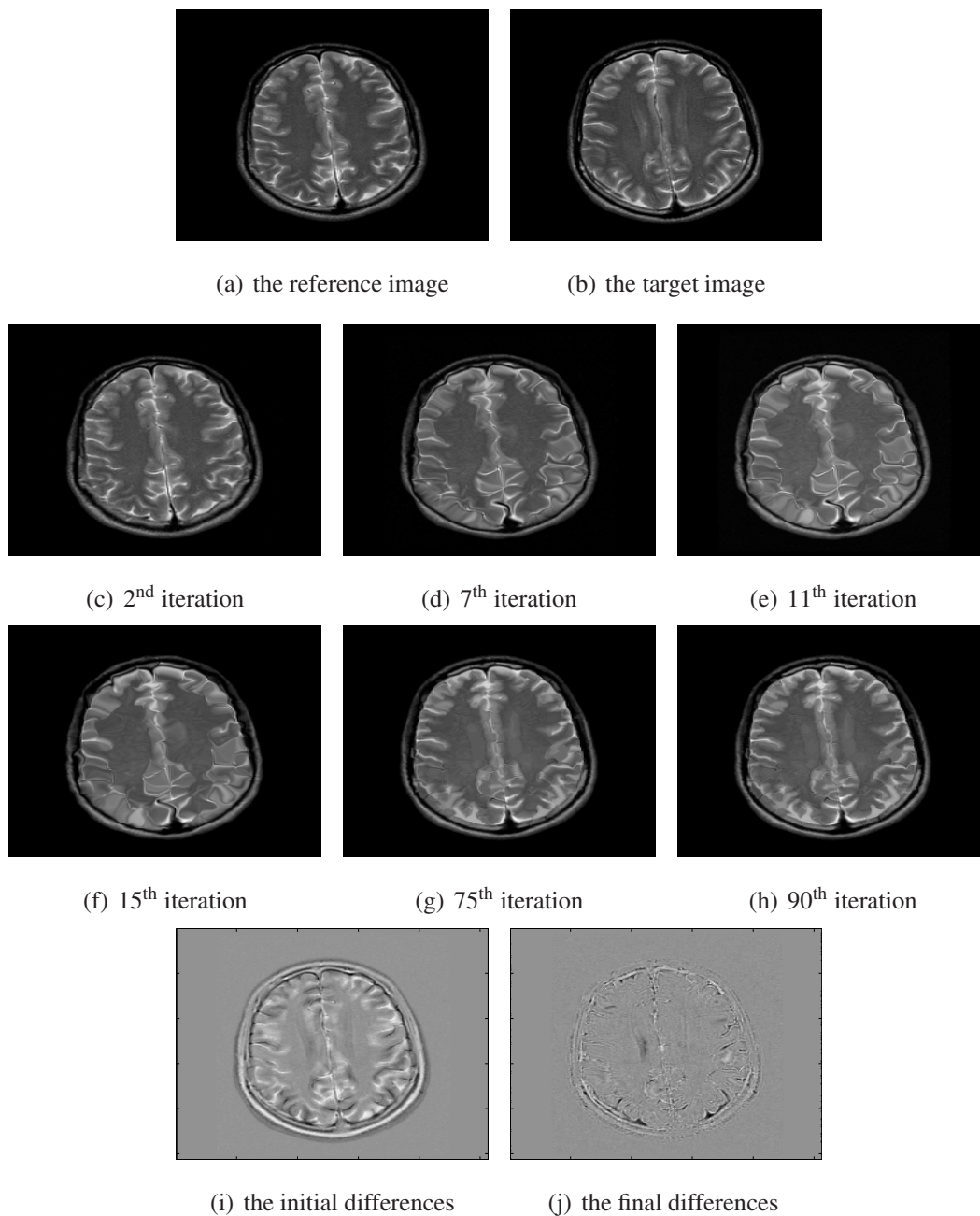


Figure 6.6: Brain image 1. Register the target image (b) from the reference image (a). (c-f) are the first level registration results with different iteration steps. (g) and (h) are the fifth and sixth level results, respectively. (i) shows the initial differences between (a) and (b), and (j) shows the differences between (h) and (b).

## 6.4 Numerical Examples

---

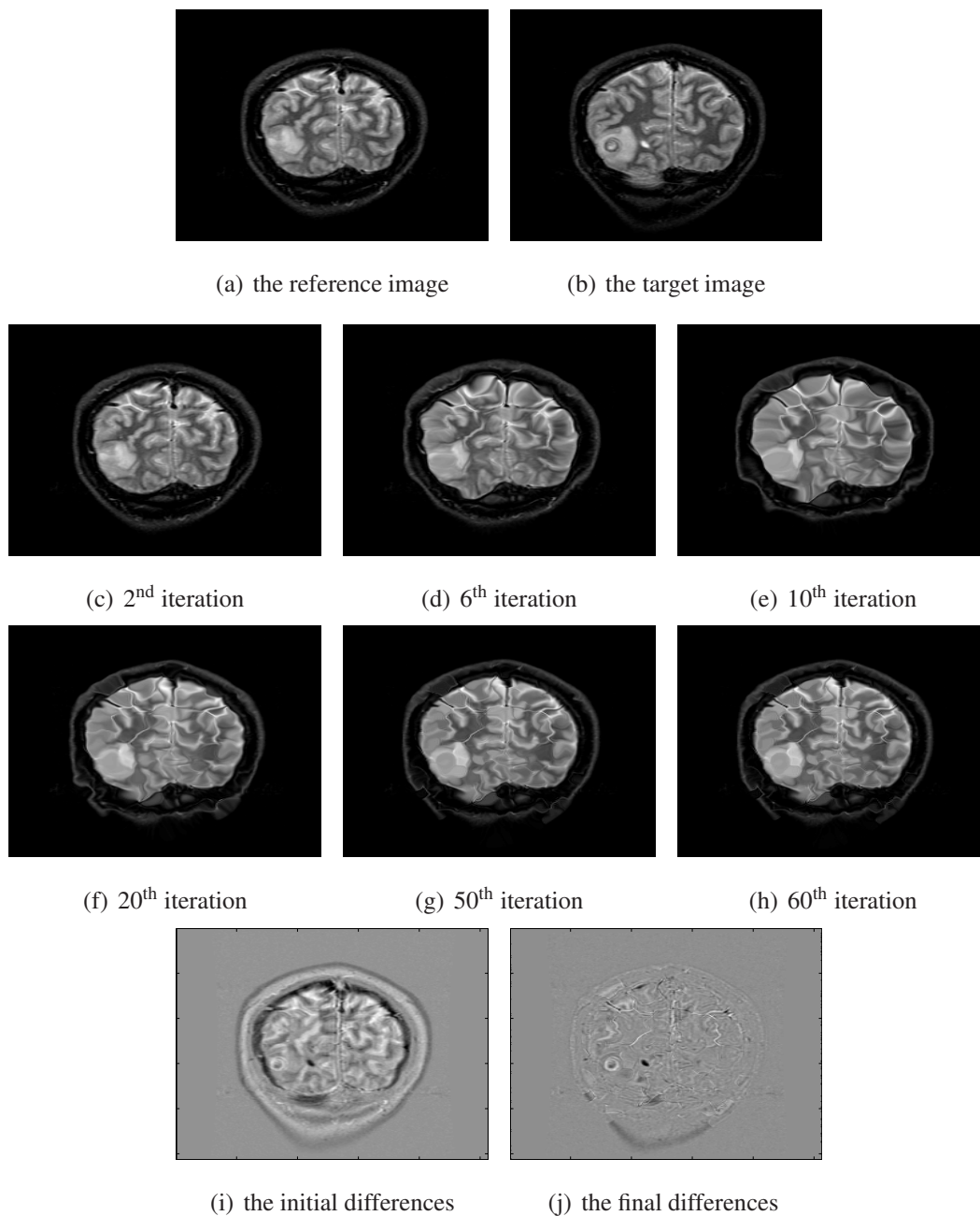


Figure 6.7: Brain image 2. Register the target image (b) from the reference image (a). (c-f) are the first level registration results with different iteration steps. (g) and (h) are the fifth and sixth level results, respectively. (i) shows the initial differences between (a) and (b), and (j) shows the differences between (h) and (b).

## 6.4 Numerical Examples

---

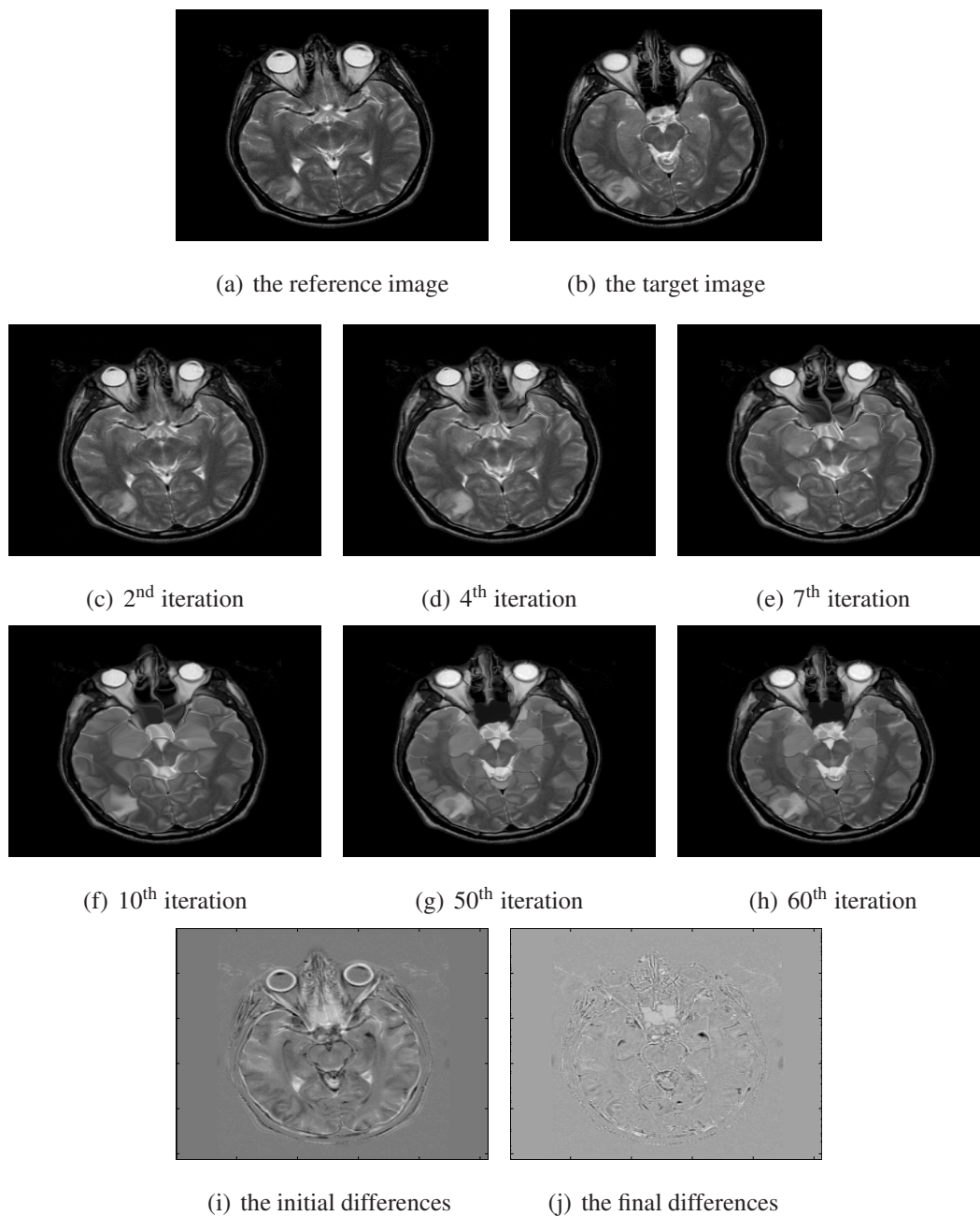


Figure 6.8: Brain image 3. Register the target image (b) from the reference image (a). (c-f) are the first level registration results with different iteration steps. (g) and (h) are the fifth and sixth level results, respectively. (i) shows the initial differences between (a) and (b), and (j) shows the differences between (h) and (b).

## 6.4 Numerical Examples

---

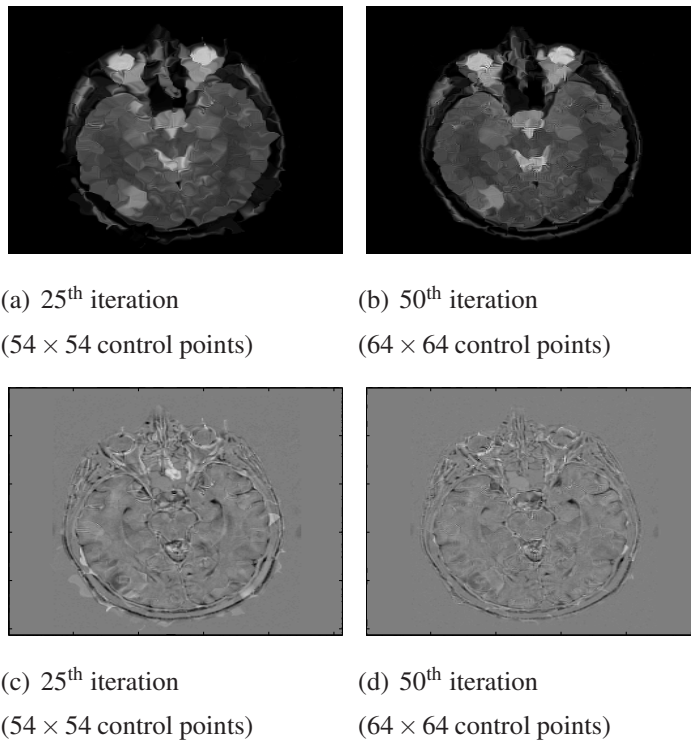


Figure 6.9: Registration results from method in <sup>1</sup>. (a) is the result after getting stopped in the 25<sup>th</sup> iteration step with  $54 \times 54$  control points. (b) is the result after getting stopped in the 50<sup>th</sup> iteration step with  $64 \times 64$  control points. (c) shows the differences between (a) and the target image. (d) shows the differences between (b) and the target image.



## 6.4 Numerical Examples

---

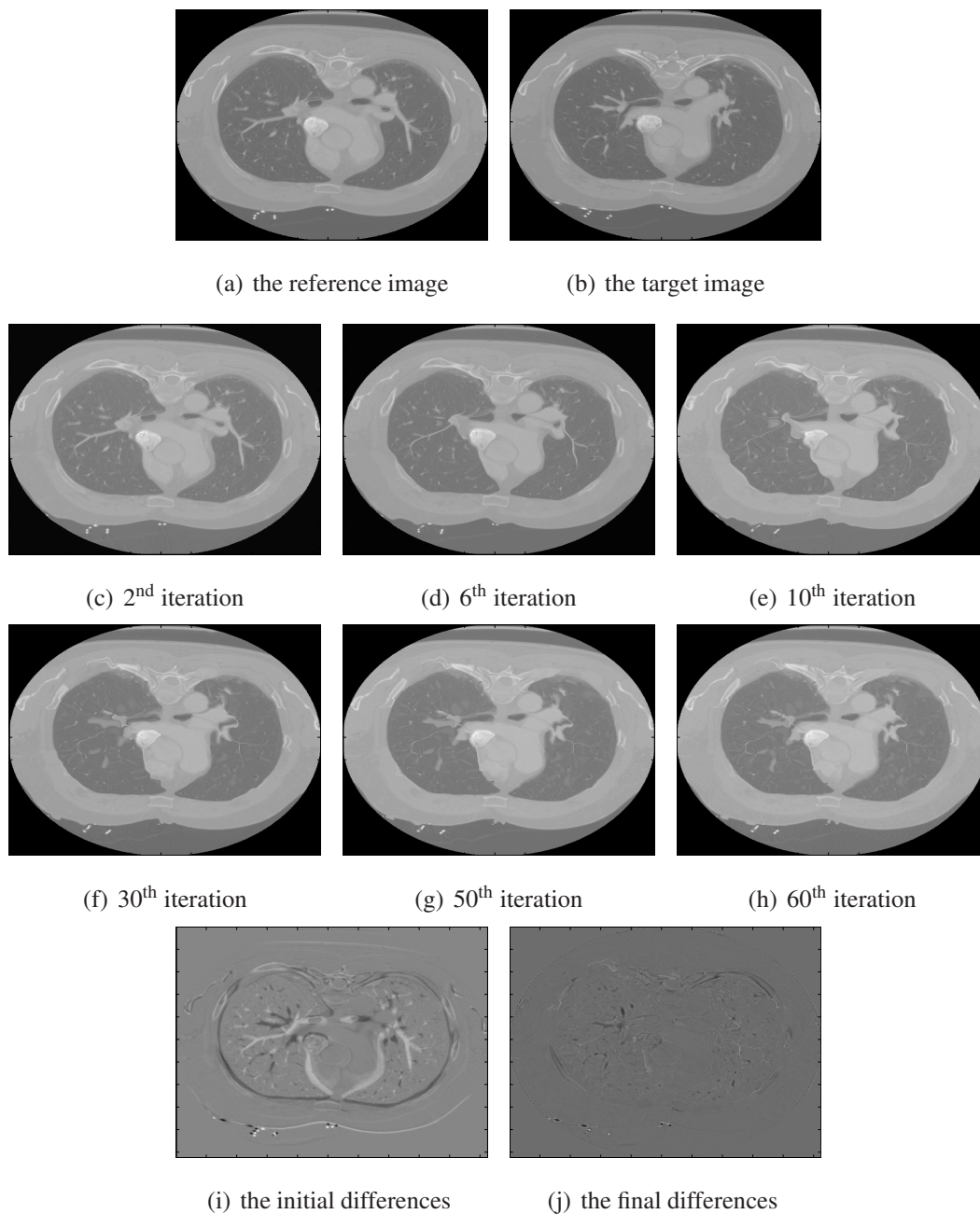


Figure 6.10: Cardiac image 1. The reference and target images are shown in (a) and (b). (c-e) are our registration results during the first level. (g) and (h) are solved after the fifth level and sixth level. (i) is the initial differences, and (j) shows the differences between our final solved result (h) and (b).

## 6.4 Numerical Examples

---

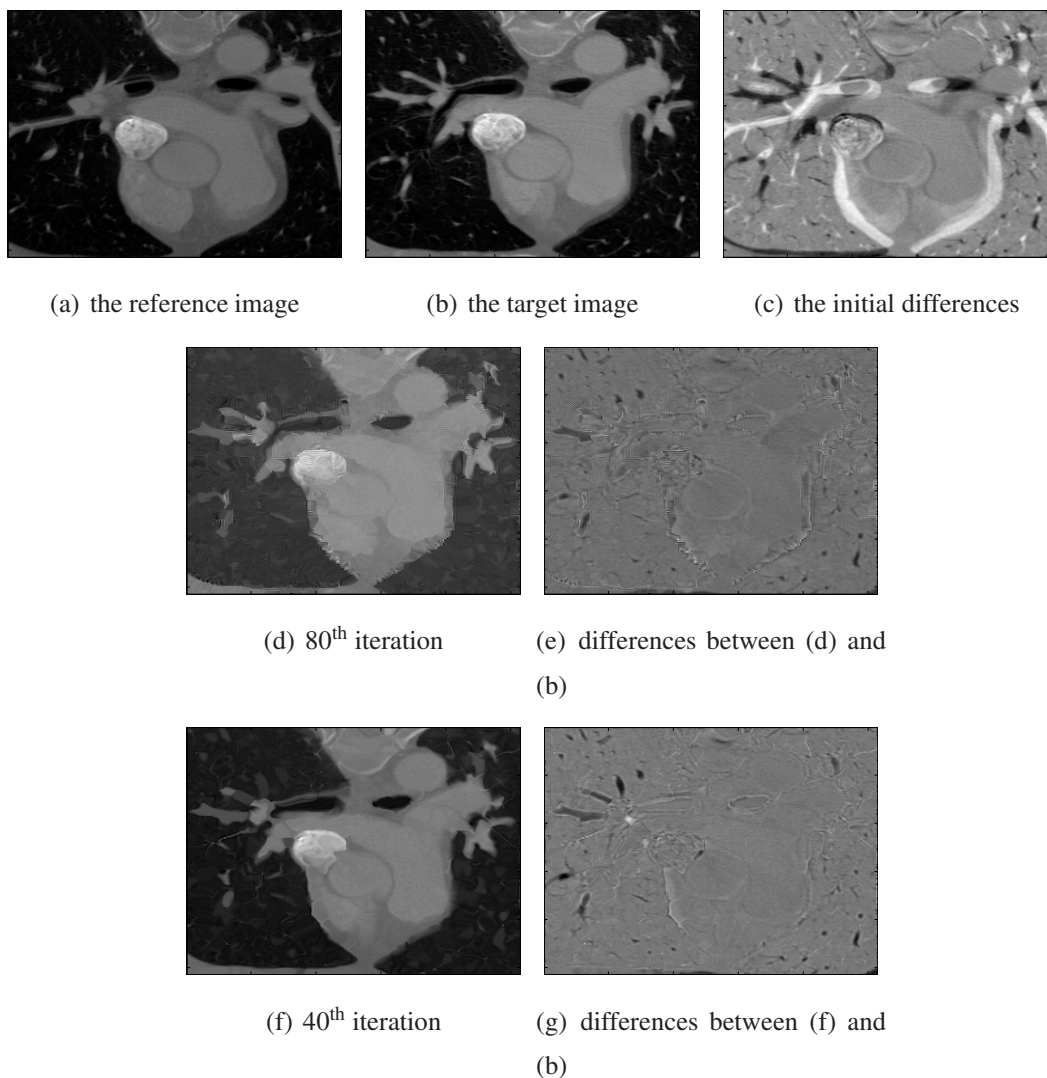


Figure 6.11: Cardiac image 2. The reference and the target images are shown in (a) and (b). (c) is the initial differences between (a) and (b). (d) is the registration result using the method in <sup>1</sup>. (e) shows the differences between (d) and (b). (f) is the registration result using our method, and (g) shows the differences between (f) and (b).

**Discussion:** In this section, we test our dynamic multilevel model by seven registration examples, two pairs of artificial images and five pairs of medical images. All the registration procedures follow a dynamic strategy, enhanced by a multilevel tech-

## 6.5 Conclusions

---

nique. For each example, the registration process is divided into several levels, and at each level we solve for a certain amount of control points dynamically in each iteration step. The number of control points is increased gradually. The method is flexible and efficient, especially for dealing with large deformation problems. The coarser levels are used for large deformation registration, and then gradually finer levels are used in order to capture detailed deformations. These effects can be observed from our registration results of each example. Since we use a small system with fewer control points in the beginning stage, the time spent is much less than carrying out a big solving system during the entire registration process. This is one of the important reasons why we save more time in our method than the method proposed in<sup>1</sup>. Another reason for faster performances of our method is that unlike FEM-based image registration methods, we avoid solving a big matrix system. We solve for control points directly. For image registration, the large matrix system usually brings some limitations or even obstacles. For example, building and solving the matrix system are time consuming and complicated. In addition, images usually contain many complicated information and even irregular features, all of these will influence the accuracy of the matrix-based registration methods. This also explains why some results of the method in<sup>1</sup> are not smooth and not accurate, see Figures 6.9 and 6.11. In our method, we still keep the right-hand side of the energy function with energy minimization terms, but we do not need to solve a large linear system. This is an important reason why our method is more stable and accurate.

## 6.5 Conclusions

We have developed a dynamic multilevel modeling technique for solving image registration problems. The mathematical model is also enhanced by a novel multilevel idea. The coarser levels focus on large deformations at the beginning iterations. The model works faster and more efficiently. The finer levels deal with detailed registration. The model works more accurately and stably. Our model is tested by two pairs of artificial

## 6.5 Conclusions

---

images and five pairs of real medical images. The numerical results show our model is successful in solving different cases. We also perform the quantitative study by calculating the similarity ratio for each example. The experimental results demonstrate that the proposed method is stable and efficient.

The accuracy usually depends on the complexity of the registered images. Therefore as part of our future work, we plan to study more to promote the accuracy. In addition, we also plan to apply our algorithm to other type images and extend our work to three dimensional problems.

# Chapter 7

## Conclusions

The thesis focuses on IGA, developing several techniques for modeling and analysis. It mainly contains four parts, including triangular B-spline basis functions, discontinuous problems modeling, superconvergent collocation method and image registration application. The following paragraphs summarize the main achievements of the current work and give several suggestions for the future work.

First, the thesis investigated a special triangular B-spline function which has gained success in geometry designs but has not often been applied in analysis. The particular definition of the triangular B-spline gives itself the flexibility property. However, it also leads to the instability problem during analysis, causing unexpected errors. In order to control this triangular B-spline, we applied the reproducing kernel approximation technique, calculating a correction term from the subset basis functions chosen by the compact support of the triangular B-spline, and adding the correction term to the triangular kernel function. Through the numerical comparisons, it can be seen that the instability effect is reduced significantly and the improved triangular B-spline is capable of solving PDEs with higher accuracy and almost optimal convergence rates. As part of our future work in this part, we plan to apply our algorithm to complicated problems, study singular and locking problems. In addition, we would also like to extend our technique to 3D based on tetrahedral meshes, the trivariate triangular B-spline.

Secondly, the thesis proposed several strategies to deal with material interface prob-

---

lems by the proposed extended isogeometric analysis method (XIGA). We incorporated the main idea of XFEM into IGA, introducing extra enrichment functions in the traditional IGA solving space. The considered enrichment functions include the ramp and the Moës enrichment functions, which are continuous functions themselves but with discontinuous derivatives. The resulting new basis space can model the discontinuity efficiently in weak discontinuous problems such as the material interfaces. For the curved material interface, in order to carry out the integration exactly, the inverse mapping and the curved triangular integral elements are applied. In the numerical tests, comparing with the traditional IGA, the accuracy was considerably improved and the optimal convergence rates were retrieved by our XIGA method. As part of our future work, we plan to apply our algorithm to more complex material interface and fracture problems. In addition, we would also like to extend our technique to three dimensions.

In the third part, the thesis has developed the IGA-C which is based on strong PDE forms. Comparing with the traditional IGA Galerkin weak form based strategy, the IGA-C is simple to be implemented with less solving procedure, and it does not need numerical integration calculation, so that the method has been demonstrated superior to the classical IGA-G in terms of the computational efficiency and this is an important advantage for higher polynomial degree approximation. However, a crucial point of IGA-C approach is the proper collocation points choosing. The tradition IGA-C choose the Greville abscissae as the collocation points. But it has been found the accuracy of IGA-C is not comparable to IGA-G and the results are suboptimal for the various norm evaluation except for the second order energy norm. In order to cope with these drawbacks, we proposed IGA-SC which is based on the superconvergent theorem and apply superconvergent points. The presented comparison tests show IGA-SC can obtain significantly better results, i.e. the accuracy can be close in accuracy to the IGA-G results and the convergent rates are optimal especially for odd polynomial degrees. In terms of computational cost, IGA-SC requires asymptotically  $2^d$  times more basis function evaluations than the Greville abscissae method, where  $d$  is the dimension of the space. For odd  $p$ , the method has much improved accuracy while for even  $p$  the improvement

---

is of a fixed constant factor. For higher polynomial degrees, IGA-SC also requires more finely refined meshes to work correctly. A more thorough theoretical investigation of collocation methods, as well as applications to more interesting engineering problems that include nonlinearities remains the subject of future work.

The last part of the thesis presented a novel dynamic multilevel technique for dealing with image registration. This part has explored the applications of B-spline basis on image process field. Image registration aims to align a target image from a reference image by defining a spatial transformation. In our work, the spatial transformation is the linear combination of B-spline functions with the to-be-determined control points. Our strategies start from a energy function which is the same as one used in a previous FEM based image registration. However, we solved the image registration in a different way. On one hand, we simplified the complicated solving procedure and work on the energy function directly, avoiding the matrix construction and the linear system solving. It resulted our method more simple and fast. On the other hand, we proposed a multilevel technique to prevent the instability occurring. It yields our method be more accurate and efficient. Our numerical tests include two artificial images and four real bio-medical MRI brain and CT heart images. All the numerical results show our technique can handle image registration well, especially for large deformation problems. The accuracy usually depends on the complexity of the two registered images. Therefore as part of our future work, we plan to study more to promote the accuracy. In addition, we also plan to apply our algorithm to other types of images.

# References

- [1] J. Leng, G. Xu, and Y. Zhang. Medical image interpolation based on multi-resolution registration. *Computers and Mathematics with Applications*, 66:1–18, 2013. [xvi](#), [xvii](#), [xviii](#), [xx](#), [5](#), [127](#), [128](#), [129](#), [130](#), [132](#), [138](#), [139](#), [142](#), [143](#), [144](#), [148](#), [150](#), [151](#)
- [2] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis*. John Wiley & Sons, Ltd, 2009. [1](#), [2](#), [10](#), [22](#), [101](#), [109](#), [126](#)
- [3] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005. [1](#), [2](#)
- [4] J. A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5296, 2006. [1](#)
- [5] J. A. Evans, Y. Bazilevs, I. Babuška, and T.J.R. Hughes. n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198(21-26):1726–1741, 2009. [1](#)
- [6] T.J.R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-



## REFERENCES

---

- method finite elements with k-method NURBS. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4104–4124, 2008. [1](#), [2](#)
- [7] A. Dede, L. Quarteroni. Isogeometric analysis for second order partial differential equations on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 284:807–834, 2015. [1](#)
- [8] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, and T. J. R. Hughes. Rubustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199:357–373, 2010. [1](#)
- [9] L. De Lorenzis, I. Temizer, P. Wriggers, and G. Zavarise. A large deformation frictional contact formulation using NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 87:1278–1300, 2011. [1](#)
- [10] L. De Lorenzis, P. Wriggers, and G. Zavarise. A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the augmented Lagrangian method. *Computational Mechanics*, 49(1):1–20, 2012. [1](#)
- [11] C. J. Corbett and R. A. Sauer. Three-dimensional isogeometrically enriched finite elements for frictional contact and mixed-mode debonding. *Computer Methods in Applied Mechanics and Engineering*, 284:781–806, 2015. [1](#)
- [12] D. J. Benson, Y. Bazilevs, M. C. Hsu, and T.J.R. Hughes. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 5–8:276–289, 2010. [2](#)
- [13] D. J. Benson, Y. Bazilevs, E. De Luycker, M. C. Hsu, M. A. Scott, T. J. R. Hughes, and T. Belytschko. A generallized finite element formulation for arbitrary basis functions: From isogeometric analysis to XFEM. *International Journal for Numerical Methods in Engineering*, 83:765–785, 2010. [2](#)

## REFERENCES

---

- [14] D. J. Benson, S. Hartmann, Y. Bazilevs, M. C. Hsu, and T.J.R. Hughes. Blended isogeometric shells. *Computer Methods in Applied Mechanics and Engineering*, 255:133–146, 2013. [2](#)
- [15] N. Valizadeh, S. Natarajan, O. A. Gonzalez-Estrada, T. Rabczuk, T. Quoc Bui, and S. P. A. Bordas. NURBS-based finite element analysis of functionally graded plates: Static bending, vibration, buckling and flutter. *Composite Structures*, 99:309–326, 2013. [2](#)
- [16] X. Deng, A. Korobenko, J. Yan, and Y. Bazilevs. Isogeometric analysis of continuum damage in rotation-free composite shells. *Computer Methods in Applied Mechanics and Engineering*, 284:349–372, 2015. [2](#)
- [17] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007. [2](#)
- [18] Y. Bazilevs, V. M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008. [2](#)
- [19] Y. Bazilevs, M. C. Hsu, and M. A. Scott. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249–252:28–41, 2012. [2](#)
- [20] D. Kamensky, M. C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015. [2](#)

## REFERENCES

---

- [21] J. A. Cottrell, T. J. R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007. [2](#)
- [22] H. Gomez, V. M. Calo, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197:4333–4352, 2008. [2](#)
- [23] H. Gomez, T. J. R. Hughes, X. Nogueira, and V. M. Calo. Isogeometric analysis of the isothermal Navier-Stokes-Korteweg equations. *Computer Methods in Applied Mechanics and Engineering*, 199(25-28):1828–1840, 2010. [2](#)
- [24] H. Gomez, A. Reali, and G. Sangalli. Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal of Computational Physics*, 262:153–171, 2014. [2](#)
- [25] D. Schillinger, M. J. Borden, and H. K. Stolarski. Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284:583–610, 2015. [2](#)
- [26] A. P. Nagy, M. M. Abdalla, and Z. Gurdal. On the variational formulation of stress constraints in isogeometric design. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2687–2696, 2010. [2](#)
- [27] X. Qian. Full analytical sensitivities in nurbs based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199(29-32):2059–2071, 2010. [2](#)
- [28] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976–2988, 2008. [2](#)

## REFERENCES

---

- [29] A. Buffa, G. Sangalli, and R. Vazquez. Isogeometric analysis in electromagnetics: B-splines approximation. *Computer Methods in Applied Mechanics and Engineering*, 17–20:1143–1152, 2010. [2](#)
- [30] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. J. R. Hughes, and T. W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013. [2](#)
- [31] M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213:206–222, 2012. [2](#)
- [32] M. A. Scott, M. J. Borden, C. V. Vehoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88:126–156, 2011. [2](#)
- [33] 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. *Computer Methods in Applied Mechanics and Engineering*, 284:1–20, 2015. [2](#)
- [34] T. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(22):477–484, 2003. [2](#)
- [35] T. Sederberg, D. L. Cardon, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 22(23):276–283, 2004. [2](#)
- [36] <http://tsplines.com>. [2](#)
- [37] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, 20(11):2075–2107, 2010. [2](#), [4](#), [5](#), [88](#), [94](#), [97](#)

## REFERENCES

---

- [38] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Iso-geometric collocation for elastostatics and explicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 249–252:2–14, 2012. [2](#), [4](#), [93](#)
- [39] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, and T.J.R. Hughes. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering*, 267:170–232, 2013. [2](#), [4](#), [5](#), [113](#), [114](#), [116](#), [118](#)
- [40] C. Anitescu, Y. Jia, Y. Zhang, and Rabczuk. T. An isogeometric collocation method using superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 284:1073–1097, 2015. [2](#)
- [41] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45:601–620, 1999. [2](#), [3](#)
- [42] T. P. Fries. A corrected XFEM approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*, 75:503–532, 2007. [2](#), [63](#), [72](#)
- [43] T. Belytschko, N. Moës, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 50:993–1013, 2001. [2](#)
- [44] A. Menk and S. P. A. Bordas. Crack growth calculations in solder joints based on microstructural phenomena with X-FEM. *Computational Materials Science*, 50:1145–1156, 2011. [2](#)
- [45] A. Menk and S. P. A. Bordas. Numerically determined enrichment functions for the extended finite element method and applications to bi-material anisotropic fracture and polycrystals. *International Journal for Numerical Methods in Engineering*, 83:805–828, 2010. [2](#)

## REFERENCES

---

- [46] T. Rabczuk, S. P. A. Bordas, and G. Zi. On three-dimensional modelling of crack growth using partition of unity methods. *Computers and Structures*, 88:1391–1411, 2010. [2](#)
- [47] E. De Luycker, D. J. Benson, T. Belytschko, Y. Bazilevs, and M. C. Hsu. X-FEM in isogeometric analysis for linear fracture mechanics. *International Journal for Numerical Methods in Engineering*, 87:541–565, 2011. [2](#), [4](#), [60](#)
- [48] S. S. Ghorashi, N. Valizadeh, and S. Mohammadi. Extended isogeometric analysis for simulation of stationary and propagating cracks. *International Journal for Numerical Methods in Engineering*, 89:1069–1101, 2012. [2](#), [4](#), [60](#)
- [49] Y. Jia, C. Anitescu, S. S. Ghorashi, and T. Rabczuk. Extended isogeometric analysis for material interface problems. *IMA Journal of Applied Mathematics Oxford Journals*, page DOI: 10.1093/imamat/hxu004, 2014. [2](#)
- [50] D. Schillinger, S. J. Hossain, and T.J.R. Hughes. Reduced Bezier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45, 2014. [2](#), [4](#), [121](#)
- [51] C. Adam, T. J. R. Hughes, S. Bouabdallah, M. Zarroug, and H. Maitournam. Selective and reduced numerical integrations for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:732–761, 2015. [2](#)
- [52] W. Dahmen, C.A. Micchelli, and H.-P. Seidel. Blossoming begets B-splines built better by B-patches. *Mathematics of Computation*, 59:97–115, 1992. [3](#), [37](#)
- [53] G. Xu, G. Z. Wang, and X. D. Chen. Free form deformation with rational DMS-spline volumes. *Journal of Computer Science and Technology*, 23:862–873, 2008. [3](#)

## REFERENCES

---

- [54] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999. 3
- [55] J. M. Melenk and I. Babůska. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):289–314, 1996. 4, 60
- [56] Y. Cai, X. Zhuang, and H. Zhu. International journal of computational methods. *Computational Mechanics*, 10(5):DOI: 10.1142/S021987621350028X, 2013. 4
- [57] T. Rabczuk and T. Belytschko. Crack particles: a simplified meshfree method for arbitrary evolving cracks. *International Journal for Numerical Methods in Engineering*, 61:2316–2343, 2004. 4
- [58] T. Rabczuk and T. Belytschko. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering*, 196:2777–2799, 2007. 4
- [59] X. Y. Zhuang, C. E. Augarde, and K. M. Mathisen. Fracture modeling using meshless methods and level sets in 3D: Framework and modeling. *Int. J. Numer. Meth. Engng.*, 92:969–998, 2012. 4
- [60] T. Rabczuk and P. Areias. A meshfree thin shell for arbitrary evolving cracks based on an external enrichment. *Computer Modeling in Engineering and Sciences*, 16(2):115–130, 2006. 4
- [61] T. Rabczuk, P.M.A. Areias, and T. Belytschko. A meshfree thin shell method for non-linear dynamic fracture. *International Journal for Numerical Methods in Engineering*, 72(5):524–548, 2007. 4
- [62] T. Rabczuk, R. Gracie, J-H Song, and T. Belytschko. Immersed particle method for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 81:48–71, 2010. 4

## REFERENCES

---

- [63] 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. *Computer Methods in Applied Mechanics and Engineering*, 249252(0):15 – 27, 2012. [4](#)
- [64] R. Cools. Monomial cubature rules since “ Stroud ”: A compilation - part 2. *Journal of Computational and Applied Mathematics*, 112(1-2):21–27, 1999. [4](#)
- [65] R. Cools and P. Rabinowitz. Monomial cubature rules since “ Stroud ”: A compilation. *Journal of Computational and Applied Mathematics*, 48:309–326, 1993. [4](#)
- [66] T. J. R. Hughes. *The Finite Element Method Linear Static and Dynamic Finite Element Analysis*. Dover Publications, INC. Mineola, New York, 2000. [4](#), [44](#), [128](#), [129](#)
- [67] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for nurbs-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(58):301 – 313, 2010. [4](#)
- [68] H. Lin, Q. Hu, and Y. Xiong. Consistency and convergence properties of the isogeometric collocation method. *Computer Methods in Applied Mechanics and Engineering*, 267:471–486, 2013. [4](#)
- [69] L. Beirão da Veiga, C. Lovadina, and A. Reali. Avoiding shear locking for the Timonshenko beam problem via isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 241–244:38–51, 2012. [4](#)
- [70] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):326–376, 1992. [5](#), [124](#)
- [71] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003. [5](#), [124](#)



## REFERENCES

---

- [72] H. Lester and S. R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32:129–149, 1999. [5](#), [124](#)
- [73] B. McGregor. Automatic registration of images of pigmented skin lesions. *Pattern Recognition*, 31(6):805–817, 1998. [5](#), [124](#)
- [74] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. H. Kuhn. Landmark-based elastic registration using approximating thin-plate splines. *IEEE Transactions on Medical Imaging*, 20(6):526–534, 2001. [5](#), [124](#)
- [75] F. L. Bookstein and W. D.K. Green. A feature space for edges in images with landmarks. *Journal of Mathematical Imaging and Vision*, 3:231–261, 1993. [5](#), [124](#)
- [76] S. Sull and N. Ahuja. Integrated matching and segmentation of multiple features in two views. *Computer Vision and Image Understanding*, 62(3):279–297, 1995. [5](#), [124](#)
- [77] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18(2):99–119, 1996. [5](#), [124](#)
- [78] A. K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):267–278, 1996. [5](#), [124](#)
- [79] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. *IEEE Conf. Computer Vision and Pattern*, pages 194–201, 1994. [5](#), [124](#)
- [80] J. Kybic and M. Unser. Fast parametric elastic image registration. *IEEE Transactions on Image Processing*, 12(11):1427–1442, 2003. [5](#), [124](#), [126](#)
- [81] B. C. Vemuri, S. Huang, S. Sahni, C. M. Leonard, C. Mohr, R. Gilmore, and J. Fitzsimons. An efficient motion estimator with application to medical image registration. *Medical Image Analysis*, 2(1):79–98, 1998. [5](#), [124](#)

## REFERENCES

---

- [82] Y. Zhang, Y. Jing, X. Liang, G. Xu, and L. Dong. Dynamic lung modeling and tumor tracking using deformable image registration and geometric smoothing. *Tech Science Press*, 9(3):213–226, 2012. [5](#), [124](#)
- [83] Y. Bentoutou, N. Taleb, K. Kpalma, and J. Ronsin. An automatic image registration for applications in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9):2127–2137, 2005. [5](#), [124](#)
- [84] A. A. Goshtasby. *2-D and 3-D image registration for medical, remote sensing, and industrial application*. Wiley Press, 2005. [5](#), [124](#)
- [85] <http://szeliski.org/Book>. [5](#), [124](#)
- [86] W. K. Liu. An introduction to wavelet reproducing kernel particle methods. *USACM Bulletin*, 8:3–16, 1995. [6](#), [42](#), [43](#)
- [87] W. K. Liu and Y. Chen. Wavelet and multiple scale reproducing kernel methods. *International Journal for Numerical Methods in Fluids*, 21:901–933, 1995. [6](#), [42](#), [43](#)
- [88] W. K. Liu, S. Jun, S. Li, and T. Belytschko. Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering*, 38:1655–1679, 1995. [6](#), [42](#), [43](#), [45](#)
- [89] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal Numerical Methods in Fluids*, 20:1081–1106, 1995. [6](#), [42](#), [43](#), [45](#)
- [90] C. de Boor. *A practical guide to splines revised edition*. Applied Mathematical Sciences, 2001. [6](#), [92](#)
- [91] C. de Boor and B. Swartz. Collocation at Gaussian points. *SIAM Journal on Numerical Analysis*, 10(4):582–606, 1973. [6](#), [92](#)

## REFERENCES

---

- [92] L. Piegl and W. Tiller. *The NURBS BOOK*. Springer-Verlag, Berlin, 1997. [15](#), [28](#), [29](#), [30](#), [66](#), [68](#), [126](#)
- [93] G. Farin. *Curves and Surfaces for CAGD*. Morgan Kaufmann Publishers, 1999. [15](#), [37](#), [63](#), [127](#)
- [94] <http://encyclopediaofmath.org/index.php/Generalized.derivative>. [21](#)
- [95] R. F. Curtain. *Functional Analysis in Modern Applied Mathematics*. Academic Press INC. (London) Ltd., 1977. [22](#)
- [96] A. R. Forrest. Iterative interpolation and approximation by bézier polynomials. *CAD*, 22(9):527–537, 1990. [28](#)
- [97] G. G. Lorentz. *Bernstein Polynomials*. New York: Chelsea Publishing Co., 1986. [28](#)
- [98] C. de Falco, A. Reali, and R. Vázquez. GeoPDEs: a research tool for IsoGeometric Analysis of PDEs. *Advances in Engineering Software*, 42(12):1020–1034, 2011. [30](#)
- [99] H. P. Seidel. Polar forms and triangular B-spline surfaces. *Blossoming: The New Polar-Form Approach to Spline Curves and Surfaces, SIGGRAPH '91 Course Notes*, 26:8.1–8.52, 1991. [37](#), [38](#)
- [100] H. Qin and D. Terzopoulos. Triangular NURBS and their dynamic generalization. *Computer-Aided Geometric Design*, 14:325–347, 1997. [39](#), [40](#)
- [101] P. Fong and H. P. Seidel. Control points for multivariate B-spline surfaces over arbitrary triangulations. *Computer Graphics Forum*, 10:309–317, 1991. [40](#)
- [102] M. Franssen, R. C. Veltkamp, and W. Wesselink. Efficient evaluation of triangular B-spline surfaces. *Computer Aided Geometric Design*, 17:863–877, 2000. [40](#)

## REFERENCES

---

- [103] J. Cao, X. Li, G. Wang, and H. Qin. Surface reconstruction using bivariate simplex splines on Delaunay configurations. *Computer & Graphics*, 33:341–350, 2009. [40](#)
- [104] P. Fong and H.-P Seidel. An implementation of triangular B-spline surfaces over arbitrary triangulations. *Computer Aided Geometric Design*, 10:267–275, 1993. [40](#)
- [105] R. Peifle and H. P. Seidel. Fitting triangular B-splines to functional scattered data. *Computer Graphics Forum*, 15:15–23, 1996. [40](#)
- [106] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parameterization of computational domain in isogeometric analysis: Methods and comparison. *Computer Methods in Applied Mechanics and Engineering*, 200(23-24):2021–2031, 2011. [42](#)
- [107] A. Shaw and D. Roy. A NURBS-based error reproducing kernel method with applications in solid mechanics. *Computational Mechanics*, 40:127–148, 2007. [43](#), [44](#)
- [108] N. Sunilkumar and D. Roy. A smooth finite element method based on reproducing kernel DMS-splines. *Computer Modelling in Engineering & Sciences*, 1655:1–47, 2010. [43](#), [44](#)
- [109] S. Li and W. K. Liu. Moving least-square reproducing kernel methods. part II: Fourier analysis. *Computer Methods in Applied Mechanics and Engineering*, 139:159–194, 1996. [43](#)
- [110] S. Li, W. K. Liu, and T. Belytschko. Moving least-square reproducing kernel methods. part I: Methodology and convergence. *Computer Methods in Applied Mechanics and Engineering*, 139:113–154, 1997. [43](#)

## REFERENCES

---

- [111] S. Li, H. Lu, W. Han, W. K. Liu, and D. C. Simkins. Reproducing kernel element method part II: Globally conforming  $I^m/C^n$  hierarchies. *Computer Methods in Applied Mechanics and Engineering*, 193:953–957, 2004. [43](#)
- [112] W. K. Liu, W. Han, H. Lu, S. Li, and J. Cao. Reproducing kernel element method part I: Theoretical formulation. *Computer Methods in Applied Mechanics and Engineering*, 193:933–951, 2004. [43](#)
- [113] H. Lu, S. Li, D. C. Simkins, Jr., W. K. Liu, and J. Cao. Reproducing kernel element method part III: Generalized enrichment and applications. *Computer Methods in Applied Mechanics and Engineering*, 193:989–1011, 2004. [43](#)
- [114] D. C. Simkins, Jr., S. Li, H. Lu, and W. K. Liu. Reproducing kernel element method part IV: Globally compatible  $C^n (n \geq 1)$  triangular hierarchy. *Computer Methods in Applied Mechanics and Engineering*, 193:1013–1034, 2004. [43](#)
- [115] S. Hao, W. K. Liu, and T. Belytschko. Moving particle finite element method with global smoothness. *International Journal for Numerical Methods in Engineering*, 59:1007–1020, 2004. [43](#), [45](#)
- [116] W. Han and W. K. Liu. Flexible piecewise approximations based on partition of unity. *Advances in Computational Mathematics*, 23:191–199, 2005. [43](#)
- [117] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, Sixth Edition*. Elsevier Butterworth-Heinemann, 2005. [44](#), [128](#), [129](#)
- [118] I. Babuška and J. M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40:727–758, 1997. [60](#)
- [119] H. Ji, D. Chopp, and J. E. Dolbow. A hybrid extended finite element/level set method for modelling phase transformations. *International Journal for Numerical Methods in Engineering*, 54:1209–1233, 2002. [61](#), [85](#)

## REFERENCES

---

- [120] N. Moës, M. Cloirec, P. Cartraud, and J.-F. Remacle. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering*, 192:3163–3177, 2003. [62](#), [84](#)
- [121] O. C. Zienkiewicz. *The Finite Element Method*. London : McGraw-Hill, 1971. [70](#)
- [122] Eugene L. Wachspress. A rational basis for function approximation II. curved sides. *IMA Journal of Applied Mathematics*, 11:83–104, 1973. [70](#)
- [123] M. Zlamal. Curved elements in the finite element method I. *SIAM Journal on Numerical Analysis*, 10:229–240, 1973. [70](#)
- [124] M. Zlamal. The finite element method in domains with curved boundaries. *International Journal for Numerical Methods in Engineering*, 5:367–373, 1973. [70](#)
- [125] W. J. Gordon and C. A. Hall. Transfinite element methods: Blending-function interpolation over arbitrary curved element domains. *Numerische Mathematik*, 21:109–129, 1973. [70](#)
- [126] R. Sevilla and Fernández-Méndez. NURBS-shaped domains with applications to NURBS-enhanced FEM. *Finite Elements in Analysis and Design*, 47:1209–1220, 2011. [70](#)
- [127] R. Sevilla, Fernández-Méndez, and A. Huerta. NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering*, 76:56–83, 2008. [70](#)
- [128] R. Sevilla, Fernández-Méndez, and A. Huerta. NURBS-enhanced finite element method (NEFEM). a seamless bridge between CAD and FEM. *Archives of Computational Methods in Engineering*, 18:441–484, 2011. [70](#), [71](#)

## REFERENCES

---

- [129] R. Sevilla and Fernández-Méndez. Numerical integration over 2D NURBS-shaped domains with applications to NURBS-enhanced FEM. *Finite Elements in Analysis and Design*, 47:1209–1220, 2011. [71](#)
- [130] S. Mohammadi. *Extended finite element method for fracture analysis of structures*. Oxford-Blackwell, 2008. [72](#)
- [131] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992. [89](#)
- [132] I. Babuška, T. Strouboulis, C. S. Upadhyay, and S. K. Gangaraj. Computer-based proof of the existence of superconvergence points in the finite element method; superconvergence of the derivatives in finite element solutions of Laplace’s, Poisson’s, and the elasticity equations. *Numerical Methods for Partial Differential Equations*, 12(3):347–392, 1996. [89](#)
- [133] I. Babuška, U. Banerjee, and J. E. Osborn. Superconvergence in the generalized finite element method. *Numerische Mathematik*, 107(3):353–395, 2007. [89](#)
- [134] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, 1994. [92](#)
- [135] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982. [92](#)
- [136] X. Zhang, X. Liu, K. Song, and M. Lu. Least-squares collocation meshless method. *International Journal for Numerical Methods in Engineering*, 51:1089–1100, 2001. [92](#)

## REFERENCES

---

- [137] L. De Lorenzis, J. A. Evans, T.J.R Hughes, and A. Reali. Isogeometric collocation: Neumann boundary conditions and contact. *Computer Methods in Applied Mechanics and Engineering*, 284:21–54, 2015. [93](#), [105](#)
- [138] P. L. Gould. *Introduction to Linear Elasticity*. Springer-Verlag, 1999. [101](#)
- [139] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. [118](#)
- [140] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, and V.M. Calo. The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers. *Computer Methods in Applied Mechanics and Engineering*, 213-216(0):353 – 361, 2012. [119](#)
- [141] M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers. Robust and optimal multi-iterative techniques for IgA Galerkin linear systems. *Computer Methods in Applied Mechanics and Engineering*, 284(0):230 – 264, 2015. [119](#)
- [142] L. Beirão da Veiga, D. Cho, L.F. Pavarino, and S. Scacchi. Overlapping Schwarz preconditioners for isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 278(0):239 – 253, 2014. [119](#)
- [143] P. H. Schöneman. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. [124](#)
- [144] J. C. Gower and G. B. Dijksterhuis. *Procrustes Problems*. Oxford University Press, 2004. [124](#)
- [145] M. Unser. Splines: a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, 1999. [126](#)
- [146] M. Ferrant, S. K. Warfield, C. R. G. Guttmann, R. V. Mulkern, F. A. Jolesz, and R. Kikinis. 3D image matching using a finite element based elastic deforma-



## REFERENCES

---

- tion model. *Medical Image Computing and Computer-Assisted Intervention - MICCAI'99 Lecture Notes in Computer Science*, 1679:202–209, 1999. [127](#)
- [147] M. E. Sherif and S. Oudom. Non-rigid image registration using a hierarchical partition of unity finite element method. *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. [127](#)
- [148] <http://overcode.yak.net/15>. [142](#)

# Academic Curriculum Vitae

## Jia, Yue

Institute of Structural Mechanics

Bauhaus-Universität Weimar

Marienstrasse 15, 99423 Weimar, Germany

Email: yue.jia@uni-weimar.de

## Education

- Ph.D student: Institute of Structural Mechanics, Bauhaus-Universität Weimar, Germany, 2011-2015.
- Ph.D visiting student: Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, USA, 2013-2014.
- M.Sc: Master of Science, University of Nice-Sophia Antipolis, Nice, France, 2009-2011.
- M.Sc: Master of Engineering Mathematics, University of L'Aquila, L'Aquila, Italy, 2009-2011.
- B.S: Bachelor of Mathematic & Information and Computing Sciences, Weinan Teachers University, Weinan City, Shannxi Province, China, 2005-2009.

## Publications

1. Yue Jia, Yongjie Jessica Zhang, Gang Xu, Xiaoying, Zhuang and Timon Rabczuk *Reproducing kernel triangular B-spline-based FEM*

- for solving PDEs*. Computer Methods in Applied Mechanics and Engineering. 2013, 267, 342-358.
2. Yue Jia, Cosmin Anitescu, Seyed Shahram Ghorashi and Timon Rabczuk *Extended isogeometric analysis for material interface problem*. IMA Journal of Applied Mathematics, Oxford Journals, DOI: 10. 1093/ima-mat/hxu004. 2014, 1-26.
  3. Yue Jia, Yongjie Jessica Zhang and Timon Rabczuk *A novel dynamic multilevel technique for image registration*. Computers and Mathematics with Applications. 2015, 69(9), 909-925.
  4. Cosmin Anitescu, Yue Jia, Yongjie Jessica Zhang and Timon Rabczuk *An isogeometric collocation method using superconvergent points*. Computer Methods in Applied Mechanics and Engineering. 2015, 284, 1073-1097.
  5. Junhua Zhao, Jin-Wu Jiang, Yue Jia, Wanlin Guo and Timon Rabczuk *A theoretical analysis of cohesive energy between carbon nanotubes, graphene and substrates*. Carbon. 2013, 57, 108-119.
  6. Yancheng Zhang, Junhua Zhao, Yue Jia, Tarek Mabruki, Yadong Gong, Ning Wei and Timon Rabczuk *An analytical solution on the interface debonding for large diameter carbon nanotube-reinforced composite with functionally graded variation interphase*. Composite Structures. 2013, 104, 261-269.

## Conference and Workshop Presentations

1. “A novel dynamical multilevel technique for solving image registration,” Poster Presentation, *Sixth International Workshop on High-Order Finite Element and Isogeometric Methods (HOFEIM)*, 15<sup>th</sup>-18<sup>th</sup> July, 2014, Frauenchiemsee Island, Germany.

2. “A novel dynamical multilevel technique for solving image registration,” Presentation, *5<sup>th</sup> ITN INSIST workshop*, 15<sup>th</sup>-17<sup>th</sup> September, 2015, Cardiff, UK.
3. “An isogeometric collocation method using superconvergent points,” Presentation, *4<sup>th</sup> ITN INSIST workshop*, 26<sup>th</sup>-30<sup>th</sup> January, 2015, Strobl, Wolfgang, Austria.
4. “Reproducing kernel triangular B-spline-based FEM for solving PDEs,” Presentation, *3<sup>th</sup> ITN INSIST workshop*, 22<sup>nd</sup>-27<sup>th</sup> September, 2013, Weimar, Germany.
5. “Extended isogeometric analysis for material interface problem,” Presentation, *2<sup>nd</sup> ITN INSIST workshop*, 4<sup>th</sup>-10<sup>th</sup>, February, 2013, Valencian, Spain.