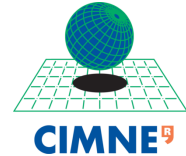
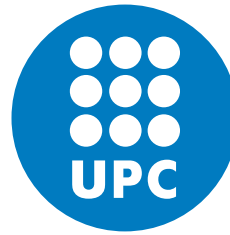
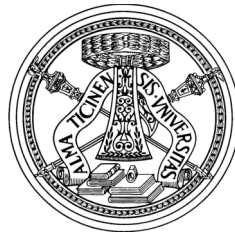




Erasmus
Mundus



Erasmus Mundus Joint Doctorate Programme
SIMULATION IN ENGINEERING AND
ENTREPRENEURSHIP DEVELOPMENT (SEED)



Simulation Tools for Biomechanical Applications with PGD-Based Reduced Order Models

by

Xi Zou

A Thesis Submitted to

DIPARTIMENTO DI INGEGNERIA CIVILE ED ARCHITETTURA
UNIVERSITÀ DEGLI STUDI DI PAVIA

and

DEPARTAMENT D'ENGINYERIA CIVIL I AMBIENTAL
UNIVERSITAT POLITÈCNICA DE CATALUNYA

in Partial Fulfillment of the Requirements for the Degree
Doctor of Philosophy

in

Computational Mechanics and Advanced Materials

Supervisors:

Prof. Ferdinando Auricchio (Università degli Studi di Pavia)

Prof. Pedro Díez (Universitat Politècnica de Catalunya)

Pavia & Barcelona

October 2017

To Wei

ABSTRACT

Simulation Tools for Biomechanical Applications with PGD-Based Reduced Order Models

Xi Zou

Numerical simulation tools are generally used in all modern engineering fields, especially those having difficulties in performing large number of practical experiments, such as biomechanics. Among the computational methods, Finite Element (FE) is an essential tool. Nowadays, the fast-growing computational techniques, from the upgrading hardware to the emerging of novel algorithm, have already enabled extensive applications in biomechanics, including mechanical analysis from musculoskeletal or cardiovascular system in macro scale to cell structures or tissue behaviours in micro scale. For applications that require fast response and/or multiple queries, Reduced Order Modelling (ROM) methods have been developed based on existing methods such as FE, and have eventually enabled real-time numerical simulation for a large variety of engineering problems.

In this thesis, several novel computational techniques are developed to explore the capability of Proper Generalised Decomposition (PGD), which is an important approach of ROM. To assess the usability of the PGD-based ROM for biomechanical applications, a real human femur bone is chosen to study its mechanical behaviour as an example. Standard image-based modelling procedure in biomechanics is performed to create an FE model which is then validated with *in vitro* experimental results.

As a major contribution, a non-intrusive scheme of the PGD framework is developed and implemented using commonly-used industrial software such as Matlab and Abaqus. It uses Abaqus as an external FE solver, which is called by in-house Matlab codes implementing the PGD algorithms. An example code is available at <https://github.com/xizou/NIPGD>. This scheme takes advantages of the maturity, robustness and availability of existing FE solvers, and demonstrates a great potential for being applied to industrial projects.

To solve parametrised partial differential equations with a parameter space subjected to physical or geometric constraints, a novel strategy is proposed. This strategy provides an approach that collects the most correlated parameters, and then separates them into 2D/3D spaces, instead of separating the parameter space into tensor products of 1D spaces in a Cartesian fashion as it is done in conventional PGD

framework.

Inspired by the fast-developing methods of isogeometric analysis, it is interesting to borrow the isogeometric idea to exploit the ways of discretising the parameter space inside the PGD framework. The high continuity of B-spline shape functions enables more accurate results for the computation of sensitivities with respect to the parameters. A classical mechanical problem is investigated with orthotropic materials in 2D, with the intention of further application in biomechanics.

In addition, an exploration of the generalisation of PGD to nonlinear problems in solid mechanics is presented as another main contribution. Following the large strain theory, Picard linearisation is used to establish a consistent PGD framework within total Lagrange formulation. As a preliminary example, the St.Venant-Kirchhoff constitutive model is adopted.

A practical example of the femur bone simulation is provided, the material parameters are obtained through an identification problem using the PGD vademecum, and in a further step, another PGD vademecum is generated for real-time simulation accounting for various loading locations.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratefulness to my supervisors, Prof Ferdinando Auricchio and Prof Pedro Díez, for their kind support and excellent supervision. I was always able to learn a lot from every discussion that we had.

Based on Università di Pavia, I am lucky to have spent more than two years in this historical institute. I'd like to thank Dr Michele Conti for his tutoring and frequent follow-up on the biomechanical problems. I'd like also to thank Prof Alessandro Reali, it was him who inspired my interest on isogeometric analysis and provided lots of instruction and support. It was a pleasure to have worked together with Rodrigo Romarowski, Margherita Coda, Dr John-Eric Dufour, Valentina Mercuri, Marco Fedele, Dr Elena Faggiano, Gianluca Alaimo, Stefania Marconi, Massimo Carraturo, Lorenzo Casagrande, Alberto Cattenone, Alice Finotello, Dr Simone Morganti, Sonia Padovan, Dr Giulia Scalet, and all my colleagues in CompMech group. I appreciate their warm companionship and selfless help.

This joint doctorate requires an approximately 12-month mobility to the secondary institute. I spent almost a year in Universitat Politècnica de Catalunya, Barcelona. I enjoyed this amazing city and the vivid life there. I'd like to thank all LaCáN people, especially Prof Antonio Huerta for his inspirations on computational mechanics, Dr Pablo Sáez for his patient support, and Dr Albert Costa, Luca Borchini, Simona Vermiglio, Dr Matteo Giacomini, et al. It is also notable that I stated a pleasant collaboration on PGD-DIC with Dr Jean-Charles Passieux during his visit to LaCáN.

This work is funded by European Education, Audiovisual and Culture Executive Agency (EACEA) of the European Commission under the *Erasmus Mundus* Joint Doctorate “Simulation in Engineering and Entrepreneurship Development” (SEED), FPA 2013-0043. I'd like also to thank all the SEED guys that we shared together our different experiences and feelings, especially Alberto Sibileau, Magnus Winter, Ceren Gürkan, Boyi Ye, Tuong Hoang, Karim El-Ghamrawy, Osama Ibrahim Ismail Hassan, Farhad Mani, et al.

Finally, I owe my thanks to my beloved family members who supported me during the unforgettable three years.

Contents

Abstract	v
Acknowledgments	vii
Contents	ix
List of Figures	xiii
List of Tables	xvii
List of Algorithms	xix
1 Introduction	1
1.1 Background	7
1.2 State of the art	8
1.2.1 Reduced order modelling with RB	9
1.2.2 Reduced order modelling with PGD	11
1.2.3 Application of reduced order models in biomechanics	14
1.3 Thesis objectives and chapter organisation	15
2 Image-based FE model generation and validation	19
2.1 From CT data to FE model	19
2.2 From grey scale data to material properties	23
2.3 Validation of the FE model	24
2.4 Summary	27
3 Non-intrusive PGD scheme	29
3.1 PGD formulation	31
3.1.1 PGD generalised weak form	31
3.1.2 PGD separated representation	32
3.1.3 Alternative direction iteration	33
3.2 FE discretisation	35
3.3 Non-intrusive scheme	36

3.3.1	Material properties as extra coordinates	37
3.3.2	Loading locations as extra coordinates	40
3.4	Matlab–Abaqus implementation	42
3.5	Numerical examples	44
3.5.1	Example I: A 1D problem with material properties as extra coordinates	44
3.5.2	Example II: A 1D model problem with loading locations as extra coordinates	46
3.5.3	Example III: A 2D problem with material properties as extra coordinates	49
3.6	Summary	51
4	PGD for constrained parameter space	53
4.1	Physical constraints	54
4.1.1	PGD formulation for Young modulus mapped from CT image	54
4.1.2	Example: A 2D problem with material properties as extra co- ordinates	58
4.2	Geometrical constraints	62
4.3	Summary	63
5	Online biomechanical application of PGD vademecum	67
5.1	Material parameter identification	68
5.2	Towards real-time simulation	71
5.3	Summary	73
6	NURBS shape function discretisation of parameter spaces	75
6.1	Constitutive parameters of 2D orthotropic materials	76
6.2	Decomposition of element stiffness matrix	77
6.3	PGD separated representation	79
6.4	Discretisation with B-Spline interpolation	79
6.4.1	B-spline discretisation in the IGA fashion	80
6.4.2	Fully discretised PGD formulation	84
6.5	Post-processing	84
6.5.1	Displacement	85
6.5.2	Derivatives with respect to parameters	85
6.6	Numerical examples	87
6.7	Summary	89
7	Nonlinear problems with PGD	95
7.1	A brief review of large strain theory	96
7.1.1	Large strain kinematics	96
7.1.2	Stress and equilibrium	97
7.2	PGD formulation with Picard linearisation	98

7.2.1	Separated representation	99
7.2.2	Picard linearisation	99
7.3	Application to St.Venant-Kirchhoff material with discretised formulation	101
7.3.1	Alternating direction iteration scheme	103
7.3.2	Discretised formulation for parametric modes	105
7.3.3	Discretised formulation for spatial modes	106
7.3.4	Discretised formulation of PGD alternative direction iteration scheme	109
7.4	Numerical example	111
7.4.1	Material properties	112
7.4.2	Mechanical model	112
7.4.3	SVD analysis	112
7.4.4	PGD results	114
7.5	Summary	116
8	Conclusions and future work	119
8.1	Conclusive remarks	119
8.2	Future works	121
	Bibliography	123
A	PGD formulation for static condensation	139
A.1	Variable Dirichlet boundary condition	139
A.2	Prescribed Dirichlet BC with variable material properties	141
A.3	Prescribed BC with variable material properties	143
B	PGD formulation for Young modulus and Poisson's ratio	145
B.1	Basics of plane stress elasticity	145
B.2	PGD separated representation	145
B.3	Decomposition of elemental stiffness matrix	146
B.3.1	Discretised formulation	147
C	Nonlinear dimensionality reduction for geometric parameters	149
C.1	Nonlinear dimensionality reduction with Isomap	149
C.2	Geometric shape analysis of proximal femur	152

List of Figures

1.1	Page of <i>De Motu Animalium</i> by Giovanni Alfonso Borelli in the 17th century.	2
1.2	SVD spectrum of an 667×500 image.	6
1.3	First 10 basis images reconstructed with $\sigma_i \mathbf{U}_i \otimes \mathbf{V}_i, (i = 1, 2, \dots, 10)$.	6
1.4	Reduction of an 667×500 image, reconstructed with $\sum_{i=1}^k \sigma_i \mathbf{U}_i \otimes \mathbf{V}_i, (k = 10, 50, 100, 500)$.	7
1.5	Anatomic terminologies of a right proximal femur, seen from the back. (Case courtesy of A.Prof Frank Gaillard, https://radiopaedia.org , rID: 7555).	8
2.1	<i>In vitro</i> experiment settings.	20
2.2	A CT slice example.	20
2.3	Segmentation with ITK-SNAP.	22
2.4	A mesh convergence study for the femur FE model. Mesh size is characterised by edge length of tetrahedrons, U denotes displacement of the loading point.	23
2.5	Spatial mesh of the human proximal femur model, illustrating locations with strain gauges attached for strain acquisition. Each location is attached two strain gauges in longitudinal and circumferential direction, respectively.	23
2.6	Comparison of Experimental (E) and simulated (S) strain results.	26
2.7	Young modulus contour. Left: CT image; Right: FE model.	27
2.8	Comparison of strains on Section SG1. Left: longitudinal strains; Right: circumferential strains. Solid line: homogeneous model; Dashed line: inhomogeneous model.	27
3.1	Flowchart of non-intrusive PGD implementation with material properties as extra coordinates.	45
3.2	Sketch of the 1D problem with material properties as extra coordinates.	46
3.3	Amplitude of each PGD mode for the 1D problem with material properties as extra coordinates.	46
3.4	Results of nodal displacement for the 1D problem with material properties as extra coordinates.	47

3.5	Sketch of the 1D problem with loading location as extra coordinates. . .	47
3.6	Comparison of PGD and SVD solutions. The modal amplitudes drastically drop after the 26th mode.	49
3.7	Comparison of the exact solution and PGD solution with 26 modes. . . .	49
3.8	End loaded beam. Left: geometry; Right: spatial meshes.	50
3.9	Comparison of the PGD modal amplitudes with two different spatial meshes.	51
3.10	Vertical displacement result comparison of the selected location. Top: from coarse spatial mesh; Bottom: from fine spatial mesh.	52
4.1	Shape of the constrained parametric domain.	56
4.2	Sketch of the 2D problem with material properties as extra coordinates. .	58
4.3	Normalised modal amplitude plots for both separated and unseparated approach.	59
4.4	Normalised spatial modes $\mathbf{U}_{\text{sep}}^i$ solved from $I_{E_1} \times I_{E_2}$. Bottom plane: spatial domain. Vertical axis: magnitude of nodal displacement.	60
4.5	Normalised spatial modes $\mathbf{U}_{\text{col}}^i$ solved from Ω_c . Bottom plane: spatial domain. Vertical axis: magnitude of nodal displacement.	60
4.6	Normalised parametric modes $\omega_1^j(E_1) \otimes \omega_2^j(E_2)$ solved from $I_{E_1} \times I_{E_2}$. Bottom plane: parametric domain. Vertical axis: parametric function value.	61
4.7	Normalised parametric modes $\omega^j(E_1, E_2)$ solved from unseparated Ω_c . Bottom plane: parametric domain. Vertical axis: parametric function value.	61
4.8	Relative error of both separated and unseparated PGD approach for the 2D model problem. For the separated approach, three different meshes for the parameter space are used. For the unseparated approach, the discretisation remains the same.	64
4.9	Illustration of the possible equivalent loading location on the femur head.	65
4.10	Extraction of the involved nodes (left) within Abaqus GUI and parametric mesh (right) reconstructed with in-house code.	65
4.11	Parametric mesh (right) constructed by direct extraction of involved tetrahedral elements (left) using Abaqus GUI.	65
5.1	Discretisation of the constrained parametric domain. Left: coarse mesh with 83 triangles, $(\alpha_{\text{max}}, \beta_{\text{max}}) = (10, 10)$; right: subdomain mesh with 456 triangles, $(\alpha_{\text{max}}, \beta_{\text{max}}) = (10, -8)$	69
5.2	Error norm $\ \boldsymbol{\varepsilon}_{\mathbb{P}}^* - \boldsymbol{\varepsilon}_{\mathbb{P}}(\alpha, \beta)\ $ distribution on the parametric space. Left: preliminary result from coarse mesh; right: secondary result from fine mesh on local subdomain.	70
5.3	Normalized modal amplitude plot for triangular and tetrahedral parametric meshes.	72
5.4	The z component of normalized spatial modes computed with triangular mesh (upper row) and tetrahedral mesh (lower row).	72

5.5	Parametric modes $\omega^j(\boldsymbol{\mu})$ computed with triangular mesh (upper row) and tetrahedral mesh (lower row).	73
6.1	Typical set of B-spline shape functions and the corresponding derivatives.	81
6.2	FE mesh of the mechanical model.	87
6.3	Load cases of the mechanical model.	87
6.4	Reference displacement field computed with standard FE method, amplified for visualisation. Left: LC A. Right: LC B.	88
6.5	Displacement fields computed with PGD with different discretisations of parameter spaces for LC A.	89
6.6	Displacement fields computed with PGD with different discretisations of parameter spaces for LC B.	90
6.7	Differences of displacement fields computed with PGD with different discretisations of parameter spaces for LC A.	91
6.8	Differences of displacement fields computed with PGD with different discretisations of parameter spaces for LC B.	91
6.9	Convergence of the displacement field. Left: LC A. Right: LC B.	92
6.10	Reference sensitivity fields computed with PGD, parametric space polynomial order $p = 6$, number of knot span (element) $n_{\text{span}} = 6$, LC A.	92
6.11	Reference sensitivity fields computed with PGD, parametric space polynomial order $p = 6$, number of knot span (element) $n_{\text{span}} = 6$, LC B.	92
6.12	Convergence of the sensitivities. LC A.	93
6.13	Convergence of the sensitivities. LC B.	93
7.1	Deformation of a continuum body.	97
7.2	Illustration of the model (1/8 model due to symmetry). Edge length both 50 mm, hole radius 10 mm, model thickness 2 mm.	113
7.3	Dependency of displacement norm $\ \mathbf{U}\ $ on parameter μ	113
7.4	Singular value spectrum for each parameter.	114
7.5	Amplitude of the mode $\ \boldsymbol{\varphi}_i\ \cdot \ \boldsymbol{\omega}_i\ $ decreases with modal index i , while computational cost increases when more modes are computed.	115
7.6	Compare between SVD singular values and PGD amplitudes. It shows the less optimality of PGD before compression.	116
7.7	First 10 PGD parametric modes.	116
7.8	First 10 PGD spatial modes.	117
C.1	Collected femur samples. Left: individual view. Right: Pile-up view.	153
C.2	A level set example.	153
C.3	2D embedding of Isomap result.	154
C.4	3D embedding of Isomap result.	154
C.5	Residual variance of Isomap result.	155

List of Tables

2.1	Calibration results for femurs	25
4.1	Randomly selected parameter sets	62
7.1	Acceleration of PGD in a typical online computation	114
7.2	Relative error for selected parameters	115

List of Algorithms

3.1 Typical PGD algorithm	35
4.1 Fully non-intrusive PGD algorithm	57

Chapter 1

Introduction

Biomechanics is a subject dedicated to the study of structures and functions of mechanical aspects of biological systems, at levels from whole organisms to organs, cells and cell organelles, using the methods of classical mechanics. It is both an old and new discipline from the historic point of view. It is known that Aristotle wrote the first book on the motion of animals, *De Motu Animalium*, which is regarded as the first antique literature on biomechanics. Interestingly, one of the first works of biomechanics in 17th century, a page of which is shown in [Figure 1.1](#), written by a Renaissance Italian academic Giovanni Alfonso Borelli used the same title. The well-known Leonardo da Vinci made notable contribution to the development of biomechanics during the Renaissance period, thanks to his talents of drawing and interests in anatomy in the context of mechanics. Later scientists like Galileo Galilei paid special attention on biomechanics, for instance, Galilei suggested that bones are hollow because this affords maximum strength with minimum weight. The nomenclature of biomechanics did not emerge in the long history of the establishment of classical mechanics, which based on the milestone works by Issac Newton, Joseph-Louis Lagrange and William Rowan Hamilton, until the late 20th century when [Hatze \[1974\]](#) firstly discussed “the meaning of the term ‘biomechanics’”. Having stated “biomechanics is mechanics applied to biology” in the monograph [Fung \[1993\]](#), the Chinese-American professor Yuan-Cheng Fung devoted most of his life on this “new” branch and is regarded as the father of modern biomechanics.

Without exception from problems in classical mechanics, biomechanics problems deal with mechanical behaviours of bio-solids and/or bio-fluids. The prefix “bio-”

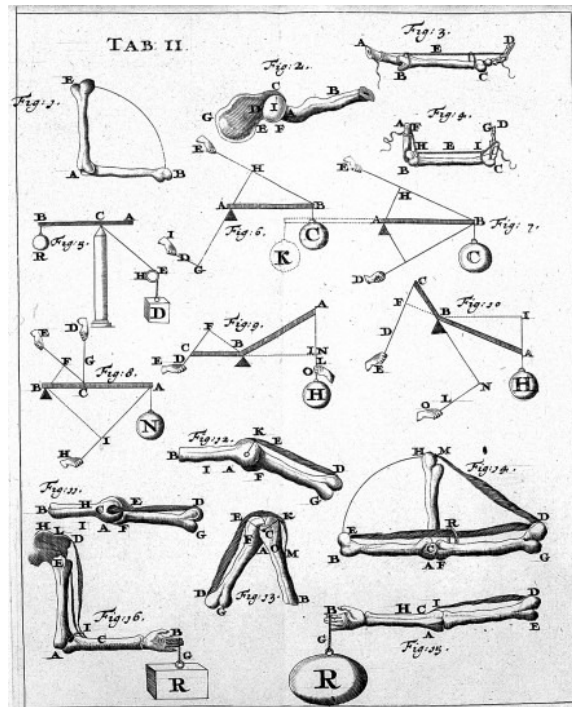


Figure 1.1: Page of *De Motu Animalium* by Giovanni Alfonso Borelli in the 17th century.

demonstrates the special biological mechanisms which are commonly represented by specific constitutive laws, but for brevity purposes, we will remove this prefix later on. This thesis will discuss mechanical problems focusing on solids. The solid constitutive models widely used on biomechanics are roughly divided into two types corresponding to the biological tissues. For “hard” tissues such as bones, teeth and so on, linear elastic models are generally used; for “soft” tissues such as hair, muscles, blood vessels and so forth, hyperelastic models are mainly adopted.

Simulation-Based Engineering Sciences (SBES) today require conceptual and numerical models to solve problems involving multi-physics and multi-scales. Most numerical models are mathematically based on partial differential equations (PDEs). Numerical methods for solving PDEs, such as finite difference method (FDM) [LeVeque \[2007\]](#), finite element method (FEM) [Hughes \[1989\]](#), [Zienkiewicz et al. \[2013\]](#) and finite volume method (FVM) [Versteeg and Malalasekera \[2007\]](#), are intensely implemented in engineering research and industry, covering a wide range of fields from aerospace to biomedical engineering. However, the intrinsic complexity of numerical methods for solving PDEs prevents finding solutions with high fidelity and high efficiency simultaneously. Therefore, further development of novel techniques is

necessary.

In this work, FEM will be used as the basic method. As a well established numerical method, FEM has dominated the branch of structural mechanics since the 1960s [Stein \[2014\]](#). Afterwards, it has been intensively studied mathematically, and its effectiveness for elliptic boundary value problems (BVPs) was later investigated, together with the *a priori* and *a posteriori* error analysis [Babuška and Rheinboldt \[1978\]](#), [Ladevèze and Leguillon \[1983\]](#), [Larson \[2000\]](#), [Ainsworth and Oden \[2000\]](#), [Oden and Prudhomme \[2001\]](#), [Chamoin and Díez \[2016\]](#). Parallel to this, engineering developments have been made for linear and nonlinear problems of the classical PDEs in mathematical physics to solve complicated industrial problems. Inspired and driven by the fast growing computer science and technology, the dimension of solvable algebraic equations is becoming larger and larger. Nowadays, a large number of finite element (FE) software are available, both commercial and open-sourced. Among all the popular commercial FE solvers, Abaqus (Dassault Systèmes, France) is chosen for the thesis work.

It is notable that the numerical methods, including but not limited to FDM, FEM and FVM, are able to provide a virtual or *in silico* platform for material/mechanics testing or *in vitro* experiments, and to produce full-order (or the so-called *high-fidelity*) approximations of the physical reality. However, for the aforementioned problems involving multi-physics and multi-scales, the computational cost of high-fidelity solutions could be as expensive as many hours, or even several days, of CPU time, due to the large amount of degree of freedom (DOF) which results in calculations of very large matrices. For applications that require fast response and multi-queries, it is not efficient enough in practice. In fact, the large number of DOF is usually derived from discretised PDEs involving multiple parameters, which will naturally result in the so-called *parametrised PDEs*. An idea of solving the parametrised PDEs is to assume the parameters as extra coordinates, and solve a generalised PDE. This would cause the dimensionality of the problem increase, leading to an exponential growth of the DOF as well as computational costs, i.e. the so-called *curse of dimensionality*. Therefore, proper reduction of the dimensionality, which induces losing as little accuracy as possible, of the origin problem has become necessary. It should be noted that in this case, the dimensionality increase arises not only from the discretisation of physical space, but also from the parameter space.

There are at least two categories of techniques for the reduction of the complexity

of a physical model. The first category reduces the model conceptually in a physical or geometric aspect. For instance, in structural mechanics, it is common to use beam element and shell element to reduce the physical model from 3D spaces to 1D/2D spaces. When the geometric feature of the problem satisfies certain criteria, planar symmetric, axisymmetric assumptions are widely used to reduce the model dimension. For blood flow analysis in the circulatory network, the geometrical multi-scale modelling strategy has been developed [Quarteroni et al. \[2000\]](#), [Quarteroni and Veneziani \[2003\]](#), with the idea of reducing the global geometric model of vascular network from 3D space to 1D/0D using analogy of the electric circuit network, while keeping local Navier-Stokes model in full 3D space. Recently the strategy of hierarchical model reduction has evolved for the vessel flow model [Ern et al. \[2008\]](#), [Perotto et al. \[2010\]](#), [Perotto and Zilio \[2013\]](#). The idea is to assume a fibre bundle structure on the geometric domain, and use different approximate function spaces to characterise the dependence of full solution on the dominant flow direction and on the transverse directions. In biomechanics, geometric shape of the tissue model is usually very complicated, and thus its description relies on high dimensional spaces. It is always beneficial to reduce the dimensionality of the geometric shape space to a reasonable low order.

The second category, which will be focused on throughout this thesis, reduces the mathematical complexity of the model with devised algorithms. It is well-known that models containing multiple parameters generally contain redundant or less relevant information. The idea of *reduced order modelling*, or *model order reduction*, is to use mathematical methods to remove the redundancy or minor relevancy, keeping only the essential information in a so-called *reduced order model*. Typically, the mathematical description of the model is the aforementioned parametrised PDEs, therefore, the practical content of model order reduction is to find the solution of the parametrised PDEs.

As it is well-known, the reduction of dimensionality of the physical space would decrease the computational cost of each simulation, while that of the parameter space would lower the size of the space to be explored and therefore the number of queries. It is natural to combine the two categories of model reduction techniques globally or locally in specific problems.

In order to explain the preliminary idea of reduced order modelling, now we introduce a classical example: the compression of a digital image using the singular value

decomposition (SVD) method. Taking grey scale images for example, the mathematical model of a digital image with resolution $m \times n$ is a 2D matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, with each element containing a grayscale value $M_{ij} \in [0, 255]$. This matrix contains all information stored in the image. As mentioned before, there exists less relevant information in this matrix, and thus we can use SVD to analyse the data, and obtain a reduced or compressed image.

In mathematics, the SVD of an $m \times n$ matrix \mathbf{M} reads

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{\min(n,m)} \sigma_i \mathbf{U}_i \otimes \mathbf{V}_i, \quad (1.1)$$

where $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_m]$ is an $m \times m$ orthogonal matrix with column vectors \mathbf{U}_i , $\mathbf{\Sigma}$ is a diagonal $m \times n$ matrix with non-negative real numbers σ_i on the diagonal, and $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_n]$ is an $n \times n$ orthogonal matrix with column vectors \mathbf{V}_i . The diagonal elements σ_i of $\mathbf{\Sigma}$ are known as the singular values of \mathbf{M} . It is common to list the singular values in a descending order, i.e., $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)}$. The mathematical interpretation of reducing the image is to make a truncation on the number of summed terms for \mathbf{M} . That is, using a small number $k \ll \min(m, n)$ to substitute $\min(m, n)$, and obtaining \mathbf{M}_k as an approximation of \mathbf{M} :

$$\mathbf{M} \approx \mathbf{M}_k := \sum_{i=1}^k \sigma_i \mathbf{U}_i \otimes \mathbf{V}_i. \quad (1.2)$$

In this sense, we can interpret that $\sigma_i \mathbf{U}_i \otimes \mathbf{V}_i$, ($i = 1, 2, \dots, k$) consist a reduced basis of \mathbf{M}_k .

Now we analyse an image with a resolution 667×500 , thus the dimension is $333\,500$. Its normalised singular value spectrum is plotted in [Figure 1.2](#), from which we can observe a drastic drop of the singular value around the first 50 entries. The first 10 basis images are shown in [Figure 1.3](#). The original image, together with three reduced images reconstructed with different number of the basis images, is illustrated in [Figure 1.4](#). It can be seen that although one could not perceive any meaningful information from single basis image, with the sum of the first 10 basis images, a sketch of the original image is already presented, and more detailed information emerges when the number of summed basis increases.

The idea presented in the image compression example is generalised to different types of problems, most of which involves parametrised PDEs, and this generalisation eventually establishes the methodology of reduced order modelling.

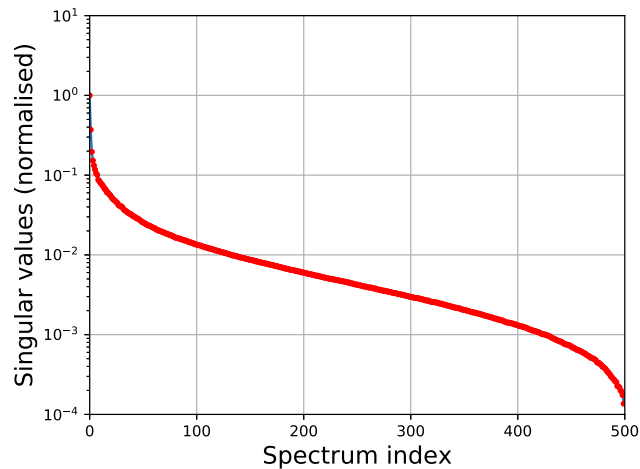


Figure 1.2: SVD spectrum of an 667×500 image.

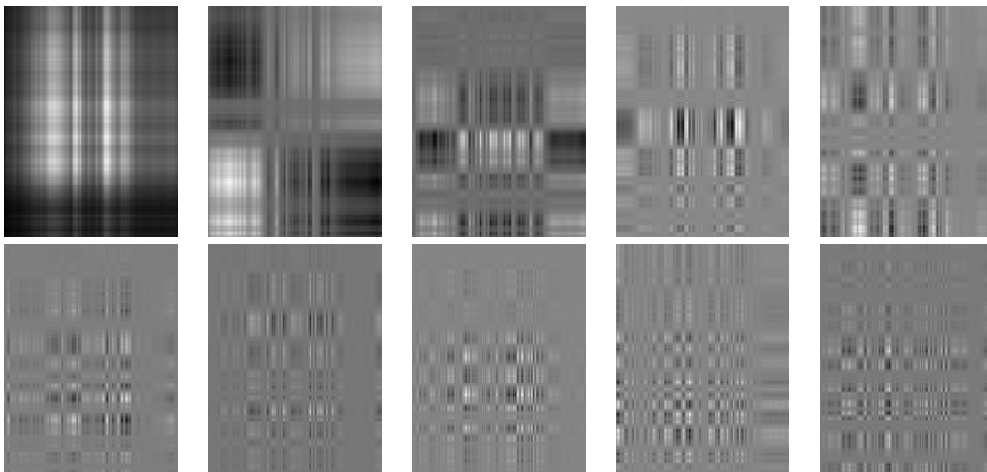


Figure 1.3: First 10 basis images reconstructed with $\sigma_i \mathbf{U}_i \otimes \mathbf{V}_i$, ($i = 1, 2, \dots, 10$).

In general, the computational process of reduced order modelling consists in two phases: an *offline* phase which constructs the reduced basis, it is usually considerably expensive, such as the SVD in previous example; and an *online* phase which reconstructs the approximation of the model using the reduced basis generated in the offline phase. The cost of the online computations is much cheaper than that at the offline phase, and thus its extremely fast speed is ideal for real-time simulations.

As mathematical techniques to build the reduced order models, in this chapter we will introduce briefly state of the art methods, such as Reduced Basis (RB) [Quarteroni et al. \[2016\]](#), proper orthogonal decomposition (POD) [Liang et al. \[2002a,b\]](#) and proper generalised decomposition (PGD) [Chinesta and Ladevèze \[2014\]](#). As the main

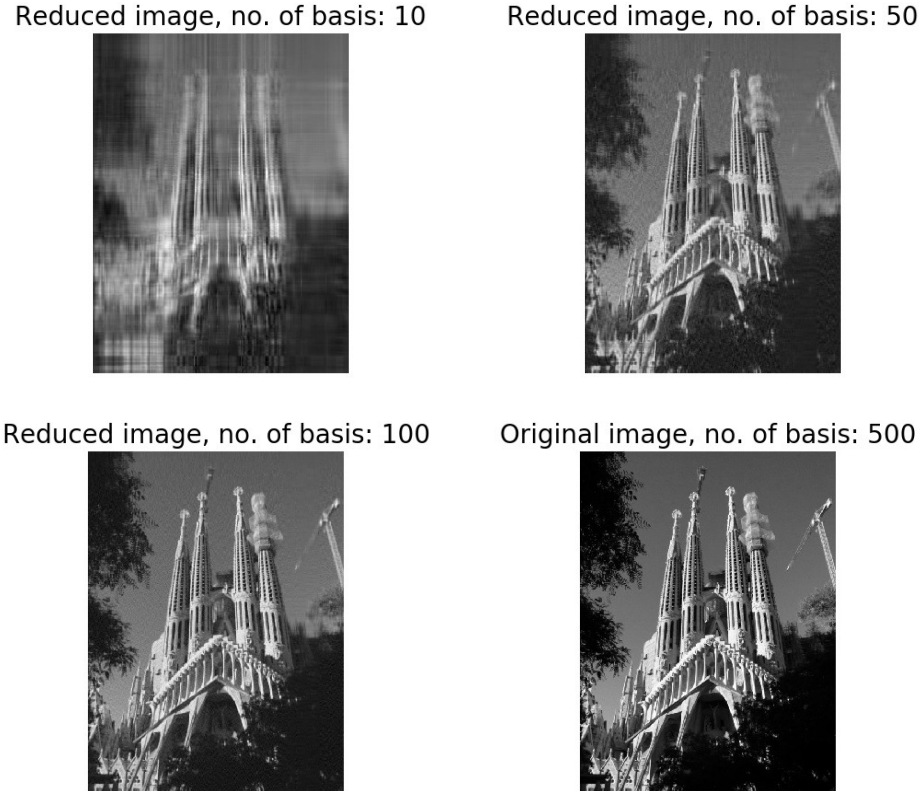


Figure 1.4: Reduction of an 667×500 image, reconstructed with $\sum_{i=1}^k \sigma_i \mathbf{U}_i \otimes \mathbf{V}_i$, ($k = 10, 50, 100, 500$).

topic in this work, PGD will be focused on in later chapters.

1.1 Background

Along with the rapid growth of computational techniques, simulation of mechanical response of biological tissues is generally adopted in the research of biomechanics. From musculoskeletal or cardiovascular systems at macro scale to cell or tissue structures at micro scale, numerical methods are commonly used, especially FE methods.

For many clinical applications, fast-response and multi-query are frequently required for patient-specific simulations. Current techniques are able to generate high-fidelity solutions [Taddei et al. \[2006\]](#), [Trabelsi et al. \[2009, 2011\]](#), however, they are usually not provided in time, especially in case of emergency. Consequently, it is very interesting and necessary to explore the capability of reduced order modelling, in order to find an optimised balance between the accuracy and speed of simulations.

1. INTRODUCTION

As a practical application, we will use the technique of reduced order modelling for the structural analysis of a human proximal femur.

The femur, also known as the thigh bone, is the largest and strongest bone in the human musculoskeletal system. As the proximal part of the leg, it connects the hip and the knee, and is vital for the functions requiring leg movement such as walking, running and jumping. The anatomic terminologies of a right proximal femur seen from the back are illustrated in [Figure 1.5](#) with annotations.

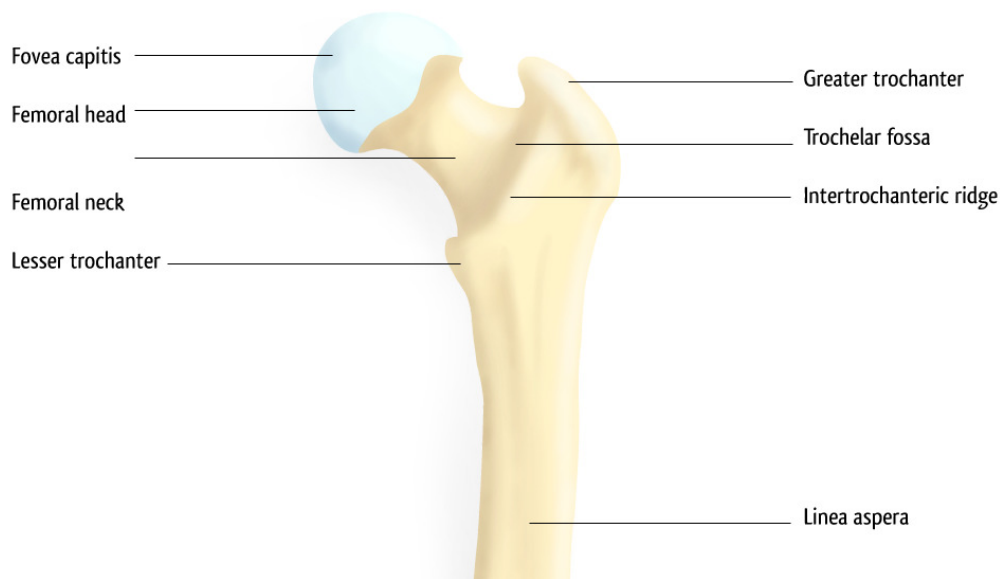


Figure 1.5: Anatomic terminologies of a right proximal femur, seen from the back. (Case courtesy of A.Prof Frank Gaillard, <https://radiopaedia.org>, rID: 7555)

Due to its important role, a femur carries heavy mechanical loads up to 30 times of the body weight, and thus it is also one of the most vulnerable bone in the musculoskeletal system. For the treatment of severe hip joint degenerative pathologies, total hip arthroplasty (THA) [Siopack and Jergesen \[1995\]](#) has become a common procedure worldwide. With the help of ROM, it is possible to construct a practical surgery simulator which is able to provide the expected fast-response and multi-query for real-time simulations.

1.2 State of the art

Models in engineering and industry are usually complex systems a priori described by a large number of variables and parameters. High-dimensional data are collected

from observations, measurements and calculations. However, it is often known that the number of dimensions in the *a priori* description typically exceeds the essential dimensionality of the system. Therefore, the objective of model order reduction is to use different techniques to reduce the number of variables to fit the intrinsic dimensionality of the system, thus to reduce the computational cost and to achieve fast-response and multi-query for real-time simulations.

To better explain the technical details, we now introduce the standard notations used for the high-fidelity problems in FEM and many other methods. Consider the abstract form of a linear problem, which reads: find $u \in U$, such that

$$a(u, v) = l(v), \quad \forall v \in V, \quad (1.3)$$

where u is called the trial or basis function while v is denoted as the test function, $a(u, v)$ is a bilinear form while $l(v)$ is a linear form, U and V are proper function spaces such as the Hilbert space.

To make the problem solvable on computers, algebraic formulation is generated by discretisations based on the abstract form [Equation 1.3](#). The corresponding algebraic equation reads

$$\mathbb{A}\mathbf{u} = \mathbf{f}, \quad (1.4)$$

where \mathbb{A} is the stiffness matrix, \mathbf{u} is the DOF vector to be solved, and \mathbf{f} is the load vector.

1.2.1 Reduced order modelling with RB

The reduction of the model is based on the dimension reduction of the discretised solution space of the governing PDEs and the parametric space to be explored. The low-order space is defined by a low-dimensional basis referred as the reduced basis. Although it is not the focus of this thesis to investigate the reduce basis methods, we find it necessary to introduce briefly the essence of RB methods. It will be highlighted in [Subsection 1.2.2](#) that the PGD formulation could be naturally established based on the clear mathematical definitions from RB methods.

We refer to [Rozza \[2005\]](#), [Rozza et al. \[2008\]](#), [Quarteroni et al. \[2011\]](#) for a general review of the reduced basis approximation and *a posteriori* error estimation methods for the rapid and reliable evaluation of engineering outputs associated with parametrised PDEs. The low-dimensional approximation space can be established by

the span of a set of characteristic solution results, i.e. the reduced basis, which also called *snapshots*.

The essential ingredients of the dimension reduction with RB include (primal-dual) Galerkin projection onto a low-dimensional space associated with a smooth “parametric manifold” Milani et al. [2008]; efficient and effective greedy sampling methods for identification of optimal and numerically stable approximations and a rapid convergence. On one hand, the incomplete RB would cause an inaccurate approximation, whilst the error can be estimated; on the other hand, consideration of too many snapshots may bring redundant information, resulting in ill-conditioned algebraic systems of equations.

Now we introduce briefly the details of the RB methods. Given a parameter space Ω_μ and denote the parameters as $\boldsymbol{\mu} \in \Omega_\mu$, we generalise the problem in Equation 1.4 to following parametrised formulation

$$\mathbb{A}(\boldsymbol{\mu})\mathbf{u}(\boldsymbol{\mu}) = \mathbf{f}(\boldsymbol{\mu}). \quad (1.5)$$

It is obvious that for any specified $\boldsymbol{\mu}$, a high-fidelity solution $\mathbf{u}(\boldsymbol{\mu})$ can be obtained by solving the linear system Equation 1.5. Let’s assume this linear system has a rank N , we have $\mathbb{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$, $\mathbf{u}(\boldsymbol{\mu}) \in \mathbb{R}^N$ and $\mathbf{f}(\boldsymbol{\mu}) \in \mathbb{R}^N$. The key idea of the RB method is to seek an approximation of \mathbf{u} in a subspace with a much lower dimension $n \ll N$. This is achieved by using a Galerkin projection of the original operator. The reduced order model represented by the corresponding linear system reads

$$\mathbb{A}_n(\boldsymbol{\mu})\mathbf{u}_n(\boldsymbol{\mu}) = \mathbf{f}_n(\boldsymbol{\mu}), \quad (1.6)$$

where $\mathbb{A}_n(\boldsymbol{\mu}) \in \mathbb{R}^{n \times n}$, $\mathbf{u}_n(\boldsymbol{\mu}) \in \mathbb{R}^n$ and $\mathbf{f}_n(\boldsymbol{\mu}) \in \mathbb{R}^n$. The new low-dimensional unknown \mathbf{u}_n is called *reduced basis solution*. Explicitly, the high-fidelity solution is approximated by

$$\mathbf{u}(\boldsymbol{\mu}) \approx \mathbb{V}\mathbf{u}_n(\boldsymbol{\mu}), \quad (1.7)$$

where $\mathbb{V} \in \mathbb{R}^{N \times n}$ is called the *transformation matrix* or *projection matrix* which depends on $\boldsymbol{\mu}$. With this projection, the approximated high-fidelity problem can be written as

$$\mathbb{V}^T[\mathbf{f}(\boldsymbol{\mu}) - \mathbb{A}(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(\boldsymbol{\mu})] = \mathbf{0}. \quad (1.8)$$

It can be seen that $\mathbb{A}_n(\boldsymbol{\mu}) = \mathbb{V}^T \mathbb{A}(\boldsymbol{\mu}) \mathbb{V}$ and $\mathbf{f}_n(\boldsymbol{\mu}) = \mathbb{V}^T \mathbf{f}(\boldsymbol{\mu})$.

In most cases, it is assumed that for the stiffness matrix $\mathbb{A}(\boldsymbol{\mu})$ and the load vector $\mathbf{f}(\boldsymbol{\mu})$, the dependency on parameters is *affine*, that is

$$\begin{aligned}\mathbb{A}(\boldsymbol{\mu}) &= \sum_{j=1}^p \varphi_j(\boldsymbol{\mu}) \mathbb{A}_j, \\ \mathbf{f}(\boldsymbol{\mu}) &= \sum_{k=1}^q \theta_k(\boldsymbol{\mu}) \mathbf{f}_k,\end{aligned}\tag{1.9}$$

where φ_j and θ_k denotes scalar functions of the parameters, while \mathbb{A}_j and \mathbf{f}_k are constant matrices or vectors independent of $\boldsymbol{\mu}$.

The most important ingredient of the RB method is the approach to generate the snapshots wisely. POD and greedy algorithm are the two mostly used methods. To obtain the RB properly during the offline phase, POD technique is commonly used [Ryckelynck et al. \[2006\]](#). Depending on the application field, POD is also known as principal component analysis (PCA) or discrete Karhunen-Loève transform (KLT) [Maccione \[2009\]](#). Mathematically, POD is highly related to SVD, which provides a convenient technique to extract the dominant elements from a highly redundant family. Recently, further reduction on the POD reduced model introduces the so-called hyper-reduction method [Ryckelynck \[2009\]](#), [Horák et al. \[2017\]](#), [Hernández et al. \[2017\]](#), considering governing equations only over a subdomain which is called reduced integration domain. In many cases, once a proper (preferably goal-oriented) *a posteriori* error estimator is available, the greedy algorithm can be used to search for the reduced basis more efficiently and thus accelerate the offline computation.

It is worth noting that POD or SVD provides the optimal reduced basis in simple cases, such as elliptic PDE with only one parameter, but this is not the case for PDE problems with more than one parameter, although the so-called high-order SVD (HOSVD) is available [Ammar et al. \[2014\]](#).

Having been developed during the past decade, several implementation of RB methods are already available, such as the `rbMIT` package (<http://augustine.mit.edu/>), the `RBmatlab` package (<http://www.ians.uni-stuttgart.de/MoRePaS/>), the `pyMOR` package (<http://pymor.org/>) [Milk et al. \[2016\]](#), the `redbKIT` package (<http://redbkit.github.io/redbKIT/>) [Quarteroni et al. \[2016\]](#), etc.

1.2.2 Reduced order modelling with PGD

In recent decade, an *a priori* model order reduction technique, PGD, has been developed [Ammar \[2010\]](#), [Chinesta et al. \[2010, 2011b\]](#), [Chinesta and Ladevèze \[2014\]](#),

Chinesta and Cueto [2014], Chinesta et al. [2014]. This technique features the separated representation of the solution, so that the relationship between the solution complexity scale and the dimension of solution space is reduced from exponentially to linearly, and also the greedy algorithm. The reduction made by PGD makes it possible to solve multidimensional models efficiently by means of treating parameters as extra coordinates and obtaining the reduced basis as a so-called *computational vademecum* Chinesta et al. [2013] during the offline phase. With the obtained vademecum, fast-response and multi-query can be achieved in real-time simulations which are denoted as online phase. The power of PGD is demonstrated in many different fields, such as structural analysis Vidal et al. [2012, 2014], structural optimisation Leygue and Verron [2010], Ammar et al. [2014], Courard et al. [2015], computational rheology Chinesta et al. [2011a], computational fluid dynamics González et al. [2013], Díez et al. [2017], heat transfer Berger et al. [2017], power supply system García-Blanco et al. [2017], parameter identification Nadal et al. [2015a] etc.

Now we introduce briefly the basics of PGD, using the previously established concepts from RB methods. In PGD, the parameters are considered as extra coordinates, and thus the solution space is generalised to a Cartesian product of physical space and the parameter space. To establish an abstract form of this generalised problem, one need not only integrate over the physical space, but also on the parameter space, obtaining the so-called PGD generalised weak form. To make the generalised problem able to be discretised and thus solvable, the separated formulation is introduced for the approximation of the unknown \mathbf{u} :

$$\mathbf{u}(\boldsymbol{\mu}) \approx \sum_{i=1}^n \omega_i(\boldsymbol{\mu}) \mathbf{u}_i, \quad (1.10)$$

where ω_i denote the functions depending on $\boldsymbol{\mu}$, and \mathbf{u}_i are the constant vectors independent of parameters. Inspired by the similarity of modal analysis in vibration mechanics, a component $\omega_i(\boldsymbol{\mu}) \mathbf{u}_i$ is often referred to the i th *mode*, and $\omega_i(\boldsymbol{\mu})$ is called the i th parametric mode while \mathbf{u}_i the i th spatial mode. In PGD, the unknown modes are solved sequentially with a initial guess.

Taking $\omega_n(\boldsymbol{\mu})$ as the test function and applying the PGD methodology, the linear system in RB formulation Equation 1.5 is now generalised to a weak form

$$\int_{\Omega_{\boldsymbol{\mu}}} \mathbb{A}(\boldsymbol{\mu}) \omega_n(\boldsymbol{\mu}) \sum_{i=1}^n \omega_i(\boldsymbol{\mu}) \mathbf{u}_i d\boldsymbol{\mu} = \int_{\Omega_{\boldsymbol{\mu}}} \omega_n(\boldsymbol{\mu}) \mathbf{f}(\boldsymbol{\mu}) d\boldsymbol{\mu}. \quad (1.11)$$

To enable the solvability of parametrised problem, the operator \mathbb{A} is also required to be separable. To achieve this, it is natural to assume the operator \mathbb{A} has a similar separated formulation as [Equation 1.10](#). Therefore, we now invoke the affine dependency assumption from RB methods [Equation 1.9](#) into [Equation 1.11](#), and the latter becomes

$$\begin{aligned} & \int_{\Omega_\mu} \sum_{j=1}^p \varphi_j(\boldsymbol{\mu}) \mathbb{A}_j \omega_n(\boldsymbol{\mu}) \sum_{i=1}^n \omega_i(\boldsymbol{\mu}) \mathbf{u}_i d\boldsymbol{\mu} = \int_{\Omega_\mu} \omega_n(\boldsymbol{\mu}) \sum_{k=1}^q \theta_k(\boldsymbol{\mu}) \mathbf{f}_k d\boldsymbol{\mu} \\ \implies & \sum_{i=1}^n \sum_{j=1}^p \left(\int_{\Omega_\mu} \varphi_j(\boldsymbol{\mu}) \omega_n(\boldsymbol{\mu}) \omega_i(\boldsymbol{\mu}) d\boldsymbol{\mu} \right) \mathbb{A}_j \mathbf{u}_i = \sum_{k=1}^q \left(\int_{\Omega_\mu} \omega_n(\boldsymbol{\mu}) \theta_k(\boldsymbol{\mu}) d\boldsymbol{\mu} \right) \mathbf{f}_k. \end{aligned} \quad (1.12)$$

It can be seen [Equation 1.12](#) is a nonlinear equation for $\omega_n(\boldsymbol{\mu})$, even if the original problem is linear. This is because a quadratic term ω_n^2 will emerge on the left-hand side when $i = n$.

Typically, a fixed-point scheme called *alternative direction iteration* is used to linearise the nonlinear equation and solve the modes in a sequence. This scheme, as will be explained in detail in the following chapters, is suitable for many different cases. However, the drawback is it is not able to be parallelised.

Unlike RB methods, we can see that the offline computation in PGD is not a collection of high-fidelity solutions with chosen parameters, but a generalised problem that lacks proper physical interpretation. Nevertheless, the online computation of PGD is conceptually faster since the modes are already computed and stored in the vademecum to be reconstructed with linear combinations. There is no need to solve any (reduced) linear system, which is necessary for RB methods. Currently, error estimation strategies of the PGD methods is under active research, several available procedures are already proposed in the literature, such as [Ammar et al. \[2010\]](#), [Alfaro et al. \[2015\]](#), [Nadal et al. \[2015b\]](#), [Zlotnik et al. \[2015b\]](#), [Allier et al. \[2015\]](#), [Chamoïn et al. \[2017\]](#).

Compared to the RB methods, there are less public codes available for PGD implementations. Most of the practical PGD frameworks currently available in the literature are based on intrusive implementations relying on academical FE source codes, commonly requiring cumbersome coding work. Typical PGD codes could be found in [Cueto et al. \[2016\]](#). It is worth noting that due to the characteristic of PGD framework, it is ideal for a non-intrusive implementation [Duval et al. \[2016\]](#). Preliminary application is performed in literature such as [Courard et al. \[2015\]](#).

1.2.3 Application of reduced order models in biomechanics

In the biomedical field, applications of model order reduction methodology is promising. For instance, reduced order model is even used to simulate the tumour growth [Colin et al. \[2012\]](#), [Quarteroni and Rozza \[2014\]](#).

Thanks to the increasing computational power as well as progress in imaging and geometry extraction/reconstruction techniques with more efficient algorithms, real-time numerical simulation of blood flow problems have become quite popular in the past two decades [Manzoni et al. \[2012a\]](#). The driving factor behind this development is the awareness that numerical models can provide real-time quantitative descriptions of blood flow behaviour in important vascular districts or in vessel networks, and to explain and assess the relationships among vessels shape, haemodynamics and a family of clinical indicators. With the techniques combining geometric parametrisation and reduced basis method, it is possible to account for patient-specific vessel configurations. After the reconstruction by solving a suitable parameter identification problem, real-time simulation of blood flows are able to be performed on each reconstructed parametrised geometry. The approach advocated in [Manzoni et al. \[2012a\]](#) can be applied to a broad variety of (different) flow problems related with geometry/shape variation, for instance the cardiovascular vessels, and can be extended to related topics such as parametric geometry shape optimisation [Manzoni et al. \[2012b\]](#), [Lassila et al. \[2013a\]](#) and inverse problems [Lassila et al. \[2013b\]](#).

Real-time simulation of surgery [Cotin et al. \[1999\]](#) has attracted the attention of a wide community of researchers, from computer scientists to mechanical engineers, together with computational geometers, surgeons, etc. The utility of such techniques are obvious, and they include, for instance, surgery planning, training of surgeons in image-guided surgery or minimally invasive surgery, etc. The real-times simulations are supposed to provide a physically accurate response so that a realistic feedback is transmitted to the surgeon in terms of both visual feedback and force feedback. For that to be possible, it is commonly recognised that a minimum bandwidth of 20–60 Hz for visual feedback and 300–1000 Hz for haptic display is necessary. Such a frequency requires very short CPU time for the simulation. For solid objects, a general survey on the real-time deformable models is found in [Meier et al. \[2005\]](#). In general, two typical approaches are used in real-time simulations: heuristic models based on simplified physical laws [Agus et al. \[2003\]](#) and continuum mechanical models that follows mechanical equilibrium equations. For the continuum mechanical approach,

traditional FE models are less widely used due to the large computational costs, and condensation techniques are usually applied to reduce the cost [Bro-Nielsen and Cotin \[1996\]](#), [Cotin et al. \[1999\]](#).

To accelerate the deformable models, reduced order models have been developed to simulate biomedical tissues, which can even be hyperelastic materials, with approaches based on RB or PGD [Niroomandi et al. \[2008, 2013a\]](#), [Chinesta et al. \[2013\]](#). Paying the price of a considerable amount of offline computations, the online simulation can be performed on portable devices such as smartphones and tablet computers on which the vademecum is stored. Such feature enables a comfortable accessibility and convenient application in a practical surgery [González et al. \[2015\]](#), [Quesada et al. \[2016a\]](#). Reduced order models are considered to be the very technique able to simulate at real-time feedback rates, highly complex constitutive models for living tissues [Niroomandi et al. \[2012a,b\]](#) (fibre-reinforced hyperelastic models, for instance). It is also feasible to extend model order reduction methodologies to the structural analysis of musculoskeletal systems.

To create a valid numerical model for bones, it is important to specify an accurate density-modulus relationship. Numerous identification studies have been performed for this purpose, most of which are based on statistics of various experimental data and a large number of corresponding FE simulations [Cong et al. \[2011\]](#). However, with the reduced order model, it is possible to perform *in silico* identification of the material properties with a few experimental data, thanks to the computational efficient online phase. For example, works on the identification of bone modulus parameters have been carried with sophisticated framework combining RB method and neural network [Zaw et al. \[2009\]](#).

1.3 Thesis objectives and chapter organisation

As a basis of this work, the medical image processing has to be performed, in order to generate an available FE model. This model should be validated according to data collected from a previously performed *in vitro* experimental test. The full procedure of image-based model generation and the validation of generated model is described in [Chapter 2](#).

As a major objective of this thesis, a non-intrusive scheme for the PGD framework is developed in [Chapter 3](#). It is implemented using in-house developed Matlab

(Mathworks, USA) code to conduct the PGD work flow, and calling Abaqus as an external solver for devised fictitious mechanical problems.

Since the FE model is image-based, the transformation of data from computed tomography (CT) image set to FE model including inhomogeneous material properties is subjected to some physical constraints, and when applying the load, there are also geometric constraints limiting the locations where load could be applied. These constraints will lead to a constrained parameter space, which possibly has difficulty to be separated in a Cartesian fashion. Therefore, a novel strategy to separate the parameters in a collective manner is proposed in [Chapter 4](#).

[Chapter 5](#) details a comprehensive application in biomechanics, the methodologies proposed in [Chapter 3](#) and [4](#) are applied on the practical model generated in [Chapter 2](#). As a typical application of the PGD vademecum, a material property identification problem is discussed. Further PGD vademecum is generated using the identified material properties with variable loading locations, and with this vademecum, real-time mechanical response of the femur is available.

In addition, for the purpose of extending the methodologies to orthotropic materials, which is commonly used in biomechanics, in [Chapter 6](#) another linear elastic model is investigated with the non-intrusive PGD scheme. Nowadays, isogeometric analysis (IGA) [Hughes et al. \[2005\]](#) is a very popular tool in computational mechanics. It is appealing to take advantage of non-uniform rational B-splines (NURBS) [Piegl and Tiller \[1997\]](#) to discretise the model. For PGD, using B-splines for the discretisation of the parameter space could improve the quality of vademecum, especially for problems involving sensitivities with respect to the parameters during the online computations.

It is important and necessary to extend the PGD framework to nonlinear solid mechanics, because most biological soft tissues have been observed nonlinear mechanical behaviours. Consequently, in [Chapter 7](#) we have developed a PGD framework for the St.Venant-Kirchhoff constitutive model using the Picard linearisation which is consistent with the fixed-point iteration algorithm commonly used in PGD.

In [Chapter 8](#), conclusive remarks are addressed as well as forecasts of possible future works.

Moreover, as complementary explanations of several technical details, [Appendix A](#) introduces PGD formulation involving the standard static condensation in FEM and [B](#) details the case of taking Young modulus and Poisson's ratio as the extra coordi-

nates.

It is well-known that a satisfactory geometric parametrisation for biological tissues is still an open question. In [Appendix C](#), attempts have been made using popular machine learning techniques to analyse the geometric shape of proximal femurs. Interesting indications have been obtained, which are assumed applicable for future developments.

Chapter 2

Image-based FE model generation and validation

In this chapter we introduce the basics of the image-based FE model generation and validation. From previously performed *in vitro* experimental tests, as shown in [Figure 2.1](#), the sample is vertically aligned with lower end fixed and upper end applied vertical loadings. Strains in longitudinal and circumferential directions are acquired at locations where strain gauges are attached. Tomographic images of the sample are later obtained through a CT scan, which is performed using a Siemens SOMATOM Emotion 6 CT scanner (Siemens Healthineers GmbH, Germany). The CT image sets are generated with Siemens Syngo CT 2006A and stored in standard DICOM (Digital Imaging and COmmunications in Medicine) format. As the essential input for numerical modelling, the CT images contain not only the geometric but also material constitutive information of the sample to be modelled. Therefore, a proper procedure of generating the FE model is important for an accurate simulation of mechanical behaviour of the bone.

2.1 From CT data to FE model

The CT images represent a 3D space of a box shape, i.e., $\Omega_{CT} = [0, a] \otimes [0, b] \otimes [0, c]$, where a, b, c denote the length, width and height of the box, respectively. Each CT image is a slice with $n_a \times n_b$ pixels, as shown in [Figure 2.2](#), the resolution r is

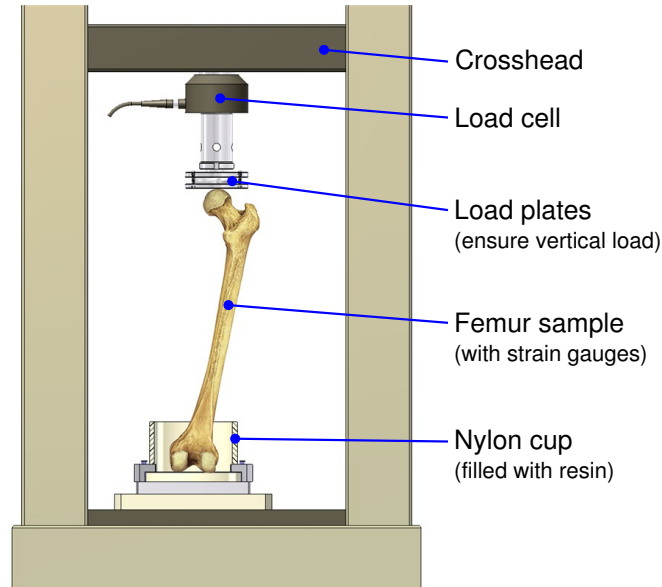


Figure 2.1: *In vitro* experiment settings.

determined by the CT machine, while $n_a = a/r$ and $n_b = b/r$. The distance between slices $\Delta_c = c/n_c$ can be controlled for different type of scans, resulting n_c slices. In total, a CT image set contains $n_a \times n_b \times n_c$ voxels and their corresponding grayscale value. Each voxel grayscale value can be mapped to the apparent density ρ .

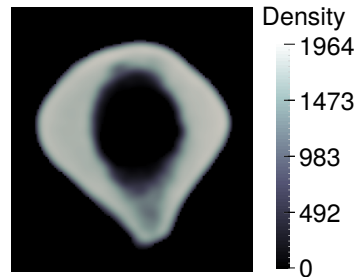


Figure 2.2: A CT slice example.

To perform image-based FE analysis, there are two typical approaches to generate practical FE models. The first approach is voxel-based, which directly transforms the CT voxel structure into hexahedral elements. This approach is convenient for direct FE modelling, and is frequently used in early FE analysis on bone biomechanics [Keyak and Skinner \[1992\]](#), [Keyak et al. \[1998\]](#). Along with the development of CT technology, the quantitative computed tomography (QCT)-based homogenised voxel FE modelling have also adopted this approach [Dall'Ara et al. \[2013\]](#). Lim-

ited by the image resolution and the corner singularity due to hexahedral shape, voxel-based models have difficulty in providing accurate stress analysis results. The second approach adds an additional procedure between the voxel CT data and the final FE model, which is a combination of segmentation and remeshing. Based on the fast development of computer image processing techniques, modern segmentation algorithm is able to extract specific objects from noisy backgrounds, and eventually obtain a smooth surface of the object in interest. Therefore, the outcome FE model is based on remeshing of a smooth geometry, and various types of element, not limited to hexahedrons, can be used. Latest FE models based on high-resolution peripheral QCT (HRpQCT) with segmentation algorithm showed significant improvement [Luisier et al. \[2014\]](#) from voxel-based FE models. Following this trend, we take advantage of the second approach in the creation of FE model of the femur.

The first step is to extract the interested spatial domain Ω from Ω_{CT} , this segmentation procedure, which will be detailed in [Section 2.2](#), filters the voxels that are void in the box, leaving only the part with $\rho > 0$. In particular, through the procedure we have obtained $\rho \in [1, 3071]$, with ρ being integer. In this work, the segmentation is performed with ITK-SNAP [Yushkevich et al. \[2006\]](#) as shown in [Figure 2.3](#). ITK-SNAP provides an active contour evolution methodology for edge detection, which is called “snake”. The contour evolves according to the following equation:

$$\frac{\partial}{\partial t} \mathbf{C}(t, u, v) = F \hat{\mathbf{N}},$$

where $\hat{\mathbf{N}}$ is the unit normal to the contour $\mathbf{C}(t, u, v)$ parametrised by variables u, v and time variable t , and F represents the sum of the internal and external forces that act on the contour in the normal direction. The internal forces are derived from the contour’s geometry and are used to impose regularity constraints on the shape of the contour, while the external forces incorporate information from the image being segmented. Active contour methods typically solve the contour evolution equation using the level set method [Osher and Sethian \[1988\]](#).

The segmented surface is triangulated, and can be exported into an STL mesh file for further editing and remeshing. Those meshes are not directly applicable for FE analysis, because

1. the mesh size is based on pixel size which is usually too fine;
2. distorted triangles prevent the successful conversion to a tetrahedral mesh.



Figure 2.3: Segmentation with ITK-SNAP.

Therefore, further smoothing and remeshing on the STL triangulation has to be performed. We choose to remesh it with VMTK [Antiga et al. \[2008\]](#). The target element edge length is set to 3 mm after a mesh convergence study as shown in [Figure 2.4](#).

Following most biomechanical FE analysis for solid bodies, the tetrahedral mesh is adopted in most image-based studies because of its relative higher efficiency/cost rate. It is also reported in [Ramos and Simões \[2006\]](#) that proximal femur experimental strains were well correlated with numerical ones using second order tetrahedral finite elements.

The spatial mesh for the finite element model of proximal femur is shown in [Figure 2.5](#). All the elements are tetrahedral element provided by Abaqus (C3D4 or C3D10). To comply with the *in vitro* experiment settings, the FE model is fixed at the distal end, and loaded on the femur head in vertical direction. To simulate the strain acquisition of the experiments, the numerical strains are extracted from the approximate positions located on two cross-sections of the shaft.

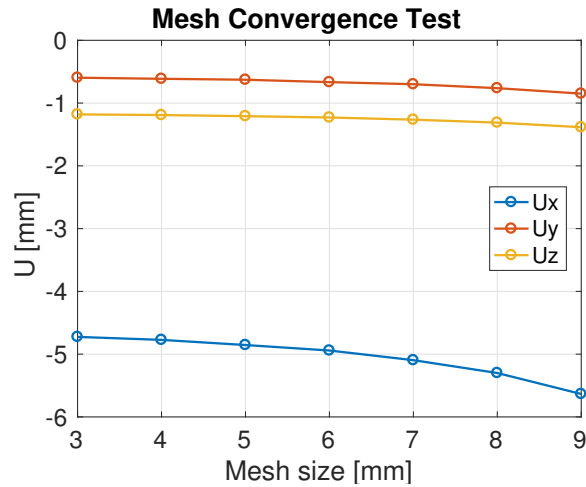


Figure 2.4: A mesh convergence study for the femur FE model. Mesh size is characterised by edge length of tetrahedrons, U denotes displacement of the loading point.

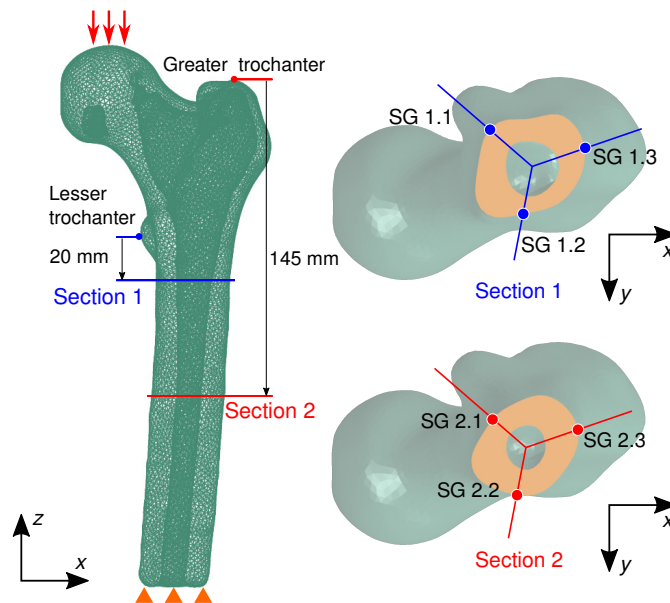


Figure 2.5: Spatial mesh of the human proximal femur model, illustrating locations with strain gauges attached for strain acquisition. Each location is attached two strain gauges in longitudinal and circumferential direction, respectively.

2.2 From grey scale data to material properties

It is observed that the influence of Poisson's Ratio ν is negligible, and $\nu = 0.3$ is usually adopted [Yosibash et al. \[2007\]](#). The isotropic assumption is generally employed as a certified simplification for the mechanical behaviour of the femur [Ramos and Simões \[2006\]](#), [Papini et al. \[2007\]](#).

The acquired CT images are stored in DICOM format, which digitally represents the Hounsfield Units (HU):

$$\text{HU}(\mu) = 1000 \cdot \frac{\mu - \mu_{\text{water}}}{\mu_{\text{water}} - \mu_{\text{air}}}, \quad (2.1)$$

where μ is the linear attenuation coefficient.

From definition of HU it is straightforward to identify that for water $\text{HU} = 0$ while for air $\text{HU} = -1000$. During the segmentation, we have already filtered HU with taking only the non-negative values.

The filtered HU is then converted to apparent density ρ using linear relations according to metadata of the DICOM files,

$$\rho \propto \text{HU}. \quad (2.2)$$

To map the density to mechanical properties, Young modulus, exponential law and linear law are both widely used in the literature. According to works by Keller [1994], Taddei et al. [2004] that focuses on the simulation of bone mechanics with structural elements, we adopt the assumption that the density of CT images is linearly mapped to Young modulus of the bone:

$$E(\mathbf{x}) = \alpha \rho(\mathbf{x}) + \beta, \quad (2.3)$$

where α and β are parameters that can be determined by experiments. With this mapping, we ignore the difference in biological tissues between the trabecular bone and cortical bone, but use variable moduli to represent its inhomogeneity in mechanics. Note that the CT image is, in fact, already discretised. For simplicity, in the finite element model we assign each element the density from the voxel which is closest to the centroid of the element, denoted as ρ^e .

2.3 Validation of the FE model

To validate the FE model, we perform some preliminary simulations with certain simplifications. In this validation, the bone material is firstly assumed to be homogeneous, regardless of the difference between trabecular bone and cortical bone.

The exact Young modulus of the real cortical bone is unknown. However, due to the linearity of the FE model, a preliminary analysis is performed with a presumed

Young modulus $E_0 = 20\,000$ MPa. Then, it is calibrated by matching the preliminary strain results with the experimental strain results.

To get optimised stress/strain results on the surface, Abaqus element type C3D10I is chosen. The C3D10I element is a 10-node quadratic tetrahedra element with improved stress visualization which is obtained through an 11-point Gauss-Lobatto quadrature rule Peano [1982], consisting of 10 integration points at the element nodes and one integration point at the centroid Simulia Corp. [2013].

Since the exact loading location is also unknown, an MPC element is used to distribute load to a set of possible nodes on the femur head from the assumed loading point. The amplitude of load is 2800 N, as it was set in previous experiments.

To calibrate Young modulus E , a parameter k is introduced. Due to the linearity we have $\varepsilon/\varepsilon_0 = E_0/E = k$. The least squared method is used to determine ε with results from simulation and experiment. The identification problem reads, find k to minimize the error norm

$$\|\boldsymbol{\epsilon} - k\boldsymbol{\varepsilon}_0\|^2 = \sum_i (\epsilon_i - k\varepsilon_{0i})^2 \quad (2.4)$$

where ϵ_i denotes experimental strains and ε_{0i} denotes strains from the preliminary simulations. Let $f(k) = \sum_i (\epsilon_i - k\varepsilon_{0i})^2$, we can find that $f(k)$ has a minimum value. Letting

$$f'(k) = 2k \sum_i \varepsilon_{0i}^2 - 2 \sum_i \epsilon_i \varepsilon_{0i} = 0, \quad (2.5)$$

we have

$$k = \frac{\sum_i \epsilon_i \varepsilon_{0i}}{\sum_i \varepsilon_{0i}^2}. \quad (2.6)$$

The calibration result for the femur FE model is listed in [Table 2.1](#).

Table 2.1: Calibration results for femurs		
E_0 (MPa)	k	$E = E_0/k$ (MPa)
20 000	1.436 274	13 924.92

Longitudinal and circumferential strains are obtained both from the simulation and *in vitro* tests, see the comparison between experimental results and calibrated simulation results in [Figure 2.6](#).

For a further comparison, the FE model with inhomogeneous material properties is also generated. By extracting the coordinates of centroids for each tetrahedron

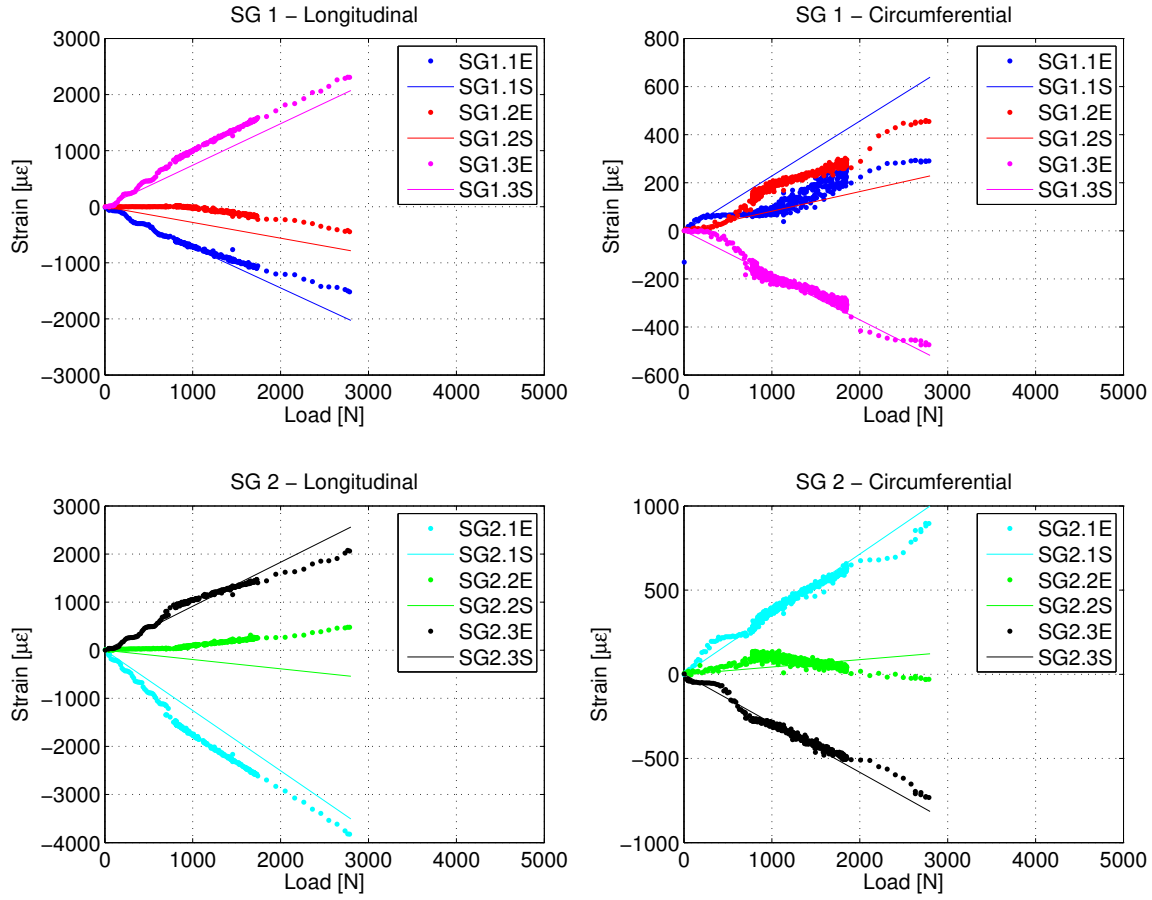


Figure 2.6: Comparison of Experimental (E) and simulated (S) strain results.

element, the density is mapped from HU value of the nearest voxel in the CT image. The mapping from HU to density ρ is taken from literature [Keyak et al. \[1994\]](#), which reads

$$E(\rho) = 14\,261\rho - 13\,430, \quad (2.7)$$

where the unit for E is MPa and that for ρ is g/ml. A typical slice of CT image and its corresponding slice in the FE model are shown in [Figure 2.7](#) with mapped Young modulus contour.

The simulation result of inhomogeneous FE model is shown in [Figure 2.8](#) in terms of strains at Section SG1. It demonstrates very small difference between homogeneous and inhomogeneous FE model. Calibration of Young modulus is still necessary for the inhomogeneous FE model.

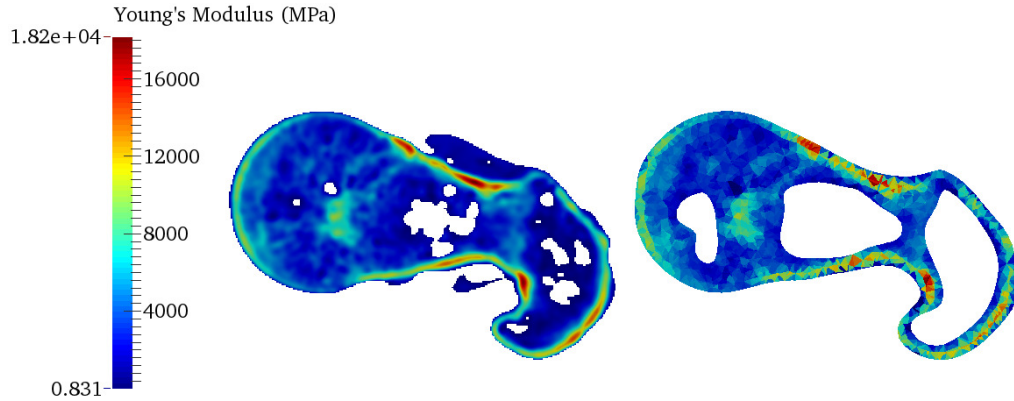


Figure 2.7: Young modulus contour. Left: CT image; Right: FE model.

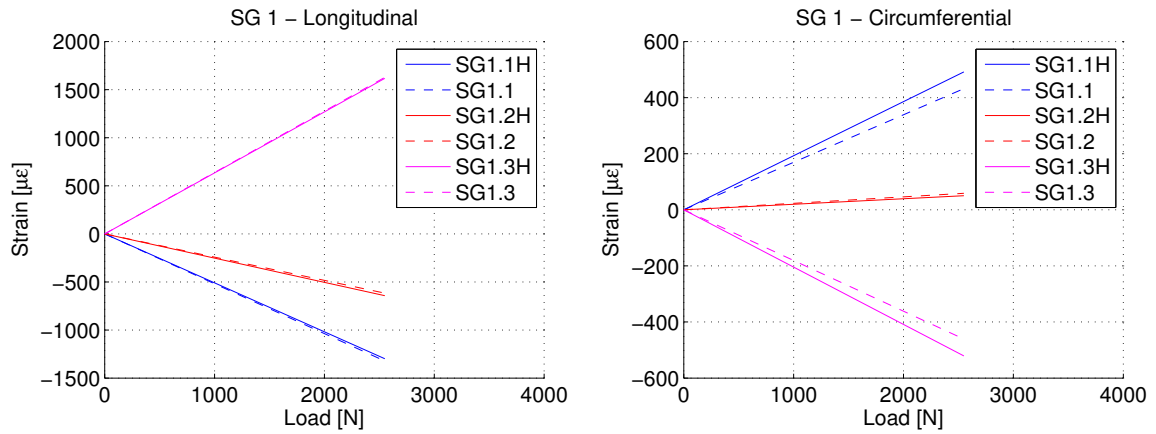


Figure 2.8: Comparison of strains on Section SG1. Left: longitudinal strains; Right: circumferential strains. Solid line: homogeneous model; Dashed line: inhomogeneous model.

2.4 Summary

This chapter briefly introduced the FE modelling procedure, especially the image-based modelling techniques that is employed. The generated FE model, which simulates the previously performed *in vitro* experimental test, is validated with both homogeneous and inhomogeneous material property distributions under isotropic assumption and a simple calibration. This model will be used for further simulations in next chapters.

Chapter 3

Non-intrusive PGD scheme

In this chapter, we introduce the general formulation of PGD framework and the details of its non-intrusive scheme, with practical implementation using Matlab and Abaqus.

For solid mechanics, linear elasticity is the simplest model for most materials under small deformation assumption. Here we review briefly the governing equations of linear elasticity for its later extensions in PGD formulation.

Consider an elastic body, which consists the computational domain, denoted as $\Omega \in \mathbb{R}^d$, ($d = 1, 2, 3$), usually the quantity of interest is the displacement field $\mathbf{u} = \mathbf{u}(\mathbf{x})$, $\mathbf{x} \in \Omega$. The boundary of Ω is often denoted as $\partial\Omega$. Under the infinitesimal deformation assumptions, the relationship between strain tensor $\boldsymbol{\varepsilon}$ and displacement $\mathbf{u}(\mathbf{x})$ is

$$\boldsymbol{\varepsilon} = \nabla^s \mathbf{u} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T], \quad (3.1)$$

where $\nabla = \partial/\partial \mathbf{x}$ is the gradient operator, and $\nabla^s = (\nabla + \nabla^T)/2$ is the symmetric gradient operator.

The stress tensor $\boldsymbol{\sigma}$ and the strain tensor $\boldsymbol{\varepsilon}$ are related with elasticity tensor \mathbb{C} by Hooke's law

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}. \quad (3.2)$$

Typically, in a strong form, the displacement field is obtained by solving the

following boundary value problem:

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0, & \text{in } \Omega, \\ \mathbf{u} = \mathbf{u}_D, & \text{on } \Gamma_D, \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}_N, & \text{on } \Gamma_N, \end{cases} \quad (3.3)$$

where \mathbf{b} represents the body force, \mathbf{u}_D denotes the prescribed displacement on Dirichlet boundary Γ_D and \mathbf{t}_N denotes the prescribed traction on Neumann boundary Γ_N . In general, we assume that $\overline{\Gamma_D \cup \Gamma_N} = \partial\Omega$ while $\Gamma_D \cap \Gamma_N = \emptyset$.

To solve the PDEs [Equation 3.3](#) with FE methods, it is necessary to convert it to a weak form. We define the trial function space V and the test function space V_0 as

$$\begin{aligned} V &:= \{\mathbf{u} \in H^1(\Omega) : \mathbf{u} = \mathbf{u}_D \text{ on } \Gamma_D\}, \\ V_0 &:= \{\mathbf{v} \in H^1(\Omega) : \mathbf{v} = 0 \text{ on } \Gamma_D\}, \end{aligned} \quad (3.4)$$

where $H^1(\Omega)$ is a Hilbert space. The weak form is constructed by multiplying an arbitrary test function \mathbf{v} to both sides of the equilibrium equation, and then integration by parts over the domain Ω . It reads: find $\mathbf{u} \in V$ such that

$$a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}), \quad \forall \mathbf{v} \in V_0, \quad (3.5)$$

with the bilinear form $a : V \times V_0 \rightarrow \mathbb{R}$ and linear form $l : V_0 \rightarrow \mathbb{R}$ are given by

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &:= \int_{\Omega} \nabla^s \mathbf{u} : \mathbb{C} : \nabla^s \mathbf{v} \, d\Omega, \\ l(\mathbf{v}) &:= \int_{\Omega} \mathbf{b} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma_N} \mathbf{t}_N \cdot \mathbf{v} \, d\Gamma. \end{aligned} \quad (3.6)$$

Since $\mathbf{u} \in V \subset H^1(\Omega)$ and $\mathbf{v} \in V_0 \subset H^1(\Omega)$, it is convenient to measure their magnitude with the standard L^2 norm inherited from $H^1(\Omega)$, i.e.

$$\begin{aligned} \|\mathbf{u}\|_V &= \|\mathbf{u}\|_{L^2(\Omega)} := \left(\int_{\Omega} \mathbf{u}^2 \, d\Omega \right)^{\frac{1}{2}}, \\ \|\mathbf{v}\|_{V_0} &= \|\mathbf{v}\|_{L^2(\Omega)} := \left(\int_{\Omega} \mathbf{v}^2 \, d\Omega \right)^{\frac{1}{2}}. \end{aligned} \quad (3.7)$$

In linear elasticity \mathbb{C} is constant, and thus it is straightforward to verify that the bilinear form $a(\mathbf{u}, \mathbf{v})$ is continuous and coercive: there exist constants α, β such that

$$\begin{aligned} |a(\mathbf{u}, \mathbf{v})| &\leq \beta \|\mathbf{u}\|_{L^2(\Omega)} \|\mathbf{v}\|_{L^2(\Omega)}, \quad \forall (\mathbf{u}, \mathbf{v}) \in V \times V_0, \\ a(\mathbf{v}, \mathbf{v}) &\geq \alpha \|\mathbf{v}\|_{L^2(\Omega)}^2, \quad \forall \mathbf{v} \in V_0. \end{aligned} \quad (3.8)$$

Therefore, according to *Lax-Milgram Lemma* [Quarteroni and Valli \[1994\]](#), the solution \mathbf{u} exists, is unique and continuous if and only if \mathbf{b} and \mathbf{t}_N are bounded. In addition, we have the *a priori* error estimation that there exists a constant C such that

$$\|\mathbf{u}\|_{L^2(\Omega)} \leq C \left(\|\mathbf{b}\|_{L^2(\Omega)} + \|\mathbf{t}_N\|_{L^2(\Omega)} \right). \quad (3.9)$$

3.1 PGD formulation

In many practical problems in solid mechanics, model parameters such as material properties, loading locations, are difficult to obtain exactly, while only their ranges are known from previously performed experiments or from the literature. Consequently, it would be greatly helpful if we are able to obtain a parametrised solution by taking advantage of PGD. Following the standard PGD procedure for parametrised problems which are detailed in [Chinesta et al. \[2010, 2011a, 2013\]](#), we construct the generalised weak form by assuming the parameters as extra coordinates.

3.1.1 PGD generalised weak form

With the idea of considering parameters as extra coordinates, the displacement field $\mathbf{u}(\mathbf{x})$ is generalised to $\mathbf{u}(\mathbf{x}, \boldsymbol{\mu})$, where $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_m) \in \Omega_\mu$ denotes a vector of m independent parameters. Let I_{μ_j} be the range of j th parameter μ_j , assume the parametric space is Cartesian, namely $\Omega_\mu = I_{\mu_1} \times I_{\mu_2} \times \dots \times I_{\mu_m}$, we generalise the trial function space to $\mathbf{u}(\mathbf{x}, \boldsymbol{\mu}) \in V \otimes_{j=1}^m L^2(I_{\mu_j})$, and thus the test function space to $\mathbf{v}(\mathbf{x}, \boldsymbol{\mu}) \in V_0 \otimes_{j=1}^m L^2(I_{\mu_j})$.

The generalised weak form of problem [Equation 3.5](#) then reads: find the displacement field $\mathbf{u} \in V \otimes_{j=1}^m L^2(I_{\mu_j})$, such that for all:

$$\mathcal{A}(\mathbf{u}, \mathbf{v}) = \mathcal{L}(\mathbf{v}), \quad \forall \mathbf{v} \in V_0 \otimes_{j=1}^m L^2(I_{\mu_j}) \quad (3.10)$$

where the bilinear and linear forms are generalised from [Equation 3.6](#):

$$\begin{aligned} \mathcal{A}(\mathbf{u}, \mathbf{v}) &:= \int_{\Omega_\mu} a(\mathbf{u}, \mathbf{v}) d\boldsymbol{\mu} = \int_{I_{\mu_1}} \cdots \int_{I_{\mu_m}} a(\mathbf{u}, \mathbf{v}) d\mu_1 \cdots d\mu_m, \\ \mathcal{L}(\mathbf{v}) &:= \int_{\Omega_\mu} l(\mathbf{v}) d\boldsymbol{\mu} = \int_{I_{\mu_1}} \cdots \int_{I_{\mu_m}} l(\mathbf{v}) d\mu_1 \cdots d\mu_m. \end{aligned} \quad (3.11)$$

3.1.2 PGD separated representation

The general procedure of solving parametrisation problems with PGD is discussed intensively in the literature [Chinesta et al. \[2010, 2011a, 2013\]](#), [Zlotnik et al. \[2015b\]](#). Here we briefly introduce the PGD separated representation to adapt our purpose.

One of the major important issues for PGD is the separability of the generalised solution. The separated representation of model parameters, boundary conditions and/or the source terms is always required for an efficient numerical computation. In this case, the generalised weak form [Equation 3.11](#) is supposed to be separable, so that it can be factorised as:

$$\begin{aligned} \mathcal{A}(\mathbf{u}, \mathbf{v}) &:= \int_{I_{\mu_1}} \cdots \int_{I_{\mu_m}} a(\mathbf{u}, \mathbf{v}) d\mu_1 \cdots d\mu_m = \left(\int_{\Omega} \cdots d\Omega \right) \prod_{j=1}^m \left(\int_{I_{\mu_j}} \cdots d\mu_j \right), \\ \mathcal{L}(\mathbf{v}) &:= \int_{I_{\mu_1}} \cdots \int_{I_{\mu_m}} l(\mathbf{v}) d\mu_1 \cdots d\mu_m = \left(\int_{\Omega} \cdots d\Omega \right) \prod_{j=1}^m \left(\int_{I_{\mu_j}} \cdots d\mu_j \right). \end{aligned} \quad (3.12)$$

To guarantee the factorisation [Equation 3.12](#), the generalised solution for the parametrised problem is assumed to be approximated by a superposition of n modes

$$\mathbf{u}(\mathbf{x}, \boldsymbol{\mu}) \approx \mathbf{u}_{\text{PGD}}^n(\mathbf{x}, \boldsymbol{\mu}) := \sum_{i=1}^n \boldsymbol{\chi}^i(\mathbf{x}) \prod_{j=1}^m \omega_j^i(\mu_j). \quad (3.13)$$

Each mode is composed by the product of a vector-valued function $\boldsymbol{\chi}^i(\mathbf{x})$ representing the spatial displacement and m scalar functions $\omega_j^i(\mu_j)$ representing the influence of the each parameter. We also call $\boldsymbol{\chi}^i(\mathbf{x})$ the i th spatial mode and $\omega_j^i(\mu_j)$ the i th parametric mode of parameter μ_j . For notational simplicity, we will frequently neglect the dependent variables.

To obtain the functions composing each PGD mode, typically we compute the modes sequentially from the fact that

$$\mathbf{u}_{\text{PGD}}^n = \mathbf{u}_{\text{PGD}}^{n-1} + \boldsymbol{\chi}^n(\mathbf{x}) \prod_{j=1}^m \omega_j^n(\mu_j). \quad (3.14)$$

Such procedure is also called enrichment. Very often, the superscript n for current mode and the arguments are omitted for brevity.

To perform the enrichment procedure of finding the n th mode based on previously obtained $(n - 1)$ modes, we put [Equation 3.13](#) and [3.16](#) into the generalised weak form [Equation 3.10](#), resulting an equation explicitly expressed by the PGD modes

to be solved. Now the problem reads: given $\mathbf{u}_{\text{PGD}}^{n-1} = \sum_{i=1}^{n-1} \boldsymbol{\chi}^i \prod_{j=1}^m \omega_j^i$, find $\boldsymbol{\chi}^n$ and ω_j^n , ($j = 1, 2, \dots, m$), such that

$$\mathcal{A} \left(\mathbf{u}_{\text{PGD}}^{n-1} + \boldsymbol{\chi}^n \prod_{j=1}^m \omega_j^n, \mathbf{v}^n \right) = \mathcal{L}(\mathbf{v}^n), \quad \forall \mathbf{v}^n \in V_0 \bigotimes_{j=1}^m L^2(I_{\mu_j}). \quad (3.15)$$

It can be seen this is a nonlinear equation, so that a proper nonlinear solver should be used to obtain solution in an iterative scheme.

3.1.3 Alternative direction iteration

A typical solver for the nonlinear problem Equation 3.15 is a fixed-point scheme, the so-called alternative direction iteration. The idea is, during an enrichment, each time solve only one unknown mode, assuming all the others known, until reaching convergence under a given tolerance.

For the PGD generalised weak form, we select the test function as the admissible variation of n th modes, which reads

$$\mathbf{v}^n := \delta \left(\boldsymbol{\chi}^n \prod_{j=1}^m \omega_j^n \right) = \delta \boldsymbol{\chi}^n \prod_{j=1}^m \omega_j^n + \boldsymbol{\chi}^n \sum_{k=1}^m \delta \omega_k^n \prod_{j=1, j \neq k}^m \omega_j^n. \quad (3.16)$$

In general, a PGD solver contains loops in two levels: one outer loop for modal enrichment, and inside each there is an iterative loop solving the $(m+1)$ functions composing the mode. During each loop for modal search, by putting Equation 3.13 and 3.16 into the weak form Equation 3.10, we obtain one mechanical problem to solve $\boldsymbol{\chi}$, and m parametric problems to solve ω_j . In this case, we have one spatial/mechanical problem in term of function $\boldsymbol{\chi}$ and m parametric problems in terms of ω_j to be solved. The alternative direction iteration scheme reads: given $\mathbf{u}_{\text{PGD}}^{n-1} = \sum_{i=1}^{n-1} \boldsymbol{\chi}^i \prod_{j=1}^m \omega_j^i$, find the n th modes as follows:

1. Assume ω_j , ($j = 1, 2, \dots, m$) are known, thus $\mathbf{v} = \delta \boldsymbol{\chi} \prod_{j=1}^m \omega_j$, find $\boldsymbol{\chi}$ such that

$$\mathcal{A} \left(\boldsymbol{\chi} \prod_{j=1}^m \omega_j, \mathbf{v} \right) = \mathcal{L}(\mathbf{v}) - \mathcal{A}(\mathbf{u}_{\text{PGD}}^{n-1}, \mathbf{v}), \quad \forall \delta \boldsymbol{\chi} \in V_0. \quad (3.17)$$

2. For $k = 1, 2, \dots, m$, assume $\boldsymbol{\chi}$ and ω_j , ($j = 1, 2, \dots, m; j \neq k$) are known, thus $\mathbf{v} = \sum_{k=1}^m \delta \omega_k \prod_{j=1, j \neq k}^m \omega_j$, find ω_k such that

$$\mathcal{A} \left(\boldsymbol{\chi} \prod_{j=1}^m \omega_j, \mathbf{v} \right) = \mathcal{L}(\mathbf{v}) - \mathcal{A}(\mathbf{u}_{\text{PGD}}^{n-1}, \mathbf{v}), \quad \forall \delta \omega_k \in L^2(I_{\mu_k}). \quad (3.18)$$

A PGD algorithm can be constructed with a hierarchical two-loop structure. The outer loop searches for PGD modes, and the inner loop applies the alternative direction iterations solving the spatial/mechanical and parametric problems iteratively. Each of the two loops needs user-specified control variables to ensure the accuracy and robustness of the algorithm. To control a loop, either the maximum number of iterations or a stopping criterion has to be given. To implementation the algorithm, we define the amplitude of the n th mode as an indication of convergence

$$M^n := \|\boldsymbol{\chi}^n\| \prod_{j=1}^m \|\omega_j^n\|, \quad (3.19)$$

where $\|\bullet\|$ denotes the proper norm of \bullet in the corresponding space, which typically is L^2 norm. Ideally, it is expected that M^n decreases monotonically with n , however, empirical observation shows that there are frequent fluctuations during this decreasing global trend.

As we mentioned in [Chapter 1](#), PGD is an *a priori* model order reduction method. In fact, the approximation error is implicitly determined before the offline computation by the loop control variables. However, verification tools for PGD are still under development [Nadal et al. \[2015b\]](#). The PGD approximation error mainly comes from two different sources: the truncation of PGD modes which is controlled by n , and the discretisation error from underlying numerical methods such as FEM.

For the study of a raw PGD model without any *a priori* knowledge of the error bounds, it is still an unsolved issue that how to properly determine the stopping criterion for the mode searching loop [Ammar \[2010\]](#). In this work, preliminary PGD computations have been performed without limiting the maximum number of modes to search, but with limiting the number of alternative direction iterations to reduce computational costs. Consequently, an empirical knowledge about the error evolution is gained in terms from observing the behaviour of the amplitude. The number of modes n dominates the accuracy of PGD approximation, while the correction on each mode obtained from the alternative direction iteration becomes less significant when the iteration number reaches some certain problem-dependent threshold.

The typical implementation of PGD is presented in [Algorithm 3.1](#). The main loop is the search for modes, which is controlled by our choice of specifying the maximum number of modes n . The inside loop is the alternative direction iteration indexed with `iter`, and the loop is controlled by a stopping criterion which compares the given

tolerance tol , typically $\text{tol} = 10^{-3}$, with stationarity measure of the amplitude

$$\frac{|M_{(\text{iter})}^i - M_{(\text{iter}-1)}^i|}{|M_{(\text{iter}-1)}^i|} < \text{tol}. \quad (3.20)$$

To ensure the robustness, a maximum iteration number iter_{\max} is also specified. Further discussions on PGD algorithm, for instance, different approaches of defining the stopping criterion, can be found in the literature such as [Chinesta et al. \[2014\]](#).

Algorithm 3.1 Typical PGD algorithm

- 1: Initialise $\boldsymbol{\chi}$ and ω_j .
 - 2: Specify user-controlled input n , tol , iter_{\max} .
 - 3: **for** $i = 1$ to n **do**
 - 4: Initialise $\boldsymbol{\chi}^i$ and ω_j^i .
 - 5: **while** $\text{iter} < \text{iter}_{\max}$ **do**
 - 6: Solve the mechanical problem to update $\boldsymbol{\chi}^i$.
 - 7: **for** $j = 1$ to m **do**
 - 8: Solve the parametric problem to update ω_j^i .
 - 9: **end for**
 - 10: Update the amplitude $M_{(\text{iter})}^i \leftarrow \|\boldsymbol{\chi}^i\| \prod_{j=1}^m \|\omega_j^i\|$.
 - 11: Check the convergence:
 - 12: **if** $|M_{(\text{iter})}^i - M_{(\text{iter}-1)}^i| / |M_{(\text{iter}-1)}^i| < \text{tol}$ **then**
 - 13: $\text{iter} \leftarrow \text{iter}_{\max}$
 - 14: **end if**
 - 15: **end while**
 - 16: Save amplitude M^i , functions $\boldsymbol{\chi}^i$ and ω_j^i into vademecum.
 - 17: **end for**
-

3.2 FE discretisation

For a better representation, we now introduce the algebraic formulation of PGD following standard FE discretisation. Conventionally, both spatial domain and parameter spaces are discretised with FE methods.

In practice, when dealing with structural problems solved through FE methods, we express the local displacement $\mathbf{u}(\mathbf{x})$ in terms of nodal degrees of freedom (DOF) vector $\hat{\mathbf{U}}$ interpolated by the matrix consisting proper shape functions $\mathbf{N}(\mathbf{x})$, i.e., in matrix form, we have $\mathbf{u}(\mathbf{x}) = \mathbf{N}(\mathbf{x})\hat{\mathbf{U}}$. Generalising this matrix formulation by taking parameters as extra coordinates, we introduce the FE approximation for both

mechanical and parametric problems

$$\begin{aligned}\boldsymbol{\chi}(\boldsymbol{x}) &= \mathbf{N}(\boldsymbol{x})\mathbf{U}, \\ \omega_j(\mu_j) &= \mathbf{N}_j^T(\mu_j)\boldsymbol{\omega}_j, \quad j = 1, 2, \dots, m,\end{aligned}\tag{3.21}$$

where $\mathbf{N}_j(\mu_j)$ are column vectors of shape functions interpolating parameter μ_j . Thus [Equation 3.13](#) becomes

$$\mathbf{u}_{\text{PGD}}^n(\boldsymbol{x}, \mu_1, \mu_2, \dots, \mu_m) = \sum_{i=1}^n \mathbf{N}(\boldsymbol{x})\mathbf{U}^i \cdot \prod_{j=1}^m \mathbf{N}_j^T(\mu_j)\boldsymbol{\omega}_j^i,\tag{3.22}$$

where $\mathbf{N}(\boldsymbol{x})$ denotes the shape functions for the spatial discretisation, and \mathbf{U}^i denotes the spatial DOF vector, while $\mathbf{N}_j(\mu_j)$ denotes shape functions for the parameters, and $\boldsymbol{\omega}_j^i$ denotes the DOF vector for the corresponding parameter. Note that $\mathbf{N}(\boldsymbol{x})$ only depends on the spatial mesh, while $\mathbf{N}_j(\mu_j)$ depends on the discretisation of the parametric space. Accordingly, we generalise the global nodal displacement DOF vector to

$$\hat{\mathbf{U}}_{\text{PGD}}^n(\mu_1, \mu_2, \dots, \mu_m) := \sum_{i=1}^n \mathbf{U}^i \cdot \prod_{j=1}^m \mathbf{N}_j^T(\mu_j)\boldsymbol{\omega}_j^i,\tag{3.23}$$

and thus we have a formulation separating the spatial and parametric variables $\mathbf{u}_{\text{PGD}}^n(\boldsymbol{x}, \boldsymbol{\mu}) = \mathbf{N}(\boldsymbol{x})\hat{\mathbf{U}}_{\text{PGD}}^n(\boldsymbol{\mu})$. We would like to address that since the spatial shape function vector $\mathbf{N}(\boldsymbol{x})$ is exactly the same as that in standard FE formulation, the generalised global displacement DOF vector $\hat{\mathbf{U}}_{\text{PGD}}^n$ is of more interest in structural analysis other than the local solution, because it depends only on the parameters, which can be explicitly written as $\hat{\mathbf{U}}_{\text{PGD}}^n(\mu_1, \mu_2, \dots, \mu_m)$ or $\hat{\mathbf{U}}_{\text{PGD}}^n(\boldsymbol{\mu})$.

3.3 Non-intrusive scheme

In this section, we introduce in detail the non-intrusive PGD scheme with the previously established mechanical problem in linear elasticity.

From [Algorithm 3.1](#), we identify that each modal enrichment is a composition of a sequence of one mechanical problem and m parametric problems. The mechanical problems are usually more demanding of the computational resources, because the FE model can be of a large number of DOFs. During FE solution of each mechanical problem, it is necessary to compute elemental stiffness matrices element-wise, and then assemble the global stiffness matrix. In the mean time, for each parametric problem, only the relatively cheaper mass matrix is needed. Therefore, a natural idea

is to isolate the mechanical problem, in Line 5 of [Algorithm 3.1](#) which corresponds to [Equation 3.17](#), and solve it with some off-the-shelf solvers. There are a large number of existing FE packages available. In this chapter, for instance, we take advantage of Abaqus as the mechanical problem solver, and implement the non-intrusive PGD scheme with Matlab as the controlling code.

We first consider only material properties as extra coordinates, and then loading location as extra coordinates. In general, they could be considered in one problem, however, to make the problem less complicated and more clear, different type of parameters are considered separately.

3.3.1 Material properties as extra coordinates

Consider a separable domain $\Omega = \Omega_1 \cup \Omega_2$ with two different material properties, for example the Young moduli, as extra coordinates. For the sake of simplicity, we consider the homogeneous, isotropic materials with the same Poisson's ratio, so that the displacement under given boundary conditions should only depend on the two parameters, E_1 and E_2 , of each subdomain. The PGD separated representation reads

$$\hat{\mathbf{U}}_{\text{PGD}}^n(E_1, E_2) := \sum_{i=1}^n \mathbf{U}^i \omega_1^i(E_1) \omega_2^i(E_2). \quad (3.24)$$

In PGD framework, to guarantee the separability of the bilinear form in [Equation 3.10](#), conventionally it is required that the elastic tensor $\mathbb{C}(\mathbf{x}, E_1, E_2)$ should also be (approximately) separable [Zlotnik et al. \[2015b\]](#).

$$\mathbb{C}(\mathbf{x}, E_1, E_2) = \sum_{k=1}^2 \mathbb{C}^k(\mathbf{x}) \lambda_1^k(E_1) \lambda_2^k(E_2), \quad (3.25)$$

where functions $\mathbb{C}^k(\mathbf{x})$, $\lambda_1^k(E_1)$ and $\lambda_2^k(E_2)$ describe the material properties in each of the two subdomains. In this two-material case, without losing generality, it is able to define $\mathbb{C}^k(\mathbf{x})$ for each domain as

$$\mathbb{C}^k(\mathbf{x}) := \begin{cases} \mathbb{C}_1, & \text{if } \mathbf{x} \in \Omega_1, \\ \mathbb{C}_2, & \text{if } \mathbf{x} \in \Omega_2, \end{cases} \quad (3.26)$$

where \mathbb{C}_1 and \mathbb{C}_2 are constant. The functions related to Young moduli can be identified as

$$\begin{aligned} \lambda_1^1(E_1) &:= E_1, & \lambda_2^1(E_2) &:= 1, \\ \lambda_1^2(E_1) &:= 1, & \lambda_2^2(E_2) &:= E_2. \end{aligned} \quad (3.27)$$

As a result of the separation, the bilinear form will become a sum of products of the factorised integrals in [Equation 3.12](#).

In FE formulation, the elastic tensor $\mathbb{C}(\mathbf{x}, E_1, E_2)$ is discretised to the elemental elastic matrix $\mathbf{D}^e(E_1, E_2)$, and is eventually integrated to the global stiffness matrix $\mathbf{K}(E_1, E_2)$, which now may be written as

$$\mathbf{K}(E_1, E_2) = E_1 \mathbf{K}_1 + E_2 \mathbf{K}_2, \quad (3.28)$$

where \mathbf{K}_1 and \mathbf{K}_2 are stiffness-like global matrices, despite that they are independent of Young moduli. For convenience, we will keep referring to \mathbf{K}_1 and \mathbf{K}_2 as stiffness matrices when there is no ambiguity in the context.

Now we introduce the definition of following mass-like matrices:

$$\begin{aligned} \mathbf{H}_1 &:= \int_{I_{E_1}} E_1 \mathbf{N}_1 \mathbf{N}_1^T dE_1, \\ \mathbf{H}_2 &:= \int_{I_{E_2}} E_2 \mathbf{N}_2 \mathbf{N}_2^T dE_2, \\ \mathbf{M}_1 &:= \int_{I_{E_1}} \mathbf{N}_1 \mathbf{N}_1^T dE_1, \\ \mathbf{M}_2 &:= \int_{I_{E_2}} \mathbf{N}_2 \mathbf{N}_2^T dE_2, \end{aligned} \quad (3.29)$$

and following load-like vectors:

$$\begin{aligned} \mathbf{Q}_1 &:= \int_{I_{E_1}} \mathbf{N}_1 dE_1, \\ \mathbf{Q}_2 &:= \int_{I_{E_2}} \mathbf{N}_2 dE_2. \end{aligned} \quad (3.30)$$

In particular, the discretised form of [Equation 3.17](#) in this example reads: assuming $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$ and $\hat{\mathbf{U}}^{n-1} = \sum_{i=1}^{n-1} \mathbf{U}^i (\mathbf{N}_1^T \boldsymbol{\omega}_1^i) (\mathbf{N}_2^T \boldsymbol{\omega}_2^i)$ are known, solve \mathbf{U} from

$$(\tilde{E}_1 \mathbf{K}_1 + \tilde{E}_2 \mathbf{K}_2) \mathbf{U} = \tilde{Q} \mathbf{F} - \sum_{i=1}^{n-1} (\tilde{E}_1^i \mathbf{K}_1 + \tilde{E}_2^i \mathbf{K}_2) \mathbf{U}^i, \quad (3.31)$$

where the scalars are computed as follows:

$$\begin{aligned} \tilde{E}_1 &= (\boldsymbol{\omega}_1^T \mathbf{H}_1 \boldsymbol{\omega}_1) (\boldsymbol{\omega}_2^T \mathbf{M}_2 \boldsymbol{\omega}_2), \\ \tilde{E}_2 &= (\boldsymbol{\omega}_2^T \mathbf{H}_2 \boldsymbol{\omega}_2) (\boldsymbol{\omega}_1^T \mathbf{M}_1 \boldsymbol{\omega}_1), \\ \tilde{E}_1^i &= (\boldsymbol{\omega}_1^T \mathbf{H}_1 \boldsymbol{\omega}_1^i) (\boldsymbol{\omega}_2^T \mathbf{M}_2 \boldsymbol{\omega}_2^i), \quad i = 1, \dots, n-1, \\ \tilde{E}_2^i &= (\boldsymbol{\omega}_2^T \mathbf{H}_2 \boldsymbol{\omega}_2^i) (\boldsymbol{\omega}_1^T \mathbf{M}_1 \boldsymbol{\omega}_1^i), \quad i = 1, \dots, n-1, \\ \tilde{Q} &= (\mathbf{Q}_1^T \boldsymbol{\omega}_1) (\mathbf{Q}_2^T \boldsymbol{\omega}_2), \end{aligned} \quad (3.32)$$

and \mathbf{F} is the standard finite element load vector, namely

$$\mathbf{F} := \int_{\Omega} \mathbf{N}^T \mathbf{b} d\Omega + \int_{\Gamma_N} \mathbf{N}^T \mathbf{t}_N d\Gamma. \quad (3.33)$$

Note that the only unknown in Equation 3.31 is \mathbf{U} . We denote the right-hand side as a fictitious load \mathbf{F}^* ,

$$\mathbf{F}^* := \tilde{Q}\mathbf{F} - \sum_{i=1}^{n-1} (\tilde{E}_1^i \mathbf{K}_1 + \tilde{E}_2^i \mathbf{K}_2) \mathbf{U}^i, \quad (3.34)$$

and define a fictitious stiffness matrix

$$\mathbf{K}^* := \tilde{E}_1 \mathbf{K}_1 + \tilde{E}_2 \mathbf{K}_2. \quad (3.35)$$

As a result, a fictitious mechanical problem in terms of a linear system

$$\mathbf{K}^* \mathbf{U} = \mathbf{F}^* \quad (3.36)$$

is created for each mechanical problem in the alternative direction iteration. For a non-intrusive implementation, the idea is we generate \mathbf{K}^* and \mathbf{F}^* , and transfer them to an external solver which will return \mathbf{U} . Details are explained in Section 3.4.

Similarly, the discretised form of the first parametric problem Equation 3.18 with E_1 as the extra coordinate reads: assuming \mathbf{U} , $\boldsymbol{\omega}_2$ and $\hat{\mathbf{U}}^{n-1} = \sum_{i=1}^{n-1} \mathbf{U}^i \omega_1^i \omega_2^i$ are known, solve $\boldsymbol{\omega}_1$ from

$$(\tilde{M}_2 \mathbf{H}_1 + \tilde{H}_2 \mathbf{M}_1) \boldsymbol{\omega}_1 = \tilde{F}_2 \mathbf{Q}_1 - \sum_{i=1}^{n-1} (\tilde{M}_2^i \mathbf{H}_1 + \tilde{H}_2^i \mathbf{M}_1) \boldsymbol{\omega}_1^i, \quad (3.37)$$

and the scalars are computed by

$$\begin{aligned} \tilde{M}_2 &:= (\mathbf{U}^T \mathbf{K}_1 \mathbf{U}) (\boldsymbol{\omega}_2^T \mathbf{M}_2 \boldsymbol{\omega}_2), \\ \tilde{H}_2 &:= (\mathbf{U}^T \mathbf{K}_2 \mathbf{U}) (\boldsymbol{\omega}_2^T \mathbf{H}_2 \boldsymbol{\omega}_2), \\ \tilde{M}_2^i &:= (\mathbf{U}^T \mathbf{K}_1 \mathbf{U}^i) (\boldsymbol{\omega}_2^T \mathbf{M}_2 \boldsymbol{\omega}_2^i), \quad i = 1, \dots, n-1, \\ \tilde{H}_2^i &:= (\mathbf{U}^T \mathbf{K}_2 \mathbf{U}^i) (\boldsymbol{\omega}_2^T \mathbf{H}_2 \boldsymbol{\omega}_2^i), \quad i = 1, \dots, n-1, \\ \tilde{F}_2 &:= (\mathbf{U}^T \mathbf{F}) (\mathbf{Q}_2^T \boldsymbol{\omega}_2). \end{aligned} \quad (3.38)$$

To solve this problem also in a non-intrusive fashion, another linear system can be created by defining

$$\begin{aligned} \mathbf{M}_1^* &:= \tilde{M}_2 \mathbf{H}_1 + \tilde{H}_2 \mathbf{M}_1, \\ \mathbf{Q}_1^* &:= \tilde{F}_2 \mathbf{Q}_1 - \sum_{i=1}^{n-1} (\tilde{M}_2^i \mathbf{H}_1 + \tilde{H}_2^i \mathbf{M}_1) \boldsymbol{\omega}_1^i, \end{aligned} \quad (3.39)$$

so that the linear system reads

$$\mathbf{M}_1^* \boldsymbol{\omega}_1 = \mathbf{Q}_1^*. \quad (3.40)$$

It can also be sent into an external solver to obtain $\boldsymbol{\omega}_1$. In addition, the other parametric problem with E_2 as the extra coordinate can be solved in a similar procedure.

3.3.2 Loading locations as extra coordinates

In many multi-query problems, we need to obtain solutions under a source term corresponding to different loading locations. Within the PGD framework, the loading location $\mathbf{s} \in \Gamma_N$ is considered as the only extra coordinate. Considering that the spatial discretisation in Equation 3.21, the global nodal displacement vector Equation 3.23 can be written as a function of the single parameter \mathbf{s}

$$\hat{\mathbf{U}}_{\text{PGD}}^n(\mathbf{s}) := \sum_{i=1}^n \mathbf{U}^i \omega^i(\mathbf{s}). \quad (3.41)$$

We consider the simplest case in problems in linear elasticity that the source term is a point force. With the help of the Dirac delta function, it can be written as a “body force”

$$\mathbf{b}(\mathbf{x}, \mathbf{s}) = \mathbf{b}_0 \delta(\mathbf{x} - \mathbf{s}), \quad (3.42)$$

where \mathbf{b}_0 is a constant vector that denotes the point load, and $\delta(\bullet)$ denotes the Dirac delta function.

Let’s focus on the mechanical problem Equation 3.17. Assuming that the first $n - 1$ modes and the n th parametric function $\omega^n(\mathbf{s})$ are known, we now try to solve \mathbf{U}^n for the n th spatial mode. We choose the test function in Equation 3.16 as $\mathbf{v}^n = \mathbf{N}(\mathbf{x})\omega^n(\mathbf{s})$, so that the global load vector that corresponding to Equation 3.33 becomes

$$\mathbf{F}^n(\mathbf{s}) = \int_{\Omega} \mathbf{N}^T(\mathbf{x}) \mathbf{b}_0 \omega^n(\mathbf{s}) \delta(\mathbf{x} - \mathbf{s}) d\mathbf{x} = \mathbf{N}^T(\mathbf{s}) \mathbf{b}_0 \omega^n(\mathbf{s}), \quad (3.43)$$

and its PGD generalisation becomes

$$\hat{\mathbf{F}}^n = \int_{\Gamma_N} \mathbf{F}^n(\mathbf{s}) d\mathbf{s} = \int_{\Gamma_N} \mathbf{N}^T(\mathbf{s}) \mathbf{b}_0 \omega^n(\mathbf{s}) d\mathbf{s}. \quad (3.44)$$

Assume the stiffness matrix \mathbf{K} of the system is acquired, the PGD generalised weak form has a formulation with only spatial discretisation as

$$\int_{\Gamma_N} \mathbf{K} \sum_{i=1}^n \mathbf{U}^i \omega^i(\mathbf{s}) \omega^n(\mathbf{s}) d\mathbf{s} = \int_{\Gamma_N} \mathbf{N}^T(\mathbf{s}) \mathbf{b}_0 \omega^n(\mathbf{s}) d\mathbf{s}. \quad (3.45)$$

The spatial DOF \mathbf{U}^i can be obtained by solving [Equation 3.45](#) with the alternative direction iteration.

Now we need to discretise $\omega^i(\mathbf{s})$ for a numerical solution. A natural approach of discretising the parameter space of loading location is to follow the spatial FE discretisation [Cueto et al. \[2016\]](#). That is, we select admissible nodes for the point force from the spatial mesh. The discretisation can be written as

$$\omega^i(\mathbf{s}) = \mathbf{N}_s^T(\mathbf{s})\boldsymbol{\omega}^i, \quad (3.46)$$

where the dimension of the DOF vector $\boldsymbol{\omega}^i$ and that of the corresponding shape function vector $\mathbf{N}_s(\mathbf{s})$ both equal to the number of selected nodes admissible for loading.

As before, by putting [Equation 3.46](#) into [Equation 3.45](#), integration over the parametric space Γ_N will result a mass-like matrix such as [Equation 3.29](#). For the left-hand side, we have

$$\mathbf{M} = \int_{\Gamma_N} \mathbf{N}_s(\mathbf{s})\mathbf{N}_s^T(\mathbf{s}) d\mathbf{s}. \quad (3.47)$$

However, the right-hand side will include a mass-like matrix which is not necessarily a squared matrix because it involves the shape functions of both spatial and parametric discretisation that may have different dimensions. This matrix has to be computed intrusively in the in-house code. We denote it as

$$\hat{\mathbf{M}} = \int_{\Gamma_N} \mathbf{N}^T(\mathbf{s})\mathbf{b}_0\mathbf{N}_s^T(\mathbf{s}) d\mathbf{s}. \quad (3.48)$$

Now the fully discretised PGD alternative direction iteration formulation reads:

1. Mechanical problem: Assume $\boldsymbol{\omega}$ and $\hat{\mathbf{U}}^{n-1} = \sum_{i=1}^{n-1} \mathbf{U}^i(\mathbf{N}_s^T\boldsymbol{\omega}^i)$ are known, solve \mathbf{U} from

$$(\boldsymbol{\omega}^T \mathbf{M} \boldsymbol{\omega}) \mathbf{K} \mathbf{U} = \hat{\mathbf{M}} \boldsymbol{\omega} - \sum_{i=1}^{n-1} (\boldsymbol{\omega}^T \mathbf{M} \boldsymbol{\omega}^i) \mathbf{K} \mathbf{U}^i. \quad (3.49)$$

2. Parametric problem: Assume \mathbf{U} and $\hat{\mathbf{U}}^{n-1} = \sum_{i=1}^{n-1} \mathbf{U}^i(\mathbf{N}_s^T\boldsymbol{\omega}^i)$ are known, solve $\boldsymbol{\omega}$ from

$$(\mathbf{U}^T \mathbf{K} \mathbf{U}) \mathbf{M} \boldsymbol{\omega} = \hat{\mathbf{M}}^T \mathbf{U} - \sum_{i=1}^{n-1} (\mathbf{U}^T \mathbf{K} \mathbf{U}^i) \mathbf{M} \boldsymbol{\omega}^i. \quad (3.50)$$

It is straightforward to see that the non-intrusive scheme still fits the solution, except that $\hat{\mathbf{M}}$ has to be created intrusively.

In addition, we would like to point out that proper selection of the location of DOF for \mathbf{s} is important for avoiding interpolative instabilities as investigated in Zlotnik et al. [2015a]. Although the selected loading locations may not necessarily be the subset of the spatial nodes, it is optimal to do so. This is due to the introduction of the Dirac delta function in Equation 3.42. When a different discretisation mesh is used on the selected admissible loading area, it is always necessary to distribute this singular function to the related nodes on the spatial mesh, otherwise the equivalent spatial problem would become faulty because point loads are not applied on consistent nodes but on element faces.

Remark 3.1 (Geometrical parameters as extra coordinates). *In general, geometrical parameters can also be considered as extra coordinates in the non-intrusive PGD scheme. Due to the geometrical complexity of biological objects, a robust geometrical parametrisation framework is still an open question, which is beyond the scope of this paper. Focusing on the community of reduced order modelling, the interesting approaches that the authors would like to mention includes the kPCA-based manifold learning method investigated in González et al. [2018] and the non-uniform rational B-spline (NURBS) parametrisation discussed in Al Akhras et al. [2017].*

3.4 Matlab–Abaqus implementation

We provide the detailed implementation of non-intrusive PGD scheme in this section, taking the case of material properties as extra coordinates for example.

In the non-intrusive manner, taking advantage of the off-the-shelf code such as Abaqus, the computation of element stiffness matrices \mathbf{K}^e and the assembly of global stiffness matrix

$$\mathbf{K} = \mathop{\bigwedge}\limits_e \mathbf{K}^e \tag{3.51}$$

is automatically performed with specified values of (E_1, E_2) and the mapping information about their corresponding subdomains. Ideally, we should be able to obtain \mathbf{K}_1 by specifying $(E_1, E_2) = (1, 0)$ and \mathbf{K}_2 by $(E_1, E_2) = (0, 1)$. However, the limitation from Abaqus that Young modulus must always be positive prevents this operation. A trick to walk around is to replace zero with a negligible value η to the machine precision, such as $\eta = 10^{-36}$.

Now we take $(E_1, E_2) = (1, \eta)$ as an example input to explain the creation of global stiffness matrix \mathbf{K}_1 . As a typical Abaqus implementation, the generation and output of global stiffness matrix is requested by adding an extra step in the input file, e.g., `job.inp`, with following commands:

```
*STEP
*MATRIX GENERATE, STIFFNESS
*MATRIX OUTPUT, STIFFNESS, FORMAT=COORDINATE
*END STEP
```

After running Abaqus with input file `job.inp`, the stiffness matrix \mathbf{K}_1 is generated by means of two files: a plain text file `job_STIF1.mtx` and a binary file `job_X1.sim`. The data format in the plain text file `job_STIF1.mtx` perfectly matches Matlab format for sparse matrix, so it works as the interface for data exchange from Abaqus to Matlab. The binary file, representing the same matrix, can be reused by Abaqus with a scale factor `sf`, which enables setting of the scalars in Equation 3.32 such as `sf_1 = \tilde{E}_1` and `sf_2 = \tilde{E}_2` . Another stiffness matrix \mathbf{K}_2 can also be obtained in the same way by prescribing $(E_1, E_2) = (\eta, 1)$, and resulting two files `job_STIF2.mtx` and `job_X2.sim`. Note that the stiffness matrices \mathbf{K}_1 and \mathbf{K}_2 need only be generated once for all, and be stored for later use.

During the solution of each mechanical problem, the fictitious stiffness matrix \mathbf{K}^* in Equation 3.35 can be generated through the following commands in another input file:

```
*MATRIX INPUT, NAME=stiff_1, INPUT=job_X1.sim, MATRIX=STIFFNESS,
SCALE FACTOR=sf_1
*MATRIX INPUT, NAME=stiff_2, INPUT=job_X2.sim, MATRIX=STIFFNESS,
SCALE FACTOR=sf_2
*MATRIX ASSEMBLE, STIFFNESS=stiff_1
*MATRIX ASSEMBLE, STIFFNESS=stiff_2
```

The standard load vector \mathbf{F} in Equation 3.33 can either be read from the Abaqus input file or be written from Matlab data to plain text files. The fictitious load vector \mathbf{F}^* has to be computed within Matlab according to Equation 3.34, and output to a plain text file, then subsequently be included into the Abaqus input file. In this manner, the plain text file establishes the interface for load data transfer.

In practical construction of the fictitious mechanical problem, only $(\tilde{E}_1, \tilde{E}_2)$ and \mathbf{F}^* need to be computed and output by Matlab through plain text files. The linear system is written to an Abaqus input file, say `job.inp`, and sent to Abaqus solver to obtain spatial DOF \mathbf{U} by the following Matlab command:

```
system('abaqus job=job interactive')
```

There are several ways of transferring \mathbf{U} from Abaqus back to Matlab. The simplest one is to require Abaqus directly output the data into a text file, e.g, `job.fil`. One can also use a Python script to extract data from Abaqus output database, e.g, `job.odb`, and write the data into a text file. Here again, this text file works as the interface for transferring the displacement DOFs. The former approach is chosen for its higher execution speed.

The flowchart of current implementation of the non-intrusive PGD scheme is illustrated in [Figure 3.1](#). In this case, the parametric problems, which are usually computationally inexpensive, are solved within the in-house developed Matlab code, while the fictitious mechanical problems, which are supposed to be much more expensive, are solved by Abaqus solver.

An example of Matlab–Abaqus implementation for the non-intrusive PGD scheme can be downloaded through <https://github.com/xizou/NIPGD>.

For the case of loading location as the extra coordinate, the implementation is still available after slight modifications. For example, [Equation 3.48](#) has to be implemented in Matlab code.

3.5 Numerical examples

3.5.1 Example I: A 1D problem with material properties as extra coordinates

In this simple example, we briefly present an instance of non-intrusive PGD implementation with material properties as extra coordinates.

Consider a single rod with two portions (l_1 and l_2) made of different materials (E_1 and E_2) as shown in [Figure 3.2](#). Assume the cross section area is uniform and has unit value, that is $A_1 = A_2 = 1$. Discretise the problem with two rod elements for

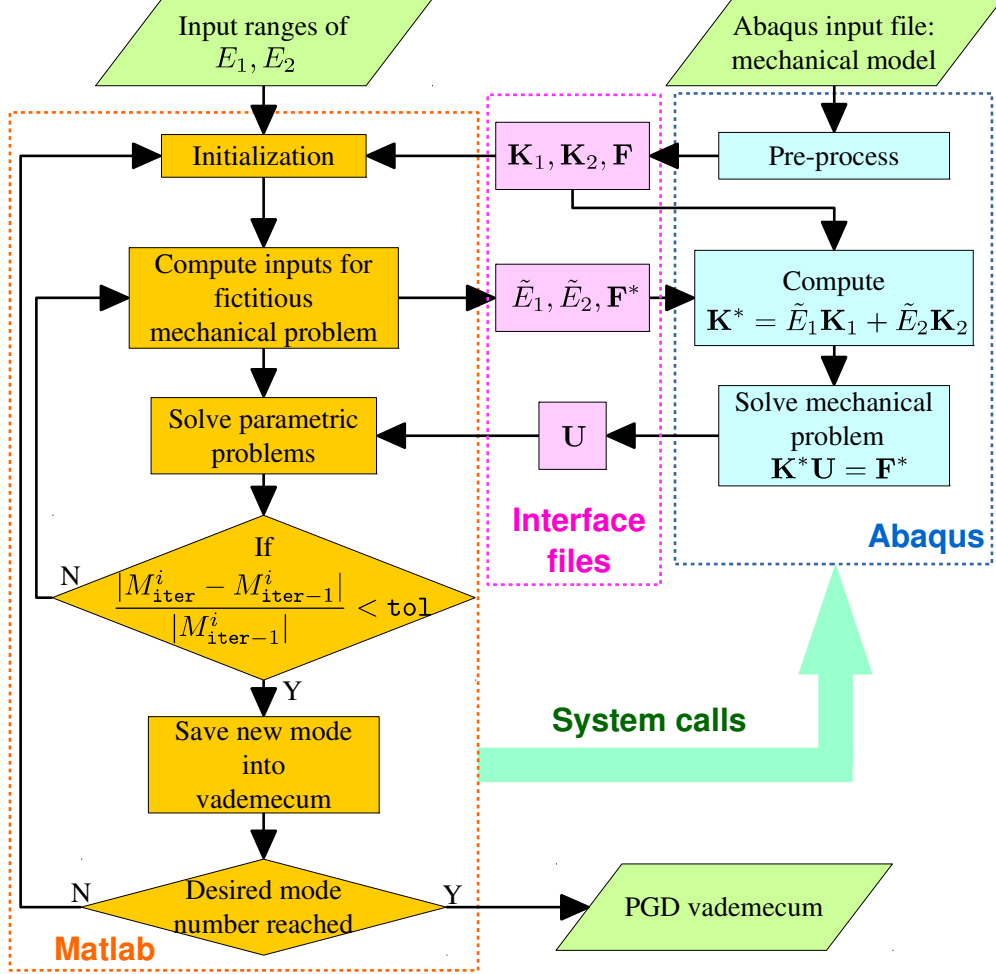


Figure 3.1: Flowchart of non-intrusive PGD implementation with material properties as extra coordinates.

both portions, it is straightforward to write the exact solution as

$$\mathbf{U}_{\text{ex}}(E_1, E_2) = \begin{bmatrix} U_1 \\ U_2 \\ 0 \end{bmatrix} = F \begin{bmatrix} \frac{l_1}{E_1} + \frac{l_2}{E_2} \\ \frac{l_2}{E_2} \\ 0 \end{bmatrix} = \begin{bmatrix} F \\ 0 \\ 0 \end{bmatrix} \frac{1}{E_1} l_1 + \begin{bmatrix} F \\ F \\ 0 \end{bmatrix} l_2 \frac{1}{E_2}. \quad (3.52)$$

The PGD solution is obtained following [Subsection 3.3.1](#), which reads

$$\hat{\mathbf{U}}_{\text{PGD}}^n(E_1, E_2) = \sum_{i=1}^n \mathbf{U}^i \omega_1^i(E_1) \omega_2^i(E_2). \quad (3.53)$$

It can be seen that the exact solution in [Equation 3.52](#) is readily separable in a PGD fashion. However, the dependency of E_1 or E_2 is nonlinear. In practice, 12 modes are needed to meet prescribed error tolerance as shown in [Figure 3.3](#). The results for U_1

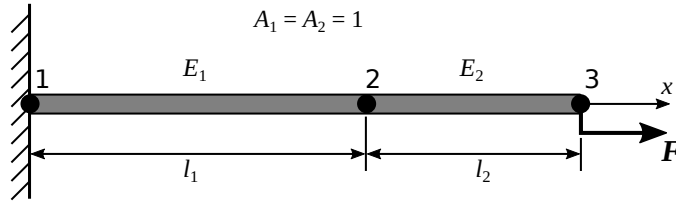


Figure 3.2: Sketch of the 1D problem with material properties as extra coordinates.

and U_2 are plotted in [Figure 3.4](#) along with relative errors. The maximum relative error for U_1 is 2.8157% while that for U_2 is 4.1761%.

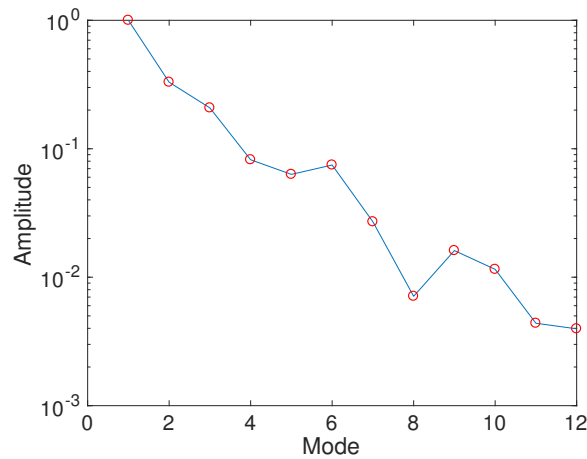


Figure 3.3: Amplitude of each PGD mode for the 1D problem with material properties as extra coordinates.

3.5.2 Example II: A 1D model problem with loading locations as extra coordinates

In this simple example, we present an instance of non-intrusive PGD implementation with loading location as extra coordinates, and compare the ways of obtaining the reduced order solution.

As depicted in [Figure 3.5](#), consider an elastic bar with left end Point 1 ($x = 0$) fixed and right end Point 3 ($x = l$) free, and loaded by F at Point 2 ($x = s$ with $s \in [s_1, s_2]$). The displacement u is a function of the querying location x and the

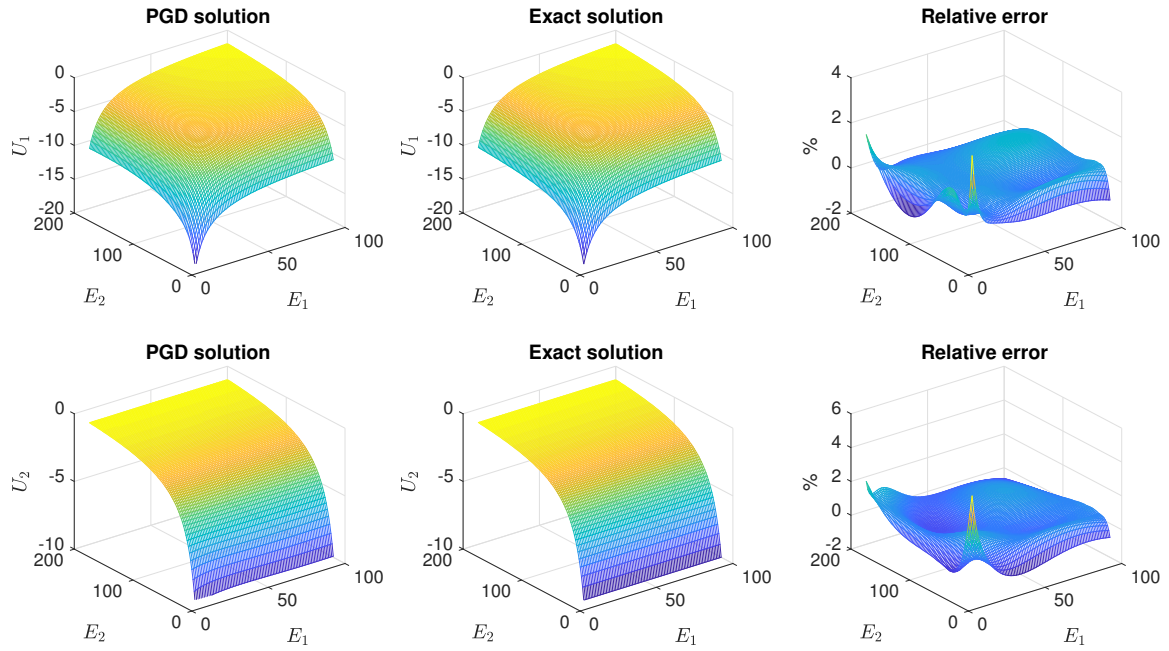


Figure 3.4: Results of nodal displacement for the 1D problem with material properties as extra coordinates.

loading location s , i.e., $u = u(x; s)$. The strong form of the problem reads

$$\left\{ \begin{array}{l} \frac{d}{dx} \left(EA \frac{du}{dx} \right) + f = 0, \quad \text{in } \Omega = (0, l), \\ u(0) = 0, \quad EA \frac{du}{dx}(l) = 0, \\ f = F\delta(x - s). \end{array} \right. \quad (3.54)$$

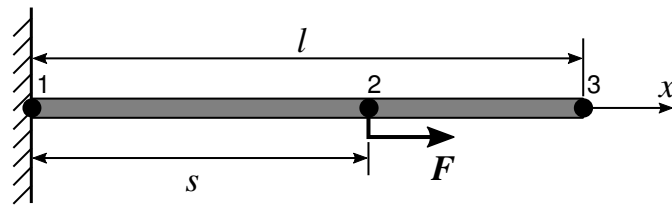


Figure 3.5: Sketch of the 1D problem with loading location as extra coordinates.

The exact solution to problem [Equation 3.54](#) is

$$u(x, s) = \begin{cases} \frac{Fx}{EA}, & \text{if } x < s, \\ \frac{Fs}{EA}, & \text{if } x \geq s. \end{cases} \quad (3.55)$$

The PGD separated representation reads

$$u(x, s) \approx u_{\text{PGD}}^n = \sum_{i=1}^n U^i(x) \omega^i(s). \quad (3.56)$$

The extended weak form reads

$$\int_{s_1}^{s_2} \int_0^l EA \frac{dv}{dx} \frac{du}{dx} dx ds = \int_{s_1}^{s_2} \int_0^l v f dx ds, \quad (3.57)$$

where $v = u^*$ is the variation. Discretise the functions as $U^i(x) = \mathbf{N}^T(x) \mathbf{U}^i$ and $\omega^i(s) = \mathbf{N}_s^T(s) \boldsymbol{\omega}^i$, we obtain the algebraic form according to the variation. Note we select the admissible loading locations from the discretised spatial nodes, so shape functions for loading locations $\mathbf{N}_s(s)$ is a subset of spatial shape functions $\mathbf{N}(x)$. Let $v = \mathbf{N}(x) \cdot \mathbf{N}_s^T(s) \boldsymbol{\omega}^i$, the fully discretised PGD alternative direction iteration formulation is exactly the same as [Equation 3.49](#) and [3.50](#), and the mass-like matrices are defined by

$$\begin{aligned} \mathbf{M} &= \int_{s_1}^{s_2} \mathbf{N}_s(s) \mathbf{N}_s^T(s) ds, \\ \hat{\mathbf{M}} &= \int_{s_1}^{s_2} F \mathbf{N}(s) \mathbf{N}_s^T(s) ds. \end{aligned} \quad (3.58)$$

For this particular problem, we set $l = 100$, $s_1 = 50$ and $s_2 = 75$. The bar is discretised into a 100-element mesh, from which 25 elements are extracted to construct the parametric space, so the non-squared mass matrix $\hat{\mathbf{M}}$ has a dimension of 101×26 . The generalised parametric problem is 2D, according to [Chinesta et al. \[2014\]](#), both SVD and PGD approach can be used to obtain the reduced basis. We may construct a matrix by concatenating the n solutions as columns, while the number of rows depends on the degrees of freedom in the problem. It is optimal to perform singular value decomposition (SVD) on the matrix to obtain the reduced order solution. Meanwhile, one may also obtain the suboptimal reduced order solution with PGD. In [Figure 3.6](#), we illustrate both results with the decreasing curve of amplitude versus modes. It demonstrates that the PGD solution is close to the SVD solution, and is suboptimal than the SVD solution which is optimal. The drastic drops occurring both after the 26th mode indicate that the real order of solution for this model problem is 26, which agrees with the discretisation of the parametric space. The comparison between PGD solution and the exact solution is shown in [Figure 3.7](#). With 26 PGD modes, the maximum relative error is 0.27%.

Remark 3.2 (Non-intrusive computation for the matrices). *Note that the mass and stiffness matrices need only compute once for all. It is possible to use Abaqus to*

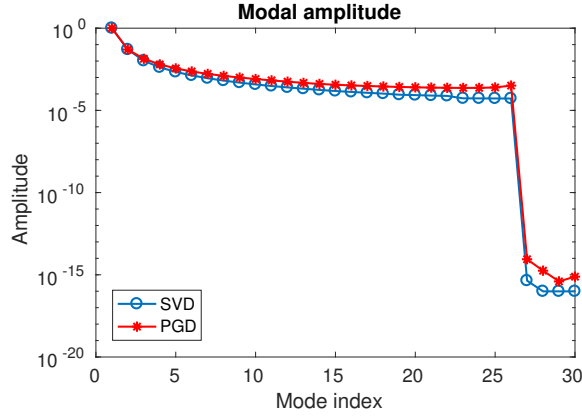


Figure 3.6: Comparison of PGD and SVD solutions. The modal amplitudes drastically drop after the 26th mode.

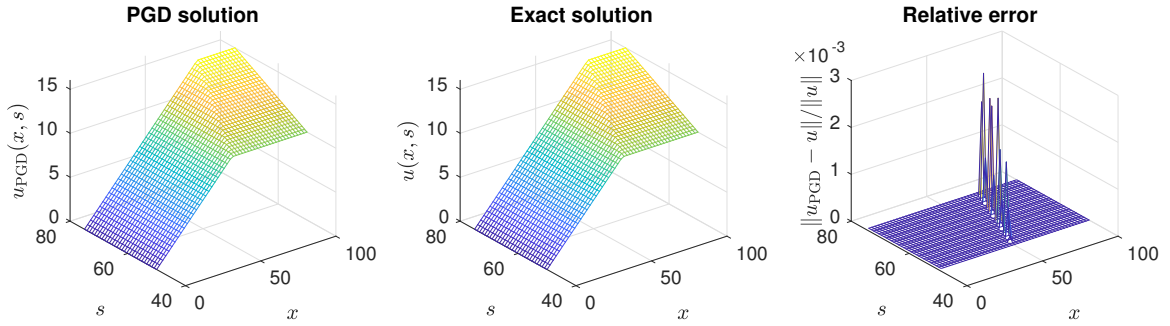


Figure 3.7: Comparison of the exact solution and PGD solution with 26 modes.

compute the stiffness matrix. In principle, it should also be possible to use Abaqus to compute the mass matrix for the discretised parametric space, however, Abaqus can only output lumped mass matrix for linear elements. To obtain consistent mass matrix, we have to compute it with in-house codes. Besides, the non-squared mass matrix has to be computed with in-house codes.

3.5.3 Example III: A 2D problem with material properties as extra coordinates

In this example, we apply PGD on a 2D elasticity problem with a single homogeneous isotropic material, taking both Young modulus E and Poisson's ratio ν as extra coordinates for a parametric study. We consider a rectangular beam in plane stress state as shown in Figure 3.8. The beam is supported on the right edge, and a distributed force with total amplitude P is applied on the left edge. The boundary

condition described in displacement components (u, v) reads

$$u(L, y) = v(L, 0) = 0. \quad (3.59)$$

In the literature [Zienkiewicz et al. \[2013\]](#) the exact solution of $u(x, y)$ and $v(x, y)$ is provided, which reads

$$\begin{aligned} u(x, y) &= -\frac{Py}{EI} \left[\frac{x^2 - L^2}{2} - (2 + \nu) \frac{y^2 - c^2}{6} \right], \\ v(x, y) &= \frac{P}{EI} \left[\frac{\nu xy^2}{2} + \frac{x^3 - L^3}{6} - (x - L) \left(\frac{4 + 5\nu}{6} c^2 + \frac{L^2}{2} \right) \right], \end{aligned} \quad (3.60)$$

where $I = 2tc^3/3$ is the area moment of inertia, t is a constant beam thickness. For a better comparison, we investigate the vertical displacement v_0 of the point at the centre of left edge, thus according to [Equation 3.60](#), we have

$$v_0(E, \nu) := v(0, 0) = \frac{PL}{6EI} [2L^2 + (4 + 5\nu)c^2]. \quad (3.61)$$

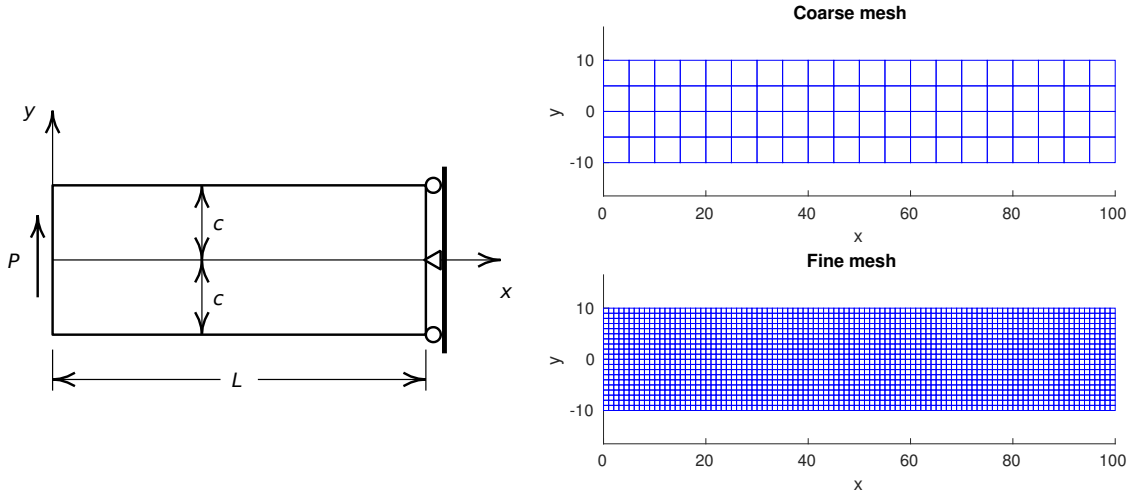


Figure 3.8: End loaded beam. Left: geometry; Right: spatial meshes.

For the numerical solution, we choose $c = 10$, $L = 100$, $t = 1$, $P = 80$ as the fixed parameters, while $E \in I_E = (800, 1200)$ and $\nu \in I_\nu = (0, 0.5)$ as the PGD extra coordinates.

The spatial domain is discretised with quadrilateral elements, as illustrated in [Figure 3.8](#). Both a coarse mesh and a fine mesh have been created to reveal the discretisation error. Detailed PGD formulation can be found in [Appendix B](#). The parametric domains are discretised with 100 nodes for each parameter and for both

meshes. We have obtained the parametric solution $v(E, \nu)$ via the PGD framework, and the results are presented in [Figure 3.10](#). For each spatial mesh, using the same stopping criterion, 8 modes have been computed, which demonstrate a drastic drop of the normalised modal amplitude from 1 to 10^{-6} , as shown in [Figure 3.9](#). It can be seen that the spatial discretisation does not affect the PGD modes significantly. As a comparison between the PGD solution $v(E, \nu)$ and exact solution $v_0(E, \nu)$, the relative error surfaces are computed with $(v - v_0)/v_0$ over the parameter domain $I_E \times I_\nu = (800, 1200) \times (0, 0.5)$. From [Equation 3.61](#) one can identify that $v_0(E, \nu)$ hyperbolically dependent on E and linearly dependent on ν , and from [Figure 3.10](#) we can see the dependencies have been well approximated by $v(E, \nu)$. In addition, as it is expected, the fine mesh improves the relative error over the coarse mesh from about 3.4% to about 0.7%.

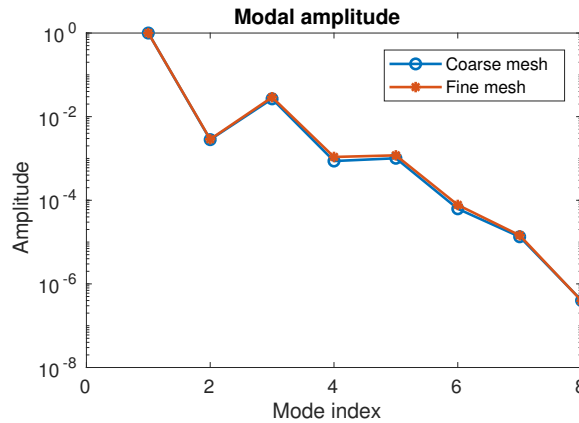


Figure 3.9: Comparison of the PGD modal amplitudes with two different spatial meshes.

3.6 Summary

As a main contribution of this thesis, the non-intrusive PGD scheme is introduced in this chapter with detailed implementations. Numerical examples are also presented demonstrating the effectiveness of the scheme.

Moreover, both spatial and parametric discretisations are performed with FE method, for the sake of the non-intrusive implementations. For further investigation, this could be improved with other discretisation methods but with some intrusive but novel implementations. In [Chapter 6](#), NURBS discretisation is used for the parameter space following an IGA fashion.

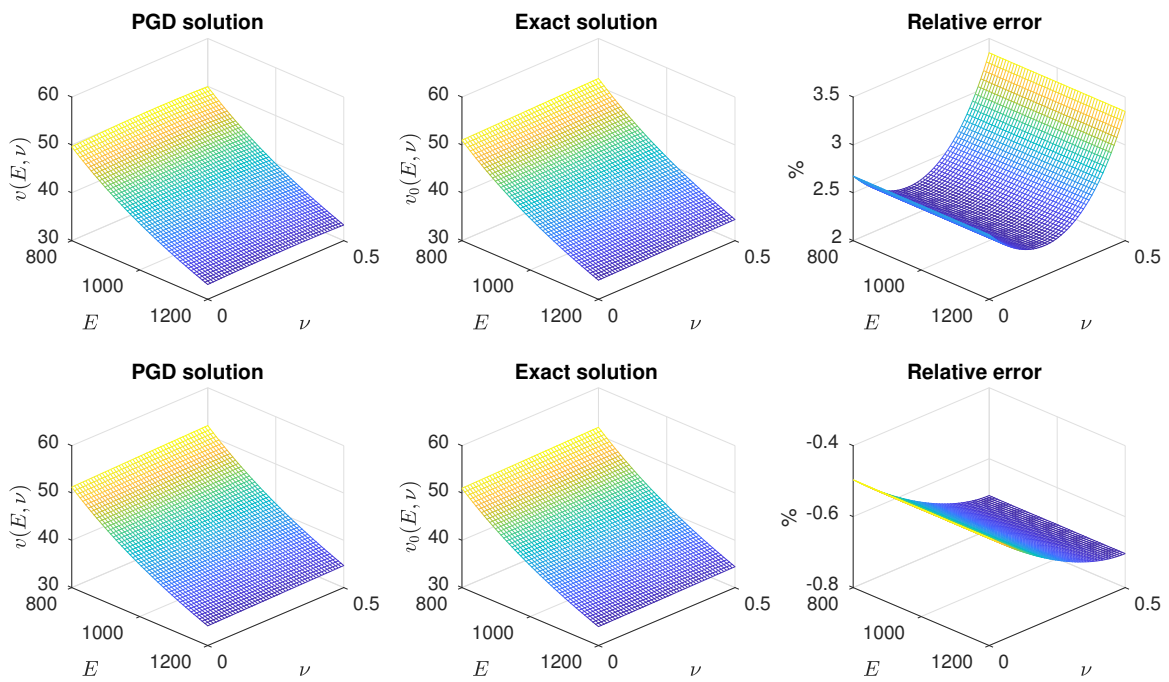


Figure 3.10: Vertical displacement result comparison of the selected location. Top: from coarse spatial mesh; Bottom: from fine spatial mesh.

Chapter 4

PGD for constrained parameter space

This chapter is dedicated to the introduction of a type of specific cases for PGD when the parameter space of parametrised PDEs is subjected to some constraints. Two common types, both physical constraints and geometrical constraints, are considered. Although it is not conceptually innovative for the PGD community to separate the parameters in a collective fashion, this strategy is seldom implemented on the parameter domain and not seen in the literature so far.

Collective separation of spatial domain is natural in structural analysis, especially in problems involving plates and shells. The membrane and transverse dimension is commonly separated in PGD solutions since they show significantly different mechanical behaviour when carrying bending load [Bognet et al. \[2012\]](#), [Giner et al. \[2013\]](#).

In conventional PGD frameworks, typically the parameters are assumed to be independent to each other, and the parameter space Ω_μ is assumed to be Cartesian [Chinesta and Ladevèze \[2014\]](#), i.e., $\Omega_\mu = I_{\mu_1} \times \cdots \times I_{\mu_m}$, so that the parametric modes in the separated representation can be written in terms of a product of m functions, each function is a 1D map from the range of the parameter to a scalar, namely $\omega_j : I_{\mu_j} \rightarrow \mathbb{R}$. Thanks to such an assumption, the parametric problems to be solved in the alternative direction iterations during each modal search are all 1D, which enables fast computations.

On the contrary, in real situations, the parameters can have mutual dependencies, due to the constraints from limitations of physical, geometrical, and/or other aspects. Those correlations may cause the parameter space not being Cartesian. The constrained parameter space has much lower separability, which could cause the separated representation ineffective. That is, the factorisation [Equation 3.12](#) may not be able to perform. To solve this problem, unlike the conventional PGD approach of separating the parameter space into tensor products of 1D subspaces, the separation strategy for parameter spaces under constraints is to collect the most correlated parameters and keep them in 2D/3D spaces during the separation, in order to respect the constraints and to avoid solutions with non-physical parameters.

In addition, to obtain more accurate derivatives with respect to the parameters, such as sensitivities in a parameter study, it is possible to replace FE discretisation with NURBS discretisation in an isogeometric analysis fashion. Thanks to its capability of describing exact geometric shape and providing highly smoothness, NURBS discretisation improves the PGD vademecum with little effort of modifying the code from original FE discretisation.

4.1 Physical constraints

Now we use a practical example in [Chapter 2](#) to explain the physical constraints of the parameter space. An example will follow, providing a comparison between the different strategies of separating the parameter space.

4.1.1 PGD formulation for Young modulus mapped from CT image

Recall that a segmentation procedure is performed to filter the voxels in CT images, leaving only ones with positive densities. After the segmentation, we have obtained $\rho \in [1, 3071]$, with ρ being integer.

According to [Equation 2.3](#), we have two parameters, α and β , to be determined. In PGD formulation, the two parameters are considered as extra coordinates. Thus the Young modulus is generalised from $E(\mathbf{x})$ to $E(\mathbf{x}, \alpha, \beta)$. Thanks to the linearity of the density-modulus mapping, Young modulus is separable in this form:

$$E(\mathbf{x}, \alpha, \beta) = \sum_{k=1}^2 G_k(\mathbf{x})R_k(\alpha)S_k(\beta), \quad (4.1)$$

where it is straightforward to identify

$$\begin{aligned} R_1(\alpha) &= \alpha, & S_1(\beta) &= 1, & G_1(\mathbf{x}) &= \rho(\mathbf{x}), \\ R_2(\alpha) &= 1, & S_2(\beta) &= \beta, & G_2(\mathbf{x}) &= 1. \end{aligned} \quad (4.2)$$

In standard finite element methods, the element stiffness matrix \mathbf{K}^e is computed as

$$\mathbf{K}^e := \int_{\Omega} \mathbf{B}^T \mathbf{D}^e \mathbf{B} d\Omega = \int_{\Omega} E^e \mathbf{B}^T \hat{\mathbf{D}} \mathbf{B} d\Omega = \int_{\Omega} (\alpha \rho^e + \beta) \mathbf{B}^T \hat{\mathbf{D}} \mathbf{B} d\Omega, \quad (4.3)$$

where $\mathbf{B} := \nabla^s \mathbf{N}$ is the displacement-strain matrix, \mathbf{D}^e is the element elastic matrix:

$$\mathbf{D}^e := \frac{E^e}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}-\nu \end{bmatrix}. \quad (4.4)$$

For the brevity of formulas, we define a matrix $\hat{\mathbf{D}} := \mathbf{D}^e/E^e$. In linear elastic case with a given Poisson's ratio ν , $\hat{\mathbf{D}}$ is constant. Now the the element stiffness matrix is separated as

$$\mathbf{K}^e = \alpha \mathbf{K}_1^e + \beta \mathbf{K}_2^e, \quad (4.5)$$

with

$$\begin{aligned} \mathbf{K}_1^e &:= \int_{\Omega} \rho^e \mathbf{B}^T \hat{\mathbf{D}} \mathbf{B} d\Omega, \\ \mathbf{K}_2^e &:= \int_{\Omega} \mathbf{B}^T \hat{\mathbf{D}} \mathbf{B} d\Omega. \end{aligned} \quad (4.6)$$

It is clear that when assembled into the global stiffness matrix, \mathbf{K}_1^e contributes to the inhomogeneous part, while \mathbf{K}_2^e consists the homogeneous part. Note the similarity between [Equation 4.5](#) and [3.28](#), the computation procedure discussed in [Subsection 3.3.1](#) well suits this problem.

However, due to the CT imaging mechanism that stronger tissue results higher intensity, apparently we have $\alpha > 0$. In addition, physically Young modulus cannot be negative, the parameters α and β must satisfy following conditions:

$$\begin{cases} \alpha > 0, \\ \alpha + \beta > 0. \end{cases} \quad (4.7)$$

Therefore, we have obtained a problem with parameters $\boldsymbol{\mu} = (\alpha, \beta)$ living in a constrained 2D parametric domain Ω_c , as shown in [Figure 4.1](#).

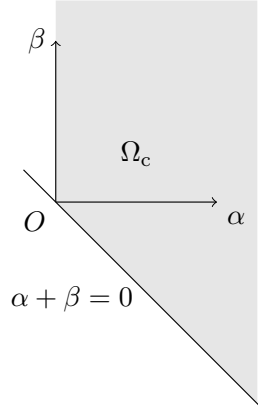


Figure 4.1: Shape of the constrained parametric domain.

Recalling [Equation 3.23](#), the separated representation for the nodal displacement vector reads

$$\hat{\mathbf{U}}_{\text{PGD}}(\boldsymbol{\mu}) := \sum_{i=1}^n \mathbf{U}^i \omega^i(\boldsymbol{\mu}). \quad (4.8)$$

The procedure is almost same as [Subsection 3.3.1](#) to obtain the corresponding computational vademecum $\hat{\mathbf{U}}_{\text{PGD}}(\boldsymbol{\mu})$ by applying the non-intrusive PGD scheme with FE discretisation of the 2D constrained parameter space $\Omega_c \subset \mathbb{R}^2$. Let $\boldsymbol{\mu} = (E_1, E_2) \in \Omega_c$, now [Equation 3.31](#) becomes: assuming $\omega(\boldsymbol{\mu})$ and $\hat{\mathbf{U}}_{\text{PGD}}^{n-1} = \sum_{i=1}^{n-1} \mathbf{U}^i \omega^i(\boldsymbol{\mu})$ are known, solve \mathbf{U} from

$$(\tilde{E}_1 \mathbf{K}_1 + \tilde{E}_2 \mathbf{K}_2) \mathbf{U} = \tilde{Q} \mathbf{F} - \sum_{i=1}^{n-1} (\tilde{E}_1^i \mathbf{K}_1 + \tilde{E}_2^i \mathbf{K}_2) \mathbf{U}^i, \quad (4.9)$$

where the scalars are computed as follows:

$$\begin{aligned} \tilde{E}_1 &:= \int_{\Omega_c} E_1 \omega^2(\boldsymbol{\mu}) d\boldsymbol{\mu}, \\ \tilde{E}_2 &:= \int_{\Omega_c} E_2 \omega^2(\boldsymbol{\mu}) d\boldsymbol{\mu}, \\ \tilde{E}_1^i &:= \int_{\Omega_c} E_1 \omega(\boldsymbol{\mu}) \omega^i(\boldsymbol{\mu}) d\boldsymbol{\mu}, \quad i = 1, \dots, n-1, \\ \tilde{E}_2^i &:= \int_{\Omega_c} E_2 \omega(\boldsymbol{\mu}) \omega^i(\boldsymbol{\mu}) d\boldsymbol{\mu}, \quad i = 1, \dots, n-1, \\ \tilde{Q} &:= \int_{\Omega_c} \omega(\boldsymbol{\mu}) d\boldsymbol{\mu}. \end{aligned} \quad (4.10)$$

On one hand, the integrals are computed repeatedly during the loops, so additional computational costs may be introduced since $\dim \Omega_c = 2$. On the other hand, we now have more options to discretise the parameter space with different meshes to improve

the accuracy. For example, one may use h -refinement to reduce the discretisation error.

When the parameters $\boldsymbol{\mu}$ are limited in a constrained space Ω_c , such a technique would lead to the total non-intrusive PGD scheme described in [Algorithm 4.1](#). In this total non-intrusive scheme, the main code acts only as the data collector and flow controller, while the solution of equations are performed by using the exterior solver as a black box.

Algorithm 4.1 Fully non-intrusive PGD algorithm

```

1: Create spatial mesh and parametric mesh for the model.
2: Transfer the definition of the model into main code.
3: Initialise  $\boldsymbol{\chi}(\boldsymbol{x})$  and  $\omega(\boldsymbol{\mu})$ .
4: for  $i = 1$  to  $n$  do
5:   Initialise  $\boldsymbol{\chi}^i(\boldsymbol{x})$  and  $\omega^i(\boldsymbol{\mu})$ .
6:   while  $\text{iter} < \text{iter}_{\max}$  do
7:     Export and solve the mechanical problem in external solver.
8:     Import the solution into main code to update  $\boldsymbol{\chi}^i(\boldsymbol{x})$ .
9:     Export and solve the parametric problem in external solver.
10:    Import the solution into main code to update  $\omega^i(\boldsymbol{\mu})$ .
11:    Update the amplitude  $M_{(\text{iter})}^i \leftarrow \|\boldsymbol{\chi}^i(\boldsymbol{x})\| \cdot \|\omega^i(\boldsymbol{\mu})\|$ .
12:    Check the convergence:
13:    if  $|M_{(\text{iter})}^i - M_{(\text{iter}-1)}^i| / |M_{(\text{iter}-1)}^i| < \text{tol}$  then
14:       $\text{iter} \leftarrow \text{iter}_{\max}$ 
15:    end if
16:  end while
17:  Save functions  $\boldsymbol{\chi}^i(\boldsymbol{x})$  and  $\omega^i(\boldsymbol{\mu})$  into vademecum.
18: end for

```

Remark 4.1 (On the solving of parametric problems). *The parametric problems, as represented in Line 9 in [Algorithm 4.1](#), does not contain derivatives with respect to the unknowns, which suggests that it is not always necessary to solve the parametric problems with the Galerkin-based FE method.*

In the non-intrusive implementation of the PGD algorithm, each call of the external solver will introduce extra cost of computation time, because upon each call, the external solver (e.g. Abaqus) will perform its initialisation.

In this work, the parametric problems for all the examples are solved within the in-house Matlab code still with FE discretisation, despite that they can also be solved by calling Abaqus.

4.1.2 Example: A 2D problem with material properties as extra coordinates

This example presents an instance of non-intrusive PGD implementation with material properties as extra coordinates, with two material parameters both separated and collected. To ensure the separation, the parameter space is kept Cartesian.

Consider a plane strain problem in a squared spatial domain $[0, 10] \times [0, 10]$, as sketched in [Figure 4.2](#), the left edge and lower edge are fixed horizontally and vertically, respectively, while the upper edge is free, and the right edge is loaded with uniform traction σ . The domain is partitioned into two subdomain composed by materials with the same Poisson's ratio $\nu = 0.3$ but different Young moduli E_1 and E_2 . Provided having used a compatible unit system, we will ignore units in this example.

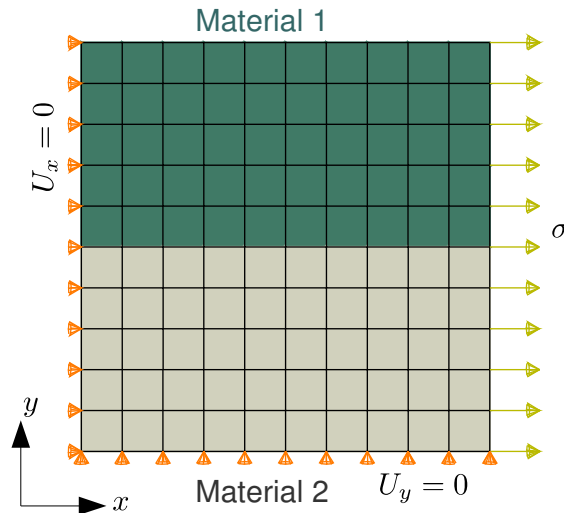


Figure 4.2: Sketch of the 2D problem with material properties as extra coordinates.

For the purpose of validating the new strategy of separating the parameter space, we implement the non-intrusive PGD scheme introduced in [Chapter 3](#) to solve the problem twice with the two parameters separated and collected (unseparated), namely

$$\hat{\mathbf{U}}_{\text{PGD}}^n = \sum_{i=1}^n \mathbf{U}_{\text{sep}}^i \underbrace{\omega_1^i(E_1) \omega_2^i(E_2)}_{\text{separated}} = \sum_{j=1}^n \mathbf{U}_{\text{col}}^j \underbrace{\omega^j(E_1, E_2)}_{\text{collected}}. \quad (4.11)$$

Note that although the spatial domain is identical, the spatial modes resulted from separated approach $\mathbf{U}_{\text{sep}}^i$ and from collected approach $\mathbf{U}_{\text{col}}^j$ are not the same, neither are the functions of parameters.

The ranges of parameters are selected as $E_1 \in [10, 100]$, $E_2 \in [20, 200]$. For the separated approach, the parameter space is $[10, 100] \otimes [20, 200]$, and 100 linear elements are used for the discretisation of both E_1 and E_2 . For the collected approach, the parameter space is $[10, 100] \times [20, 200]$. Three different linear triangular meshes are used to discretise the unseparated parameter space: one with 114 unstructured triangles, one with 100 structured triangles, and to be more comparable with the separated approach, the last one with $100 \times 100 \times 2 = 20000$ triangles. All the meshes are shown in [Figure 4.8](#). We computed 10 PGD modes for both approaches of discretising the parameter space with the same stopping criterion, $\tau_{ol} = 10^{-3}$, for the looping control.

The amplitude of each mode is plotted in [Figure 4.3](#). It demonstrates that the unseparated approach achieves higher rate of convergence than the conventional separated PGD approach. The results imply that the unseparated approach, although presumed to be more expensive than the separated approach, may be not so expensive because less modes are needed to achieve the same accuracy thanks to the higher rate of convergence. It is worth noting that some authors such as [Ammar \[2010\]](#) also compared the separated approach and unseparated approach in the spatial domain. Similarly, the unseparated approach exhibits a higher rate of convergence.

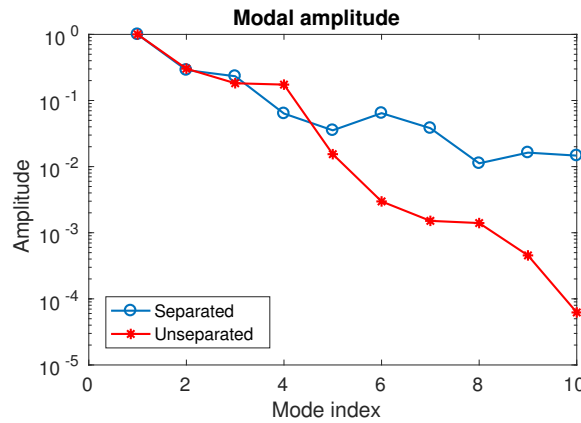


Figure 4.3: Normalised modal amplitude plots for both separated and unseparated approach.

The spatial modes for both approaches are plot in [Figure 4.4](#) and [4.5](#) with nodal displacement magnitudes $U = \sqrt{U_x^2 + U_y^2}$. To visualise the correlation between the two parameters, we plot the tensor product of results from the separated approach in [Figure 4.6](#) and the results from the unseparated approach along with the mesh in

Figure 4.7. It is observed that the first three modes, either spatial or parametric, computed by both separation strategies are very similar.

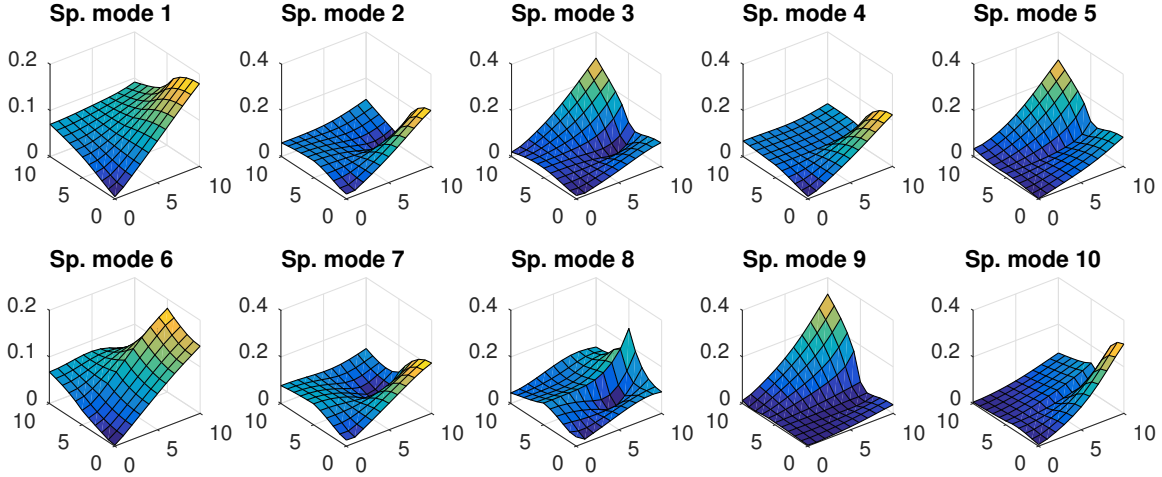


Figure 4.4: Normalised spatial modes $\mathbf{U}_{\text{sep}}^i$ solved from $I_{E_1} \times I_{E_2}$. Bottom plane: spatial domain. Vertical axis: magnitude of nodal displacement.

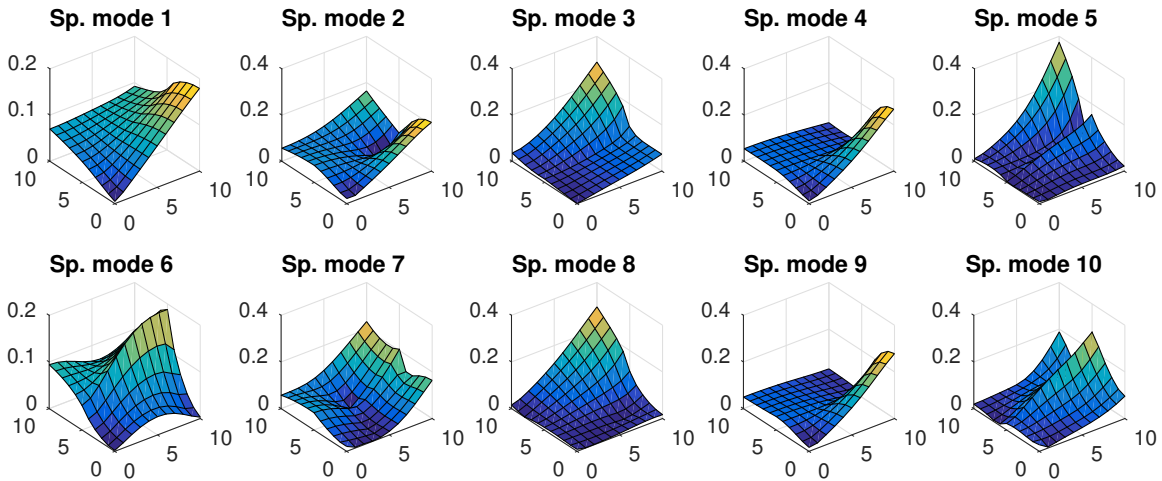


Figure 4.5: Normalised spatial modes $\mathbf{U}_{\text{col}}^i$ solved from Ω_c . Bottom plane: spatial domain. Vertical axis: magnitude of nodal displacement.

To evaluate the error for both approaches, we have randomly selected 5 samples of parameters (E_1, E_2) as listed in Table 4.1, and have computed $\mathbf{U}_{\text{ref}}(E_1, E_2)$ with corresponding ordinary FE models as references. The relative error $\|\hat{\mathbf{U}}_{\text{PGD}} - \mathbf{U}_{\text{ref}}\| / \|\mathbf{U}_{\text{ref}}\|$ for each sample for both approaches are plotted in Figure 4.8. As we have observed, the unseparated approach converges more rapidly. The refinement of mesh for the

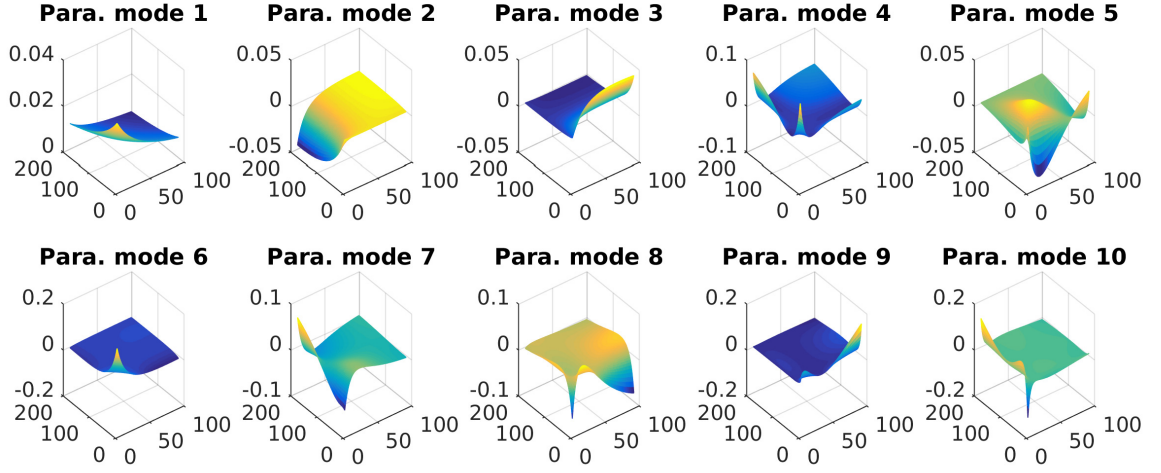


Figure 4.6: Normalised parametric modes $\omega_1^j(E_1) \otimes \omega_2^j(E_2)$ solved from $I_{E_1} \times I_{E_2}$. Bottom plane: parametric domain. Vertical axis: parametric function value.

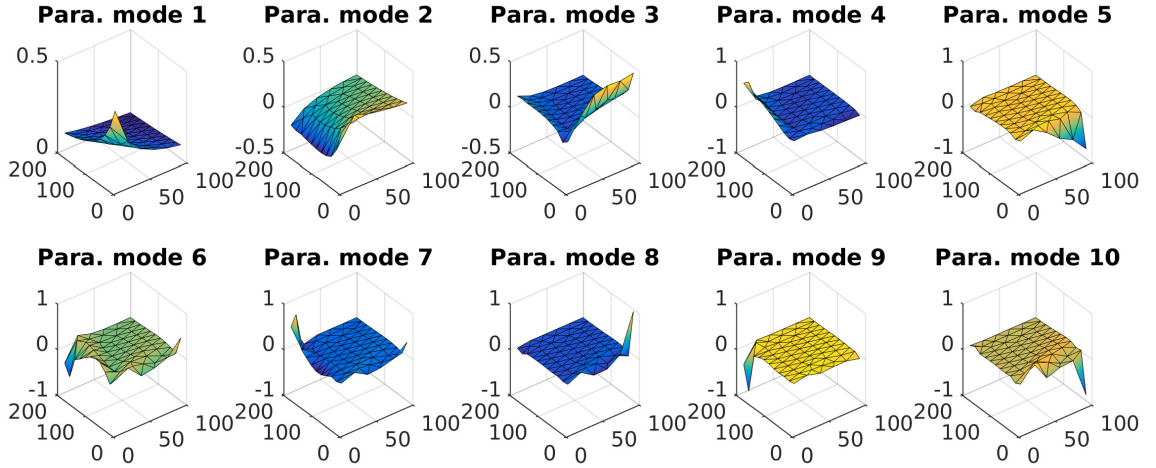


Figure 4.7: Normalised parametric modes $\omega^j(E_1, E_2)$ solved from unseparated Ω_c . Bottom plane: parametric domain. Vertical axis: parametric function value.

unseparated parameter space improves significantly the accuracy of the converged result, but it requires more modes to achieve the convergence. Take the sample $(E_1, E_2) = (76.24, 29.86)$ for example, the relative error converges near 10^{-2} in the coarse meshes with about 100 triangles and the convergence requires 4–5 modes, while it converges to near 10^{-4} in the fine mesh with 20000 triangles and the convergence requires 8 modes. From the comparison of different meshes of parameter space we can

conclude that the mesh density dominates the stagnation accuracy that the PGD solutions converge to, while under the same density the mesh quality has no significant influence on the convergence.

Table 4.1: Randomly selected parameter sets

No.	E_1	E_2
1	93.70	91.04
2	45.91	118.08
3	14.27	143.52
4	40.81	180.85
5	76.24	29.86

4.2 Geometrical constraints

A common situation is when representing the loading locations, the parameter space has to be 2D or 3D. For instance, possible equivalent loading locations on the femur head ranges a partial sphere on the its surface, as illustrated in [Figure 4.9](#).

In this case, the parameter space is generally not separable in the Cartesian fashion. Instead of separating the parameter space into tensor product of 1D parameter spaces, it is better to be kept unseparated and the parametric problem is solved in a 2D/3D space. Provided that the FE model is readily built, it is possible to extract the possible loading locations, that is a set of admissible nodes $\mathbf{s} \in \bar{\Gamma}$, and the parameter space can be reconstructed based on it. To make the discretised parametric domain compatible with the finite element model, the surface is approximated to be composed by the exterior faces of the tetrahedral elements involved. Naturally, the DOFs of the parametric mesh just live on the nodes.

Ideally, since $\dim \bar{\Gamma} = 2$, the parametric domain should be discretised by a 2D triangulation. To build the parametric mesh, we first use the graphical user interface (GUI) provided by Abaqus to extract the related nodes from the spatial mesh, and then use an in-house developed code (or even by hand) to reconstruct the triangular mesh, the procedure is illustrated in [Figure 4.10](#). During the reconstruction process, proper renumbering is required to make the data compatible with the parent finite element model.

However, although practically the loaded nodes can only lie on the exterior surface of the femur head, mathematically it is allowed that we apply some “ghost” loads on

nodes inside the solid. That is to say, the 2D parameter space can be extended to some related tetrahedral elements which consist a subset of the whole 3D spatial domain, namely $\hat{\Omega} \subset \Omega$. With this dimensional extension, the number of parameters have been increased from 2 to 3. The benefit of this extension is the set of tetrahedral elements can be directly extracted from the original finite element model. This extraction procedure, which is in a more non-intrusive fashion, can be performed within the handy Abaqus GUI as shown in [Figure 4.11](#).

The second approach does not require reconstruction of triangular mesh from node extraction, but would make the computation slightly more expensive. Considering the number of selected elements is usually far less than the total element number, the extra cost is acceptable.

A practical example of problem involving this geometric constrained parameter space is presented in [Section 5.2](#).

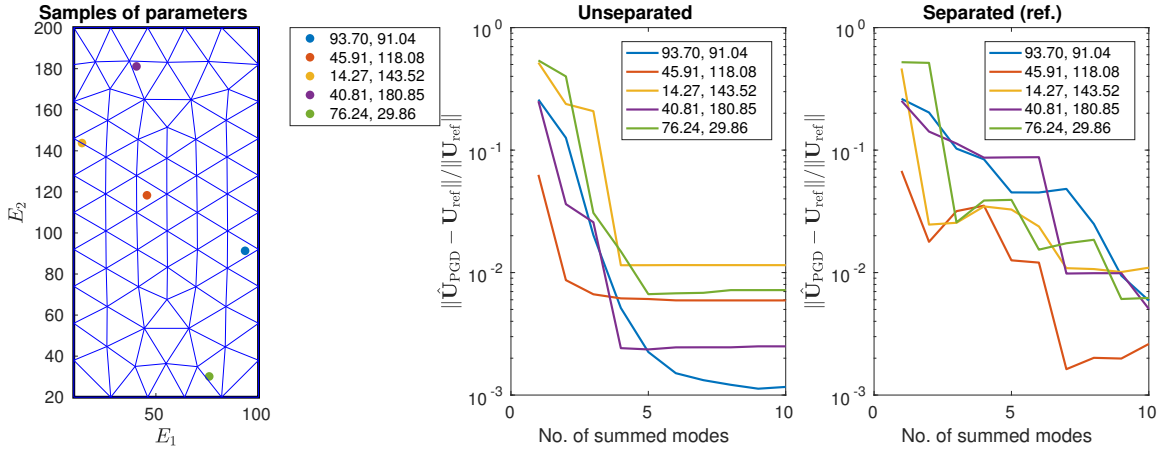
4.3 Summary

For problems involving a parameter space that subjected to some constraints, a separation strategy is proposed in this chapter. The strategy collects the most correlated parameters and keep them in a 2D/3D space which is separated from other parameters. This strategy enables using PGD to solve problems with inseparable parameter space.

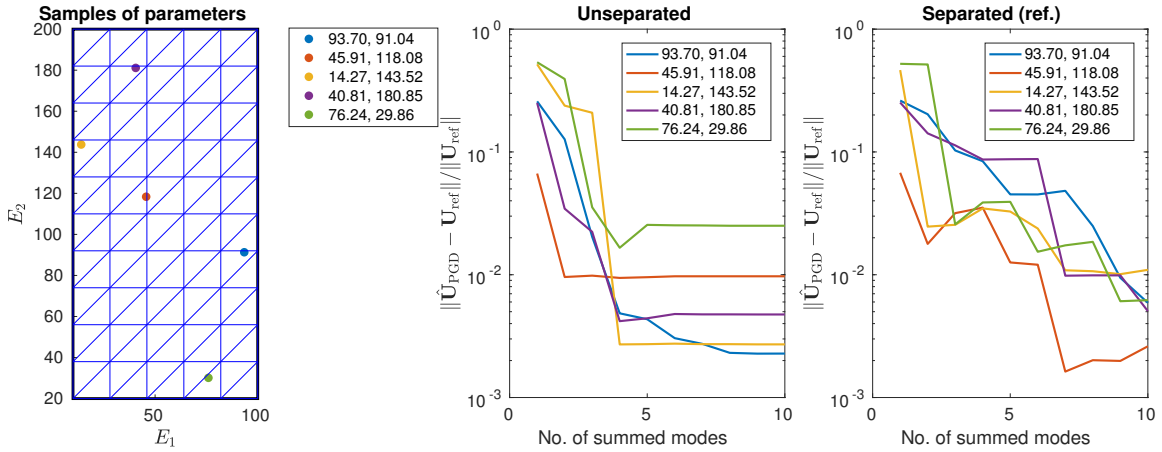
The proposed separation strategy could be more expensive, however, the higher dimensional interpolation improves the convergence rate of the parametrised problem.

In addition, as an implementation in the non-intrusive fashion, an extraction method is introduced using Abaqus GUI to construct the discretised parameter space directly from FE model.

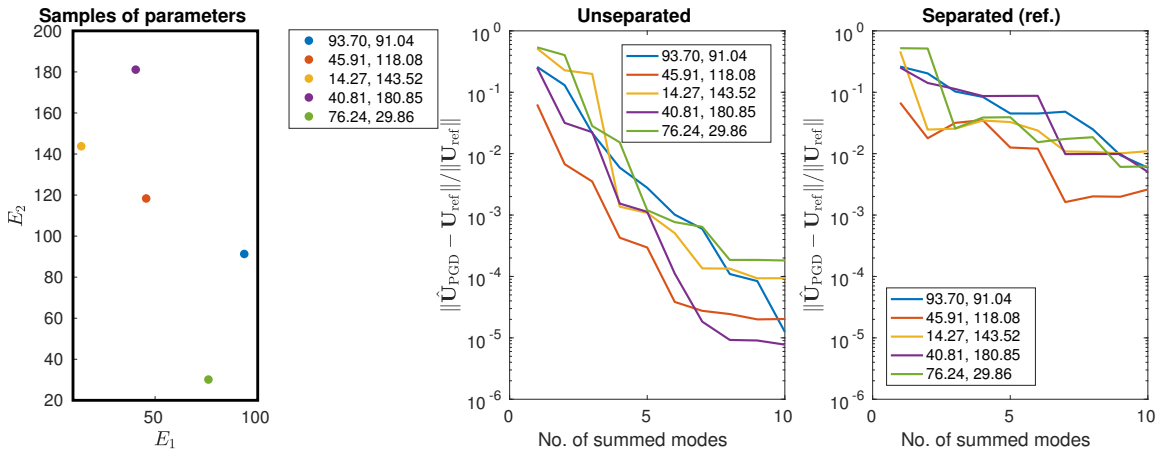
4. PGD FOR CONSTRAINED PARAMETER SPACE



(a) Unstructured coarse mesh with 114 triangles.



(b) Structured coarse mesh with 100 triangles.



(c) Structured fine mesh with 20000 triangles (mesh not shown).

Figure 4.8: Relative error of both separated and unseparated PGD approach for the 2D model problem. For the separated approach, three different meshes for the parameter space are used. For the unseparated approach, the discretisation remains the same.

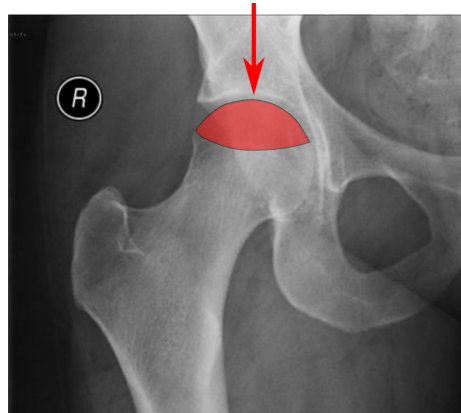


Figure 4.9: Illustration of the possible equivalent loading location on the femur head.

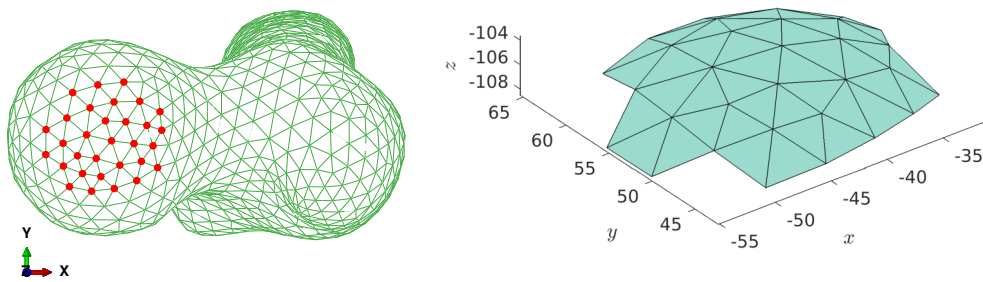


Figure 4.10: Extraction of the involved nodes (left) within Abaqus GUI and parametric mesh (right) reconstructed with in-house code.

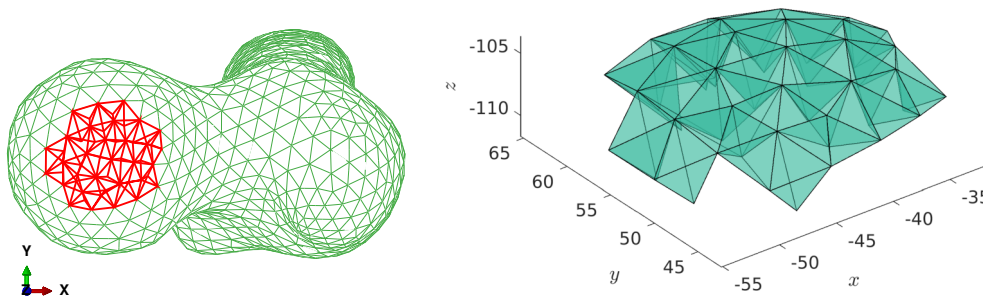


Figure 4.11: Parametric mesh (right) constructed by direct extraction of involved tetrahedral elements (left) using Abaqus GUI.

Chapter 5

Online biomechanical application of PGD vademecum

Literature [Niroomandi et al. \[2013a,b\]](#), [González et al. \[2015\]](#), [Quesada et al. \[2016a\]](#), [Mena et al. \[2015\]](#), [González et al. \[2018\]](#) shows that the PGD framework is particularly suitable for many challenging problems in biomechanics such as real-time simulation and multi-parameter identification. In a comprehensive ageing society, osteoporosis is one of the common diseases for the elders. THA is the typical surgery for osteoporosis happened on the proximal femur. A patient-specific, fast-responding decision making tool could be of great interest for surgeons facing the THA to study the mechanical response of the proximal femur.

The hypothesis of linear elastic modelling of bone tissue at macroscale is commonly accepted for biomechanical analysis under normal loads in regular daily activities [Keaveny et al. \[1994\]](#), [Yosibash et al. \[2006\]](#).

In this chapter, we present a realistic problem in bone mechanics, and apply the non-intrusive PGD scheme proposed in [Chapter 3](#) to solve the problem as another demonstration its feasibility for practical problems. As introduced in [Chapter 2](#), we have previously performed CT scans as well as experimental tests to obtain data from a real sample of femur, and an available FE model is established and validated. For

a macroscopic analysis, the structure material of bone tissue is commonly assumed to be linear elastic, so the numerical scheme well suits this application.

As a typical application of the PGD vademecum at the online phase, a material parameter identification problem is introduced. Because identification problem generally requires multi-query to the response of the model with given input parameters, the extremely fast speed of the online phase using PGD vademecum provides countable accelerations to this problem.

In addition, real-time simulation of the bone structure under loadings at variable locations is enabled by using the PGD vademecum. Proper post-processing of the vademecum further extends the real-time simulation to distributed loads.

5.1 Material parameter identification

Recall the parametrised mechanical problem in [Subsection 4.1.1](#), the PGD solution is computed following [Equation 4.8](#), with FE discretisation of the parameter space as shown in [Figure 5.1](#).

In practice, we confine the parametric domain by proper choices of $(\alpha_{\max}, \beta_{\max})$, and discretise the domain with triangular mesh. In the parameter identification problem, since the values of (α^*, β^*) is unknown while the parametric mesh is flexible, we use a coarser mesh to perform preliminary estimation, and subsequently use a finer mesh to approximate the desired values.

The real parameters $\boldsymbol{\mu}^* = (\alpha^*, \beta^*)$ for the model are unknown *a priori*. As introduced in [Chapter 2](#), *in vitro* experiments in have been performed on the femur, and have obtained data from the 12 strain gauges depicted in [Figure 2.5](#), namely $\boldsymbol{\epsilon}_{\mathbb{P}}^* = [\epsilon_1, \epsilon_2, \dots, \epsilon_{12}]^T$. Taking advantage of the fast online computation with the obtained PGD vademecum [Nadal et al. \[2015a\]](#), we now construct a parameter identification problem to obtain the parameters.

Following standard FE method, the numerical strain is computed as

$$\boldsymbol{\epsilon} = \nabla^s \mathbf{u} = \nabla^s \mathbf{N} \hat{\mathbf{U}} = \mathbf{B} \hat{\mathbf{U}}. \quad (5.1)$$

Regarding to PGD separated representation, when the displacement vademecum $\hat{\mathbf{U}}_{\text{PGD}}(\boldsymbol{\mu})$ is already obtained, the strain vademecum can be computed from

$$\boldsymbol{\epsilon}_{\text{PGD}}(\boldsymbol{\mu}) = \mathbf{B} \hat{\mathbf{U}}_{\text{PGD}}(\boldsymbol{\mu}) = \mathbf{B} \sum_i \mathbf{U}^i \omega^i(\boldsymbol{\mu}) = \sum_i \mathbf{B} \mathbf{U}^i \omega^i(\boldsymbol{\mu}). \quad (5.2)$$

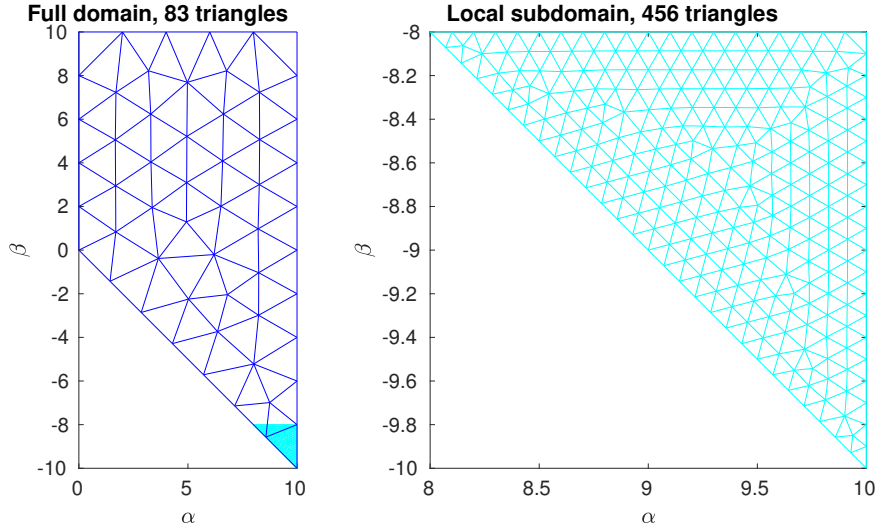


Figure 5.1: Discretisation of the constrained parametric domain. Left: coarse mesh with 83 triangles, $(\alpha_{\max}, \beta_{\max}) = (10, 10)$; right: subdomain mesh with 456 triangles, $(\alpha_{\max}, \beta_{\max}) = (10, -8)$.

Defining a selection operator \mathbb{P} to extract the 12 numerical strain values corresponding to the experiment result, the selected strain can be written as a vector

$$\boldsymbol{\varepsilon}_{\mathbb{P}}(\boldsymbol{\mu}) = \mathbb{P}\boldsymbol{\varepsilon}_{\text{PGD}}(\boldsymbol{\mu}) = \mathbb{P}\mathbf{B} \sum_i \mathbf{U}^i \omega^i(\boldsymbol{\mu}) = \sum_i \mathbb{P}\mathbf{B}\mathbf{U}^i \omega^i(\boldsymbol{\mu}). \quad (5.3)$$

The parameter identification problem reads: given $\boldsymbol{\varepsilon}_{\text{PGD}}(\boldsymbol{\mu})$, find $\boldsymbol{\mu}^*$ such that

$$\boldsymbol{\mu}^* = \arg \min_{\boldsymbol{\mu} \in \Omega_c} \|\boldsymbol{\varepsilon}_{\mathbb{P}}^* - \boldsymbol{\varepsilon}_{\mathbb{P}}(\boldsymbol{\mu})\|. \quad (5.4)$$

To certify the eligibility of the parameter identification problem, benchmark tests have been made with synthetic values for the parameters. We have computed $\boldsymbol{\varepsilon}_{\mathbb{P}}^*$ with synthetic parameters $\boldsymbol{\mu}^* = (3.02, 3680)$, and then we solved the identification problem with $\boldsymbol{\varepsilon}_{\mathbb{P}}^*$ and the previously computed vademecum $\boldsymbol{\varepsilon}_{\mathbb{P}}(\boldsymbol{\mu})$. The obtained result $\boldsymbol{\mu} = (3.0168, 3667.22)$ with relative error $\|\boldsymbol{\mu} - \boldsymbol{\mu}^*\|/\|\boldsymbol{\mu}^*\| = 0.35\%$ shows a good agreement, which proves the identification.

In real case with $\boldsymbol{\varepsilon}_{\mathbb{P}}^*$ from the experiments, results of the parameter identification problem is shown in [Figure 5.2](#). With $\boldsymbol{\varepsilon}_{\mathbb{P}}(\boldsymbol{\mu})$ computed from the coarse parametric mesh, the approximate location of the parameters is identified as $\boldsymbol{\mu}^* = (9.3712, -9.3712)$. Thereafter, the vademecum is recomputed with the fine parametric mesh, and the parameters are identified by rerunning the identification problem, finally obtaining $\boldsymbol{\mu}^* = (9.3589, -9.3589)$.

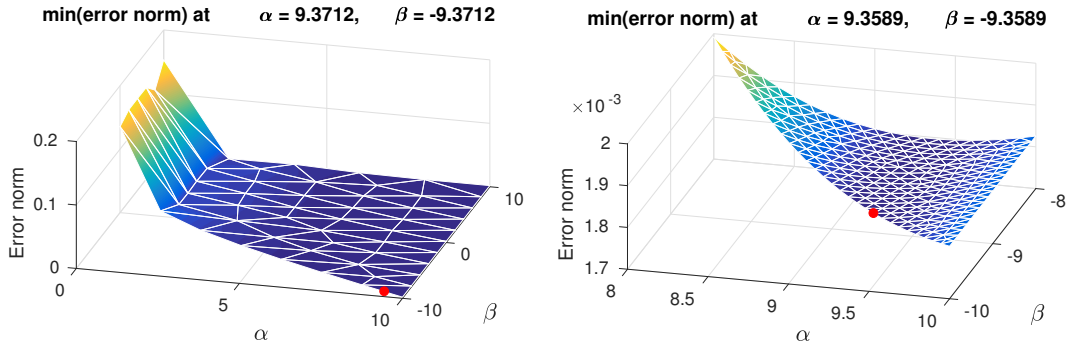


Figure 5.2: Error norm $\|\epsilon_{\mathbb{P}}^* - \epsilon_{\mathbb{P}}(\alpha, \beta)\|$ distribution on the parametric space. Left: preliminary result from coarse mesh; right: secondary result from fine mesh on local subdomain.

Remark 5.1 (Computation of numerical strains). *In practice, there are two options for the implementation of Equation 5.3. The first option is to compute \mathbf{B} within the main code after having obtained $\hat{\mathbf{U}}_{PGD}(\boldsymbol{\mu})$, and then compute $\epsilon_{PGD}(\boldsymbol{\mu})$, and eventually apply \mathbb{P} to obtain $\epsilon_{\mathbb{P}}(\boldsymbol{\mu})$; another option is compute $\mathbb{P}\mathbf{B}\mathbf{U}^i\omega^i(\boldsymbol{\mu})$ by the external solver under the non-intrusive PGD scheme, resulting directly $\epsilon_{\mathbb{P}}(\boldsymbol{\mu})$ by summing all the obtained modes. Particularly in this example, the first option computes \mathbf{B} only once, but due to the fact that the elemental strains are constant, extra post-processing is needed to compute the averaged strain thus this implementation is more intrusive. On the contrary, the second option needs to compute \mathbf{B} for many times during the loops, so that we can use Abaqus to output strains that are already averaged. Therefore, the implementation could be in a more non-intrusive fashion. In our case, we have adopted the second option for its non-intrusiveness, and thus the result we have obtain is the computational vademecum of strains $\epsilon_{PGD}(\boldsymbol{\mu})$.*

Remark 5.2 (On the identified parameters). *Regarding the resultant error norm $\|\epsilon_{\mathbb{P}}^* - \epsilon_{\mathbb{P}}(\alpha, \beta)\|$, its dependency on β is less sensitive in a small scale, this is because ρ^e raises the magnitude of \mathbf{K}_1^e with about 10^3 in Equation 4.6. The resulting parameters occur on the boundary $\alpha + \beta = 0$, this phenomenon indicates that the FE model is not exactly reflecting the mechanics of real femur due to the simplifications we have made during the modelling process. On one hand, in the image-processing procedure detailed in Chapter 2, many empty spaces which should be eliminated during segmentation may have been included in the FE mesh, therefore, the fact that they should have zero elastic modulus drives the identification result to the boundary. On the other hand, despite that the bone material might not be exact linear, the assumption that Young modulus*

is linearly dependent on the intensity of CT image may have strongly increased the discrepancy of the FE model. Even so, the discrepancy between FE model and real sample does not disprove the effectiveness of the PGD vademecum.

5.2 Towards real-time simulation

One of the most significant contributions of PGD computational vademecum is that we may take advantage of it for real-time simulations [Niroomandi et al. \[2013b\]](#). In clinical aspects, this kind of application is of great interest for patient-specific studies.

Using the material parameters $\boldsymbol{\mu}^* = (9.3589, -9.3589)$ obtained from the identification problem in [Section 5.1](#), it is possible to solve another parametrised problem with loading locations as extra coordinates using the non-intrusive PGD scheme of [Subsection 3.3.2](#). As explained in [Section 4.2](#), the loading location is a 2D surface, which can also be extended to 3D solids for the offline solution and restrict the admissible locations on the exterior surface at online phase.

We computed the first 4 dominating modes with both meshes, the modal amplitude curves decrease at almost the same rate as plotted in [Figure 5.3](#). The rate of convergence is nearly the same, which is not surprising because the parametric space is discretised in the same order. The spatial modes represented with displacement in z direction computed with both parametric meshes are shown in [Figure 5.4](#). The parametric modes computed with both meshes are shown in [Figure 5.5](#). It is observed that the modes computed with the two parametric meshes are different both spatial and parametric, and the parametric modes computed with the 2D mesh are more spiny. Even so, the resultant displacements computed with each PGD vademecum is nearly the same. Compared to the standard FE result with a specified loading location, the error norm of the 2D parametric mesh result is 3.61% while that of the 3D parametric mesh result is 3.49%.

Having computed the computational vademecum $\hat{\mathbf{U}}_{\text{PGD}}(\mathbf{s})$, we are able to perform real-time simulation in the online computations on the numerical femur model. For instance, the strain field under different loading locations can be computed with [Equation 5.2](#). It is believed to have great potential clinic applications thanks to the fact that the online computation with the computational vademecum is extremely fast.

For a prescribed distribute load $\mathbf{t}(\mathbf{s})$, with $\mathbf{s} \in \bar{\Gamma}$, referring to [Equation 3.33](#), it

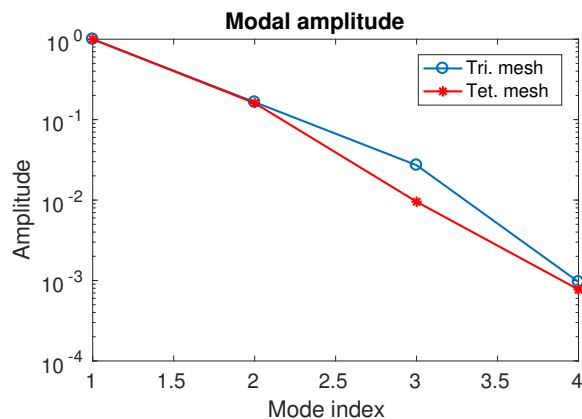


Figure 5.3: Normalized modal amplitude plot for triangular and tetrahedral parametric meshes.

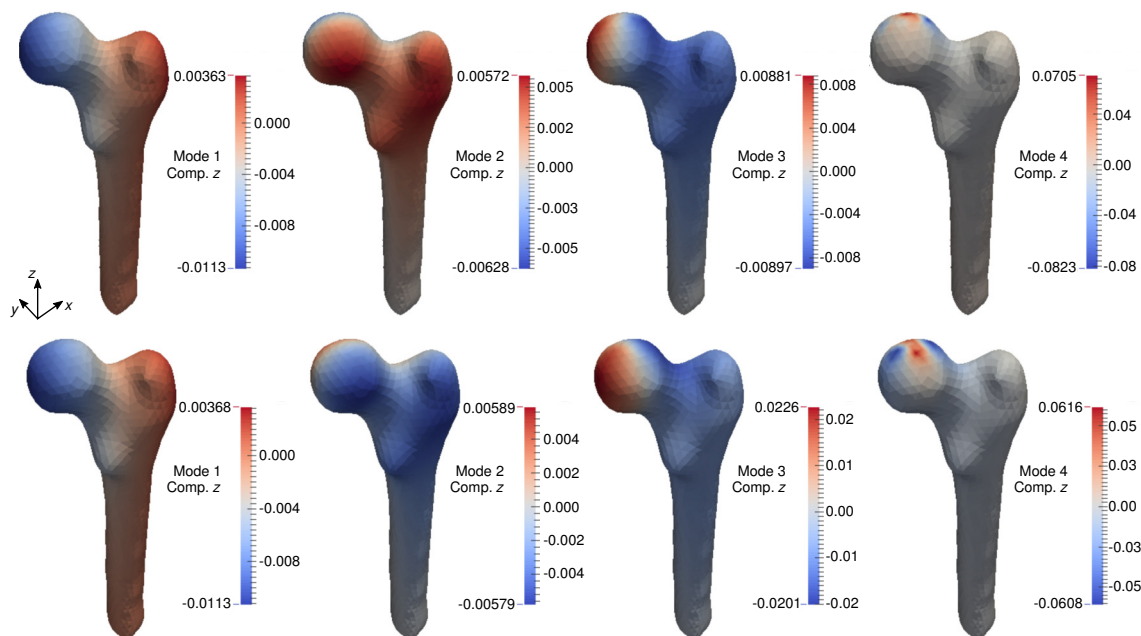


Figure 5.4: The z component of normalized spatial modes computed with triangular mesh (upper row) and tetrahedral mesh (lower row).

is possible to write the global load vector \mathbf{F} as a linear superposition of equivalent nodal forces

$$\mathbf{F} = \int_{\hat{\Gamma}} \mathbf{N}^T \mathbf{t}(\mathbf{s}) d\Gamma = \sum_i F_i \mathbf{f}(\mathbf{s}_i), \quad (5.5)$$

where \mathbf{f} is a single-point load applied at \mathbf{s}_i and F_i denotes its amplitude. With obtained (α^*, β^*) , the displacement DOF vector \mathbf{U}_{dist} under the given distributed

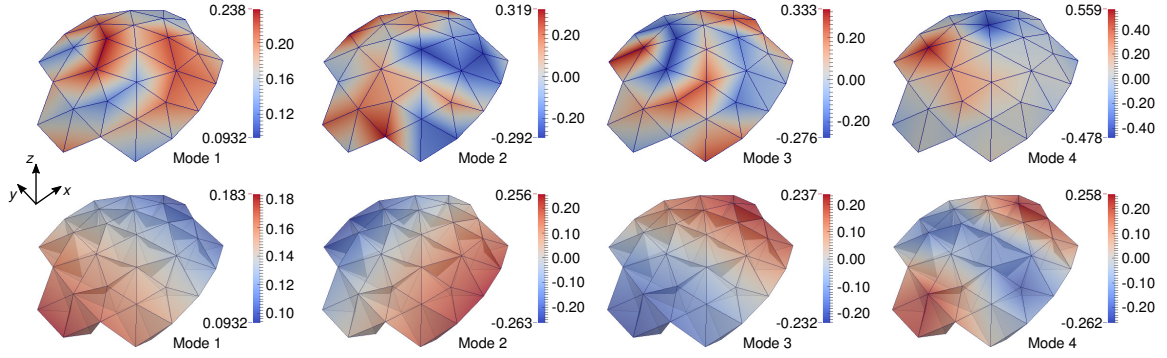


Figure 5.5: Parametric modes $\omega^j(\boldsymbol{\mu})$ computed with triangular mesh (upper row) and tetrahedral mesh (lower row).

load is computed as

$$\mathbf{U}_{\text{dist}} = (\alpha^* \mathbf{K}_1 + \beta^* \mathbf{K}_2)^{-1} \mathbf{F} = (\alpha^* \mathbf{K}_1 + \beta^* \mathbf{K}_2)^{-1} \sum_i F_i \mathbf{f}(\mathbf{s}_i). \quad (5.6)$$

Note that

$$(\alpha^* \mathbf{K}_1 + \beta^* \mathbf{K}_2)^{-1} \mathbf{f}(\mathbf{s}_i) = \hat{\mathbf{U}}_{\text{PGD}}(\mathbf{s}_i), \quad (5.7)$$

the displacement DOF vector is eventually obtained by

$$\mathbf{U}_{\text{dist}} = \sum_i F_i \hat{\mathbf{U}}_{\text{PGD}}(\mathbf{s}_i). \quad (5.8)$$

5.3 Summary

For biomechanical structures, the data of geometric shape as well as material properties are usually not directly available *in vivo*. Those data are generally obtained by processing medical image sets from CT scan, MRI, radiography, ultrasonography, etc. This chapter has demonstrated that using the computational vademecum generated from non-intrusive PGD scheme, it is possible to identify the material constitutive parameters and perform real-time simulation during the online phase.

Chapter 6

NURBS shape function discretisation of parameter spaces

This chapter is dedicated to the introduction of a practical application of the non-intrusive PGD scheme stated in [Chapter 3](#) with the separation strategy stated in [Chapter 4](#). Although the discussion is still based on linear elasticity, a generalisation is made to extend the application from isotropic materials to orthotropic materials. It is worth noting that most materials studied in biomechanics show an orthotropic behaviour, because of their fibre reinforced structures.

It is well known that for many inverse problems, such as identification of material constitutive parameters, computation of the derivatives with respect to the parameters are necessary. However, in the conventional PGD framework, low-order discretisation methods, such as finite difference, collocation with trapezoidal integration as well as linear FE, are commonly used for the parameter space, which could cause problems not only during the offline phase, but the online simulation phase. To improve the performance of the PGD framework, it is interesting to introduce IGA concepts to the discretisation of the parameter space. Using NURBS shape functions, typically the B-splines, it is not only easy to obtain high-order discretisation of the parameter space, but the smoothness of the parametric modes would be im-

proved by the high continuity. In addition, computational efficiency of the high-order discretisation could benefit from the k -refinement in an IGA fashion.

This approach is referred as NURBS Enhanced PGD (NE-PGD) framework in the following context, in order to distinguish from the conventional FE-PGD framework. To the author's knowledge, although the extension from FE-PGD to NE-PGD is a quite natural idea, it is still innovative and rarely seen in the literature. On the other hand, the combination of IGA and RB is under active investigation in the model order reduction community [Salmoiraghi et al. \[2016\]](#), [Devaud and Rozza \[2017\]](#).

6.1 Constitutive parameters of 2D orthotropic materials

Consider a 2D orthotropic material, such as a composite plate, an equivalent unidirectional lamina can be used to represent its in-plane mechanical property. For any unidirectional lamina, there are four independent parameters of the in-plane material property: E_1 , E_2 , ν_{12} , G_{12} , where the subscripts denotes an orthogonal coordinate.

The ranges for the parameters are assumed to be

$$\begin{aligned} E_1 &\in [10, 30] \text{ GPa}, & E_2 &\in [10, 30] \text{ GPa}, \\ \nu_{12} &\in (0, 0.5), & G_{12} &\in [0.5, 10] \text{ GPa}. \end{aligned} \quad (6.1)$$

The stress-strain relationship is

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix}, \quad (6.2)$$

where the elastic constants are defined as

$$\begin{aligned} Q_{11} &= \frac{E_1}{1 - \nu_{21}\nu_{12}}, & Q_{12} &= \frac{\nu_{12}E_2}{1 - \nu_{21}\nu_{12}}, \\ Q_{22} &= \frac{E_2}{1 - \nu_{21}\nu_{12}}, & Q_{66} &= G_{12}, \end{aligned} \quad (6.3)$$

and the relationship

$$\frac{E_1}{E_2} = \frac{\nu_{12}}{\nu_{21}}. \quad (6.4)$$

To simplify the relationship between the parameters, we introduce a change of variables with the definition of

$$q_1 := \frac{E_1}{1 - \nu_{12}^2 E_2 / E_1}, \quad q_2 := \frac{E_2}{E_1}, \quad q_3 := \nu_{12}, \quad q_4 := G_{12}, \quad (6.5)$$

and it is straightforward to obtain

$$Q_{11} = q_1, \quad Q_{22} = q_1 q_2, \quad Q_{12} = q_1 q_2 q_3, \quad Q_{66} = q_4, \quad (6.6)$$

Imposing the following restrictions

$$\begin{aligned} E_1 &> E_2 > 0, \\ 0 &< \nu_{12} < 0.5, \end{aligned} \quad (6.7)$$

we can compute the admissible ranges of the parameters:

$$\begin{aligned} q_1 &\in (10, 40) \text{ GPa}, \quad q_2 \in [1/3, 1], \\ q_3 &\in (0, 0.5), \quad q_4 \in [0.5, 10] \text{ GPa}. \end{aligned} \quad (6.8)$$

After the change of variables, it is straightforward to verify that the resultant parametric space $\Omega_{\mathbf{q}} \subset \text{span}\{q_1, q_2, q_3, q_4\}$ is still Cartesian under the constraints. It can also be seen from [Equation 6.5](#) that q_1, q_2, q_3 are more correlated while q_4 is independent. Therefore, considering the strategy proposed in [Chapter 4](#), it is reasonable that we separate the first three parameters into a 3D space $\Omega_{\tilde{\mathbf{q}}} = \text{span}\{q_1, q_2, q_3\}$, and the last parameter a 1D space I_{q_4} . For convenience, we introduce the following notations

$$\begin{aligned} \tilde{\mathbf{q}} &:= (q_1, q_2, q_3), \\ \mathbf{q} &:= (q_1, q_2, q_3, q_4) = (\tilde{\mathbf{q}}, q_4). \end{aligned} \quad (6.9)$$

6.2 Decomposition of element stiffness matrix

With definition of the element elasticity matrix

$$\mathbf{D} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix}, \quad (6.10)$$

considering [Equation 6.6](#), we can decompose the elastic matrix \mathbf{D} as

$$\mathbf{D} = \begin{bmatrix} q_1 & q_1 q_2 q_3 & 0 \\ q_1 q_2 q_3 & q_1 q_2 & 0 \\ 0 & 0 & q_4 \end{bmatrix} = q_1 \mathbf{D}_1 + q_1 q_2 \mathbf{D}_2 + q_1 q_2 q_3 \mathbf{D}_3 + q_4 \mathbf{D}_4, \quad (6.11)$$

where

$$\begin{aligned}
 \mathbf{D}_1 &:= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{D}_2 &:= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
 \mathbf{D}_3 &:= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{D}_4 &:= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{6.12}$$

For the mechanical problem, we keep using FE for the spatial discretisation. The element stiffness matrix can be obtained by

$$\mathbf{K} = \int_{\Omega_e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV = q_1 \mathbf{K}_1 + q_1 q_2 \mathbf{K}_2 + q_1 q_2 q_3 \mathbf{K}_3 + q_4 \mathbf{K}_4, \tag{6.13}$$

where $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n]$ is the strain-displacement matrix for n -node plane stress elements with shape functions N_i

$$\mathbf{B}_i = \begin{bmatrix} \partial N_i / \partial x & 0 \\ 0 & \partial N_i / \partial y \\ \partial N_i / \partial y & \partial N_i / \partial x \end{bmatrix}, \quad i = 1, 2, \dots, n, \tag{6.14}$$

and

$$\begin{aligned}
 \mathbf{K}_1 &= \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_1 \mathbf{B} dV, \\
 \mathbf{K}_2 &= \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_2 \mathbf{B} dV, \\
 \mathbf{K}_3 &= \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_3 \mathbf{B} dV, \\
 \mathbf{K}_4 &= \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_4 \mathbf{B} dV.
 \end{aligned} \tag{6.15}$$

In addition, it is possible to compute the partial derivatives of \mathbf{K} with respect to the parameters as

$$\begin{aligned}
 \frac{\partial \mathbf{K}}{\partial q_1} &= \mathbf{K}_1 + q_2 \mathbf{K}_2 + q_2 q_3 \mathbf{K}_3, \\
 \frac{\partial \mathbf{K}}{\partial q_2} &= q_1 \mathbf{K}_2 + q_1 q_3 \mathbf{K}_3, \\
 \frac{\partial \mathbf{K}}{\partial q_3} &= q_1 q_2 \mathbf{K}_3, \\
 \frac{\partial \mathbf{K}}{\partial q_4} &= \mathbf{K}_4.
 \end{aligned} \tag{6.16}$$

From [Equation 6.16](#) it can be observed that the derivatives with respect to q_1, q_2, q_3 have quadratic dependency on the parameters, while that with respect to q_4 shows only linearly dependency.

6.3 PGD separated representation

Assume the displacement field is discretised with standard FEM

$$\mathbf{u}(\mathbf{x}; q_1, q_2, q_3, q_4) \approx \mathbf{N}(\mathbf{x})\mathbf{U}(q_1, q_2, q_3, q_4), \quad (6.17)$$

where $\mathbf{N}(\mathbf{x})$ is the standard FE shape function matrix, \mathbf{U} is the global displacement DOF vector.

The parametric problem is to compute the vademecum of global displacement DOF vector

$$\mathbf{U}(q_1, q_2, q_3, q_4) \approx \hat{\mathbf{U}}_{\text{PGD}}^M(\mathbf{q}) = \sum_{i=1}^M \boldsymbol{\chi}_i \lambda_i(\tilde{\mathbf{q}}) \omega_i(q_4), \quad (6.18)$$

where $\boldsymbol{\chi}_i$ denotes the spatial mode, and $\lambda_i : \Omega_{\tilde{\mathbf{q}}} \rightarrow \mathbb{R}$ and $\omega_i : I_{q_4} \rightarrow \mathbb{R}$ are the 3D and 1D parametric modes respectively.

Now we are able to write the semi-discretised alternative direction iteration scheme following [Equation 3.17](#) and [3.18](#):

1. **Mechanical problem:** assuming the first $M - 1$ modes, $\lambda_M(\tilde{\mathbf{q}})$ and $\omega_M(q_4)$ are known, solve the M th spatial mode $\boldsymbol{\chi}_M$ with

$$\int_{\Omega_{\tilde{\mathbf{q}}}} \int_{I_{q_4}} \mathbf{K} \boldsymbol{\chi}_M \lambda_M^2 \omega_M^2 d\tilde{\mathbf{q}} dq_4 = \int_{\Omega_{\tilde{\mathbf{q}}}} \int_{I_{q_4}} \left(\mathbf{F} - \sum_{j=1}^{M-1} \mathbf{K} \boldsymbol{\chi}_j \lambda_j \omega_j \right) \lambda_M \omega_M d\tilde{\mathbf{q}} dq_4. \quad (6.19)$$

2. **3D parametric problem:** assuming the first $M - 1$ modes, $\boldsymbol{\chi}_M$ and $\omega_M(q_4)$ are known, solve $\lambda_M(\tilde{\mathbf{q}})$ with

$$\lambda_M \int_{I_{q_4}} \boldsymbol{\chi}_M^T \mathbf{K} \boldsymbol{\chi}_M \omega_M^2 dq_4 = \int_{I_{q_4}} \boldsymbol{\chi}_M^T \left(\mathbf{F} - \sum_{j=1}^{M-1} \mathbf{K} \boldsymbol{\chi}_j \lambda_j \omega_j \right) \omega_M dq_4. \quad (6.20)$$

3. **1D parametric problem:** assuming the first $M - 1$ modes, $\boldsymbol{\chi}_M$ and $\lambda_M(\tilde{\mathbf{q}})$ are known, solve $\omega_M(q_4)$ with

$$\omega_M \int_{\Omega_{\tilde{\mathbf{q}}}} \boldsymbol{\chi}_M^T \mathbf{K} \boldsymbol{\chi}_M \lambda_M^2 d\tilde{\mathbf{q}} = \int_{\Omega_{\tilde{\mathbf{q}}}} \boldsymbol{\chi}_M^T \left(\mathbf{F} - \sum_{j=1}^{M-1} \mathbf{K} \boldsymbol{\chi}_j \lambda_j \omega_j \right) \lambda_M d\tilde{\mathbf{q}}. \quad (6.21)$$

6.4 Discretisation with B-Spline interpolation

As the core idea of this example, we use a NURBS solid to represent the 3D parametric space $\Omega_{\tilde{\mathbf{q}}}$, and a NURBS curve to represent the 1D parametric space I_{q_4} . To be

simplified, we use B-splines, which is a typical NURBS with all weights equal to one, for practical implementations.

6.4.1 B-spline discretisation in the IGA fashion

Now we introduce the necessary notations following the IGA conventions in the literature [Piegl and Tiller \[1997\]](#), [Hughes et al. \[2005\]](#). It is known B-splines are a specific type of NURBS with all weights equal to one. The B-spline curves are constructed by linear combinations of B-spline basis functions. The coefficients of the basis functions are referred to as control points. The unknown variables are now computed on the control points instead of nodes, thus they are called control variables and are equivalent to DOFs. Let the knot vector be $\boldsymbol{\xi} = (\xi_1, \xi_2, \dots, \xi_{n+p+1})$, where ξ_i is the i th knot, i is the knot index, $i = 1, 2, \dots, n + p + 1$, p is the polynomial order, and n is the number of basis functions used to construct the B-spline curve.

The B-spline basis functions $N_{i,p}(\xi)$ are defined using Cox–de Boor recursion formula [de Boor \[1972, 1977\]](#). Starting with piecewise constants ($p = 0$)

$$N_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \quad (6.22)$$

For $p \geq 1$, they are defined by

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \quad (6.23)$$

The first derivative of the shape functions are:

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \quad (6.24)$$

As a demonstration, a typical set of B-spline shape functions and their corresponding derivatives are plotted in [Figure 6.1](#) with 5 knot spans (which in analogy to elements in FE point of view) in different orders. In IGA, the refinement of shape functions is enriched beyond h -refinement and p -refinement inherited from FEM. Due to the non-commutability of the two legacy types of refinements, k -refinement is introduced by elevating the order prior to inserting the knots. The k -refinement is important and a superior approach to high-precision analysis than p -refinement. It not only increases the continuity of the piecewise polynomials, but also keeps homogeneous structure within the refined patch with limited proliferation of control variables.

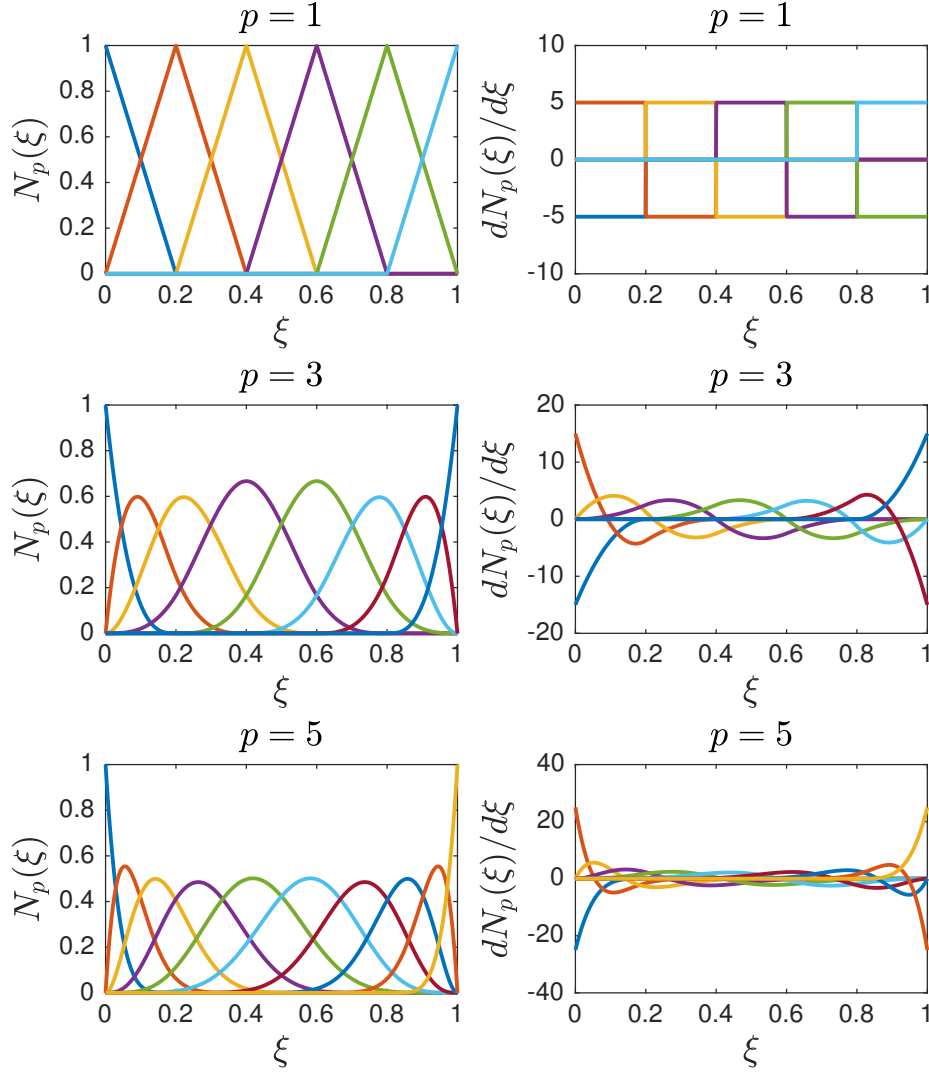


Figure 6.1: Typical set of B-spline shape functions and the corresponding derivatives.

Let \mathbf{B}_i denote the control points, a piecewise-polynomial B-spline curve $\mathbf{C}(\xi)$ is given by

$$\mathbf{C}(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i. \quad (6.25)$$

For the 1D parameter space, it is known to be a straight line since it is Cartesian. Denoting the knot vector as $\tilde{\boldsymbol{\xi}} = (\tilde{\xi}_1, \tilde{\xi}_2, \dots, \tilde{\xi}_{n+p+1})$, it can be represented with the following B-spline:

$$q_4(\tilde{\boldsymbol{\xi}}) = \sum_{i=1}^{\tilde{n}} \tilde{N}_{i,p}(\tilde{\boldsymbol{\xi}}) \tilde{q}_i, \quad (6.26)$$

where \tilde{q}_i are the control points.

For the 3D parameter space, the representative B-spline solid is defined in a tensor product way. Given a control lattice $\mathbf{B}_{i,j,k}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, $k = 1, 2, \dots, l$, polynomial orders p, q, r , and knot vectors $\boldsymbol{\xi} = (\xi_1, \xi_2, \dots, \xi_{n+p+1})$, $\boldsymbol{\eta} = (\eta_1, \eta_2, \dots, \eta_{m+q+1})$, $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_{l+r+1})$, the B-spline solid is defined by

$$\tilde{\mathbf{q}}(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) \mathbf{B}_{i,j,k}, \quad (6.27)$$

Following the isogeometric fashion, the parametric modes $\lambda(\tilde{\mathbf{q}}), \omega(q_4)$ can be approximated with shape functions from values computed on control points.

$$\begin{aligned} \lambda(\tilde{\mathbf{q}}) &= \lambda \circ \tilde{\mathbf{q}}(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) \hat{\lambda}_{i,j,k} := \mathbf{R}^T \boldsymbol{\lambda}, \\ \omega(q_4) &= \omega \circ q_4(\tilde{\xi}) = \sum_{i=1}^{\tilde{n}} \tilde{N}_{i,p}(\tilde{\xi}) \hat{\omega}_i := \mathbf{P}^T \boldsymbol{\omega}, \end{aligned} \quad (6.28)$$

where $\hat{\lambda}_{i,j,k}$ denotes the DOF computed on $\mathbf{B}_{i,j,k}$, and $\hat{\omega}_i$ the DOF computed on \tilde{q}_i . The vector form of the DOFs are denoted by $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ respectively.

Let $\mathbf{ncp} = n \cdot m \cdot l$ and \mathbf{mcp} denote the numbers of control points for the discretised 3D and 1D parameter space respectively, the shape function vectors in [Equation 6.28](#) are defined as

$$\begin{aligned} [\mathbf{R}]_A &= N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta), & A &= 1, 2, \dots, \mathbf{ncp}, \\ [\mathbf{P}]_B &= \tilde{N}_{i,p}(\tilde{\xi}), & B &= 1, 2, \dots, \mathbf{mcp}. \end{aligned} \quad (6.29)$$

For the purpose of simplifying the notations, we define the following mass-like matrices

$$\begin{aligned} \mathbf{H}_1 &:= \int_{\Omega_{\tilde{\mathbf{q}}}} q_1 \mathbf{R} \mathbf{R}^T d\tilde{\mathbf{q}}, & \mathbf{H}_2 &:= \int_{\Omega_{\tilde{\mathbf{q}}}} q_1 q_2 \mathbf{R} \mathbf{R}^T d\tilde{\mathbf{q}}, \\ \mathbf{H}_3 &:= \int_{\Omega_{\tilde{\mathbf{q}}}} q_1 q_2 q_3 \mathbf{R} \mathbf{R}^T d\tilde{\mathbf{q}}, & \mathbf{S} &:= \int_{\Omega_{\tilde{\mathbf{q}}}} \mathbf{R} \mathbf{R}^T d\tilde{\mathbf{q}}, \\ \mathbf{H}_4 &:= \int_{I_{q_4}} q_4 \mathbf{P} \mathbf{P}^T dq_4, & \mathbf{M} &:= \int_{I_{q_4}} \mathbf{P} \mathbf{P}^T dq_4, \end{aligned} \quad (6.30)$$

and the following force-like vectors

$$\mathbf{Q} := \int_{\Omega_{\tilde{\mathbf{q}}}} \mathbf{R}^T d\tilde{\mathbf{q}}, \quad \mathbf{Z} := \int_{I_{q_4}} \mathbf{P}^T dq_4. \quad (6.31)$$

Standard Gaussian quadrature is used to compute the mass-like matrices and force-like vectors. For the 3D parameter space, the total number of quadrature points is denoted by \mathbf{ngp} , whilst for the 1D parameter space, it is denoted by \mathbf{mcp} . Note

that the quadrature is performed over the index space, a coordinate transformation mapping is needed. For the 3D space, it is necessary to define the Jacobian for the mapping between the index space and the parameter space

$$\mathbf{J}(\xi, \eta, \zeta) = \frac{\partial(q_1, q_2, q_3)}{\partial(\xi, \eta, \zeta)}. \quad (6.32)$$

Let the control lattice be $\mathbf{B}_a = (B_{a,1}, B_{a,2}, B_{a,3})$, for the 3D space we have

$$q_i(\xi, \eta, \zeta) = \sum_{a=1}^{\text{ngp}} R_a(\xi, \eta, \zeta) B_{a,i}, \quad i = 1, 2, 3, \quad (6.33)$$

and for the 1D space we have

$$q_4(\tilde{\xi}) = \sum_{a=1}^{\text{ngp}} P_a(\tilde{\xi}) \tilde{q}_a, \quad \frac{dq_4(\tilde{\xi})}{d\tilde{\xi}} = \sum_{a=1}^{\text{ngp}} \frac{dP_a(\tilde{\xi})}{d\tilde{\xi}} \tilde{q}_a. \quad (6.34)$$

The mass-like matrices and force-like vectors are computed as

$$\begin{aligned} [\mathbf{H}_1]_{AB} &\approx \sum_{j=1}^{\text{ngp}} w_j \det \mathbf{J} q_1 R_A(\xi_j, \eta_j, \zeta_j) R_B(\xi_j, \eta_j, \zeta_j), \\ [\mathbf{H}_2]_{AB} &\approx \sum_{j=1}^{\text{ngp}} w_j \det \mathbf{J} q_1 q_2 R_A(\xi_j, \eta_j, \zeta_j) R_B(\xi_j, \eta_j, \zeta_j), \\ [\mathbf{H}_3]_{AB} &\approx \sum_{j=1}^{\text{ngp}} w_j \det \mathbf{J} q_1 q_2 q_3 R_A(\xi_j, \eta_j, \zeta_j) R_B(\xi_j, \eta_j, \zeta_j), \\ [\mathbf{S}]_{AB} &\approx \sum_{j=1}^{\text{ngp}} w_j \det \mathbf{J} R_A(\xi_j, \eta_j, \zeta_j) R_B(\xi_j, \eta_j, \zeta_j), \\ [\mathbf{H}_4]_{AB} &\approx \sum_{j=1}^{\text{ngp}} w_j \sum_{a=1}^{\text{ngp}} \frac{dP_a(\tilde{\xi}_j)}{d\tilde{\xi}} \tilde{q}_a \sum_b P_b(\tilde{\xi}_j) B_b P_A(\tilde{\xi}_j) P_B(\tilde{\xi}_j), \\ [\mathbf{M}]_{AB} &\approx \sum_{j=1}^{\text{ngp}} w_j \sum_{a=1}^{\text{ngp}} \frac{dP_a(\tilde{\xi}_j)}{d\tilde{\xi}} \tilde{q}_a P_A(\tilde{\xi}_j) P_B(\tilde{\xi}_j), \end{aligned} \quad (6.35)$$

and

$$\begin{aligned} [\mathbf{Q}]_A &\approx \sum_{j=1}^{\text{ngp}} w_j \det \mathbf{J} R_A(\xi_j, \eta_j, \zeta_j), \\ [\mathbf{Z}]_A &\approx \sum_{j=1}^{\text{ngp}} w_j \sum_a \frac{dP_a(\tilde{\xi}_j)}{d\tilde{\xi}} \tilde{q}_a P_A(\tilde{\xi}_j). \end{aligned} \quad (6.36)$$

where w_j are the weights for the Gaussian quadrature. On-going studies on novel efficient quadrature schemes for integrating the NURBS shape functions can be found in [Hughes et al. \[2010\]](#).

6.4.2 Fully discretised PGD formulation

Now we may obtain the fully discretised formulation for PGD alternative direction iteration.

1. For the mechanical problem:

$$\begin{aligned}
 \text{LHS} &= \left[\sum_{i=1}^3 (\boldsymbol{\lambda}_M^T \mathbf{H}_i \boldsymbol{\lambda}_M) (\boldsymbol{\omega}_M^T \mathbf{M} \boldsymbol{\omega}_M) \mathbf{K}_i + (\boldsymbol{\lambda}_M^T \mathbf{S} \boldsymbol{\lambda}_M) (\boldsymbol{\omega}_M^T \mathbf{H}_4 \boldsymbol{\omega}_M) \mathbf{K}_4 \right] \boldsymbol{\chi}_M, \\
 \text{RHS} &= (\mathbf{Q}^T \boldsymbol{\lambda}_M) (\mathbf{Z}^T \boldsymbol{\omega}_M) \mathbf{F} \\
 &\quad - \sum_{j=1}^{M-1} \left[\sum_{i=1}^3 (\boldsymbol{\lambda}_M^T \mathbf{H}_i \boldsymbol{\lambda}_j) (\boldsymbol{\omega}_M^T \mathbf{M} \boldsymbol{\omega}_j) \mathbf{K}_i + (\boldsymbol{\lambda}_M^T \mathbf{S} \boldsymbol{\lambda}_j) (\boldsymbol{\omega}_M^T \mathbf{H}_4 \boldsymbol{\omega}_j) \mathbf{K}_4 \right] \boldsymbol{\chi}_j.
 \end{aligned} \tag{6.37}$$

2. For the 3D parametric problem:

$$\begin{aligned}
 \text{LHS} &= \left[\sum_{i=1}^3 (\boldsymbol{\chi}_M^T \mathbf{K}_i \boldsymbol{\chi}_M) (\boldsymbol{\omega}_M^T \mathbf{M} \boldsymbol{\omega}_M) \mathbf{H}_i + (\boldsymbol{\chi}_M^T \mathbf{K}_4 \boldsymbol{\chi}_M) (\boldsymbol{\omega}_M^T \mathbf{H}_4 \boldsymbol{\omega}_M) \mathbf{S} \right] \boldsymbol{\lambda}_M, \\
 \text{RHS} &= (\boldsymbol{\chi}_M^T \mathbf{F}) (\mathbf{Z}^T \boldsymbol{\omega}_M) \mathbf{Q} \\
 &\quad - \sum_{j=1}^{M-1} \left[\sum_{i=1}^3 (\boldsymbol{\chi}_M^T \mathbf{K}_i \boldsymbol{\chi}_j) (\boldsymbol{\omega}_M^T \mathbf{M} \boldsymbol{\omega}_j) \mathbf{H}_i + (\boldsymbol{\chi}_M^T \mathbf{K}_4 \boldsymbol{\chi}_j) (\boldsymbol{\omega}_M^T \mathbf{H}_4 \boldsymbol{\omega}_j) \mathbf{S} \right] \boldsymbol{\lambda}_j.
 \end{aligned} \tag{6.38}$$

3. For the 1D parametric problem:

$$\begin{aligned}
 \text{LHS} &= \left[\sum_{i=1}^3 (\boldsymbol{\chi}_M^T \mathbf{K}_i \boldsymbol{\chi}_M) (\boldsymbol{\lambda}_M^T \mathbf{H}_i \boldsymbol{\lambda}_M) \mathbf{M} + (\boldsymbol{\chi}_M^T \mathbf{K}_4 \boldsymbol{\chi}_M) (\boldsymbol{\lambda}_M^T \mathbf{S} \boldsymbol{\lambda}_M) \mathbf{H}_4 \right] \boldsymbol{\omega}_M, \\
 \text{RHS} &= (\boldsymbol{\chi}_M^T \mathbf{F}) (\mathbf{Q}^T \boldsymbol{\lambda}_M) \mathbf{Z} \\
 &\quad - \sum_{j=1}^{M-1} \left[\sum_{i=1}^3 (\boldsymbol{\chi}_M^T \mathbf{K}_i \boldsymbol{\chi}_j) (\boldsymbol{\lambda}_M^T \mathbf{H}_i \boldsymbol{\lambda}_j) \mathbf{M} + (\boldsymbol{\chi}_M^T \mathbf{K}_4 \boldsymbol{\chi}_j) (\boldsymbol{\lambda}_M^T \mathbf{S} \boldsymbol{\lambda}_j) \mathbf{H}_4 \right] \boldsymbol{\omega}_j.
 \end{aligned} \tag{6.39}$$

For this particular problem, the non-intrusive scheme is used only for the mechanical problem. For the parametric problems, in-house Matlab codes are developed to implement the NURBS discretisation.

6.5 Post-processing

In this section we explore the capability of the obtained PGD vademecum in terms of computing the derivatives with respect to the parameters.

6.5.1 Displacement

With the PGD vademecum, we may reconstruct the parametrised solution of the displacement DOF vector as

$$\hat{\mathbf{U}}_{\text{PGD}}^M(\mathbf{q}) = \sum_{i=1}^M \boldsymbol{\chi}_i \sum_A R_A(\xi, \eta, \zeta) \hat{\lambda}_A^i \sum_B P_B(\tilde{\xi}) \hat{\omega}_B^i = \sum_{i=1}^n \boldsymbol{\chi}_i (\mathbf{R}^T \boldsymbol{\lambda}_i) (\mathbf{P}^T \boldsymbol{\omega}_i). \quad (6.40)$$

Because the NURBS shape functions are in terms of the knots, for a specified set of parameters $\mathbf{q} = (\tilde{\mathbf{q}}, q_4)$, we need to compute the inverse mapping of [Equation 6.27](#) and [6.26](#), and then compute the displacement with [Equation 6.40](#).

To compute the inverse mapping which is generally nonlinear, we adopt the Newton-Raphson method. Take the 3D parametric space $\Omega_{\tilde{\mathbf{q}}}$ for example, given $\tilde{\mathbf{q}}$ and initial guess of $\boldsymbol{\Xi}_0 = (\xi_0, \eta_0, \zeta_0)$, we have

$$\tilde{\mathbf{q}} = \tilde{\mathbf{q}}(\boldsymbol{\Xi}_0) + \mathbf{J}(\boldsymbol{\Xi}_0)(\boldsymbol{\Xi} - \boldsymbol{\Xi}_0), \quad (6.41)$$

$$\implies \boldsymbol{\Xi}_{k+1} = \boldsymbol{\Xi}_k + \mathbf{J}^{-1}(\boldsymbol{\Xi}_k)[\tilde{\mathbf{q}} - \tilde{\mathbf{q}}(\boldsymbol{\Xi}_k)]. \quad (6.42)$$

The inverse mapping in the 1D parametric space I_{q_4} can be computed similarly.

6.5.2 Derivatives with respect to parameters

With [Equation 6.40](#), it is possible to compute the parametric derivatives with the PGD vademecum:

$$\frac{\partial \mathbf{U}}{\partial q_\alpha} \approx \frac{\partial \hat{\mathbf{U}}_{\text{PGD}}^M}{\partial q_\alpha} = \begin{cases} \sum_{j=1}^n \boldsymbol{\chi}_j \left(\frac{\partial \mathbf{R}^T}{\partial q_\alpha} \boldsymbol{\lambda}_j \right) (\mathbf{P}^T \boldsymbol{\omega}_j), & \alpha = 1, 2, 3, \\ \sum_{j=1}^n \boldsymbol{\chi}_j (\mathbf{R}^T \boldsymbol{\lambda}_j) \left(\frac{\partial \mathbf{P}^T}{\partial q_\alpha} \boldsymbol{\omega}_j \right), & \alpha = 4. \end{cases} \quad (6.43)$$

For the 3D space, note that

$$\frac{\partial \mathbf{R}^T}{\partial (q_1, q_2, q_3)} = \begin{bmatrix} \frac{\partial \mathbf{R}^T}{\partial q_1} \\ \frac{\partial \mathbf{R}^T}{\partial q_2} \\ \frac{\partial \mathbf{R}^T}{\partial q_3} \end{bmatrix} = \mathbf{J}^{-T} \begin{bmatrix} \frac{\partial \mathbf{R}^T}{\partial \xi} \\ \frac{\partial \mathbf{R}^T}{\partial \eta} \\ \frac{\partial \mathbf{R}^T}{\partial \zeta} \end{bmatrix}, \quad (6.44)$$

and

$$\begin{aligned} \frac{\partial R_A}{\partial \xi} &= \frac{dN_{i,p}(\xi)}{d\xi} M_{j,q}(\eta) L_{k,r}(\zeta), & A = 1, 2, \dots, \text{ncp}, \\ \frac{\partial R_A}{\partial \eta} &= N_{i,p}(\xi) \frac{dM_{j,q}(\eta)}{d\eta} L_{k,r}(\zeta), & A = 1, 2, \dots, \text{ncp}, \\ \frac{\partial R_A}{\partial \zeta} &= N_{i,p}(\xi) M_{j,q}(\eta) \frac{dL_{k,r}(\zeta)}{d\zeta}, & A = 1, 2, \dots, \text{ncp}. \end{aligned} \quad (6.45)$$

For the 1D space, we have

$$\frac{d\mathbf{P}^T}{dq_4} = \frac{d\mathbf{P}^T}{d\tilde{\xi}} \frac{d\tilde{\xi}}{dq_4} = \left(\sum_a \frac{dP_a(\tilde{\xi})}{d\tilde{\xi}} \tilde{q}_a \right)^{-1} \frac{d\mathbf{P}^T}{d\tilde{\xi}}, \quad (6.46)$$

and

$$\frac{dP_B}{d\tilde{\xi}} = \frac{d\tilde{N}_{i,p}}{d\tilde{\xi}}, \quad B = 1, 2, \dots, \text{mcp}. \quad (6.47)$$

In addition, since the engineering parameters are of more interest, recalling the change of variables in [Equation 6.5](#), we may compute

$$\begin{aligned} \frac{\partial \mathbf{U}}{\partial E_1} &= \frac{\partial \mathbf{U}}{\partial q_1} \frac{\partial q_1}{\partial E_1} + \frac{\partial \mathbf{U}}{\partial q_2} \frac{\partial q_2}{\partial E_1}, \\ \frac{\partial \mathbf{U}}{\partial E_2} &= \frac{\partial \mathbf{U}}{\partial q_1} \frac{\partial q_1}{\partial E_2} + \frac{\partial \mathbf{U}}{\partial q_2} \frac{\partial q_2}{\partial E_2}, \\ \frac{\partial \mathbf{U}}{\partial \nu_{12}} &= \frac{\partial \mathbf{U}}{\partial q_3}, \\ \frac{\partial \mathbf{U}}{\partial G_{12}} &= \frac{\partial \mathbf{U}}{\partial q_4}, \end{aligned} \quad (6.48)$$

where

$$\begin{aligned} \frac{\partial q_1}{\partial E_1} &= \frac{1 - 2\nu_{12}^2 E_2 / E_1}{(1 - \nu_{12}^2 E_2 / E_1)^2}, \\ \frac{\partial q_1}{\partial E_2} &= \frac{\nu_{12}^2}{(1 - \nu_{12}^2 E_2 / E_1)^2}, \\ \frac{\partial q_2}{\partial E_1} &= -E_2 / E_1^2, \\ \frac{\partial q_2}{\partial E_2} &= 1 / E_1. \end{aligned} \quad (6.49)$$

In case of requiring the reaction forces \mathbf{F}_d , the static condensation has to be invoked. Detailed derivation is arranged in [Appendix A](#). Considering the case of

$$\mathbf{K}\mathbf{U} = \begin{bmatrix} \mathbf{K}_d & \mathbf{K}_{df} \\ \mathbf{K}_{df}^T & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \mathbf{U}_d \\ \mathbf{U}_f \end{bmatrix} = \begin{bmatrix} \mathbf{F}_d \\ \mathbf{F}_f \end{bmatrix}, \quad (6.50)$$

where subscript d denotes the fixed nodes (on which Dirichlet boundary condition is applied), and subscript f denotes the free nodes. The prescribed displacements \mathbf{U}_d and load \mathbf{F}_f are both known.

It is trivial to obtain

$$\begin{aligned} \mathbf{U}_f &= \mathbf{K}_f^{-1} (\mathbf{F}_f - \mathbf{K}_{df}^T \mathbf{U}_d), \\ \mathbf{F}_d &= \mathbf{K}_d \mathbf{U}_d + \mathbf{K}_{df} \mathbf{U}_f, \end{aligned} \quad (6.51)$$

thus the parametric derivatives are

$$\frac{\partial \mathbf{F}_d}{\partial q_\alpha} = \frac{\partial \mathbf{K}_d}{\partial q_\alpha} \mathbf{U}_d + \frac{\partial \mathbf{K}_{df}}{\partial q_\alpha} \mathbf{U}_f + \mathbf{K}_{df} \frac{\partial \mathbf{U}_f}{\partial q_\alpha}, \quad \alpha = 1, 2, 3, 4. \quad (6.52)$$

In the online phase, both \mathbf{U}_f and $\partial \mathbf{U}_f / \partial q_\alpha$ are approximated with PGD solution.

6.6 Numerical examples

Now we consider a classical mechanical model, a rectangular plate with a round open hole in the centre, for a benchmark example. The length and height of the plate are 900 mm and 300 mm respectively, while the radius of the circle is 50 mm, and the thickness is 1.26 mm. The FE mesh of the mechanical model is shown in [Figure 6.2](#).

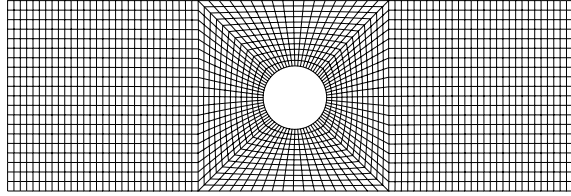


Figure 6.2: FE mesh of the mechanical model.

Two load cases (LCs), namely LC A and LC B, are considered, as shown in [Figure 6.3](#). For LC A, the left end is prescribed with inhomogeneous nodal displacements, while the right end is subjected to inhomogeneous nodal forces. For LC B, both left and right ends are prescribed with inhomogeneous nodal displacements.

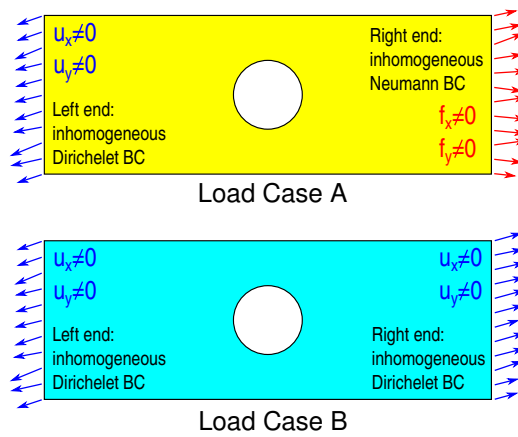


Figure 6.3: Load cases of the mechanical model.

To obtain a reference solution, material properties are chosen as:

$$\begin{aligned} E_1 &= 25 \text{ GPa}, & E_2 &= 19 \text{ GPa}, \\ \nu_{12} &= 0.25, & G_{12} &= 3 \text{ GPa}. \end{aligned}$$

For both LCs, the reference displacement fields computed with standard FE method are plotted in [Figure 6.4](#). For visualisation purposes, the deformed shape is amplified with a proper scale factor.

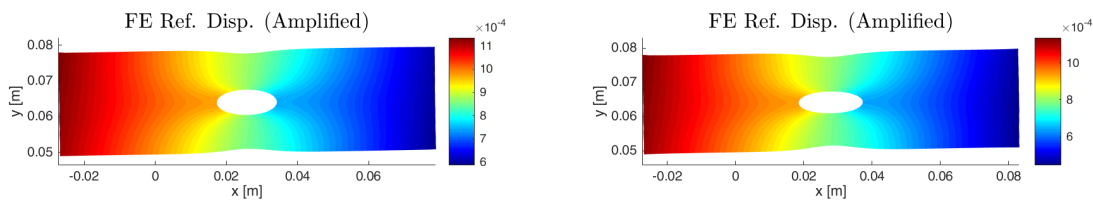


Figure 6.4: Reference displacement field computed with standard FE method, amplified for visualisation. Left: LC A. Right: LC B.

Powered by the k -refinement methodology of isogeometric analysis, it is possible to specify the number of knot spans (elements) and interpolating polynomial degree of the parametric spaces. For instance, when the knot span numbers `nspan` is set to 10 for each parameter, that is to say, $\Omega_{\bar{q}}$ is discretised with $10 \times 10 \times 10$ elements, and I_{q_4} is discretised with 10 elements.

With `nspan` = 1, the displacement fields are plotted with variable order of B-spline polynomials in [Figure 6.5](#) and [6.6](#). All the displacement fields are computed with 5 PGD modes. It is observed that polynomials with higher degree improve the error which is computed by $\text{Err} = \|\mathbf{U}_{\text{ref}} - \mathbf{U}_{\text{ref}}\| / \|\mathbf{U}_{\text{ref}}\|$. However, it can be seen that the displacement contours are almost the same for all the results. For a better comparison of the spatial distribution of errors, the differences from the reference displacement field $\mathbf{U}_{\text{ref}} - \mathbf{U}_{\text{ref}}$ is plotted in [Figure 6.7](#) and [6.8](#).

For the mechanical model with LC A, the error concentration areas include both the hole edge and the right end where loads are applied. For the model with LC B, the error concentration areas emerge from the hole edge, and expands to the bulk area in the plate as the polynomial order p elevates.

By setting the polynomial order p from 1 to 6, and the knot span `nspan` from 1 to 6, we computed all the 36 cases to investigate the convergence of the displacement field, as shown in [Figure 6.9](#). The displacement field computed with $p = 6$ and `nspan` = 6 is taken as the reference.

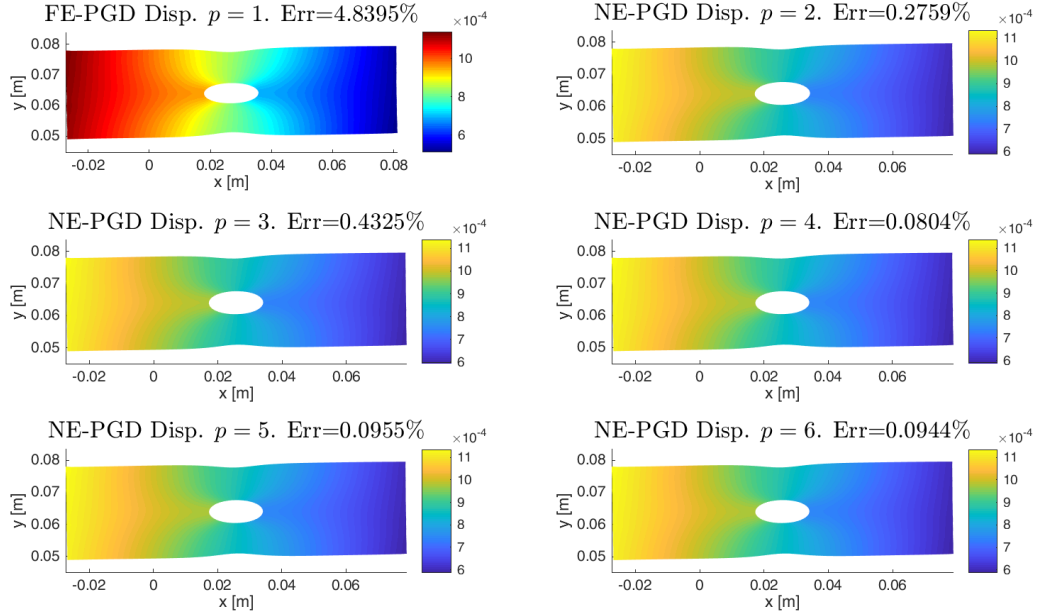


Figure 6.5: Displacement fields computed with PGD with different discretisations of parameter spaces for LC A.

Similarly, the sensitivities with respect to the engineering parameters are also computed, using the same strategy. The reference results is plotted in [Figure 6.10](#) and [6.11](#). For the mechanical model with LC A, areas on the right end where loads are applied have higher sensitivities with respect to the parameters E_1 , E_2 and ν_{12} , while the right part of the hole edge has higher sensitivities with respect to G_{12} . For the model with LC B, areas near the hole edge have higher sensitivities with respect to E_1 , E_2 and G_{12} , while the upper and lower free end have higher sensitivities with respect to ν_{12} .

The convergence analysis is plotted in [Figure 6.12](#) and [6.13](#). It can be seen that for both LCs, the relative errors for $\partial\mathbf{U}/\partial E_2$ and $\partial\mathbf{U}/\partial G_{12}$ are significantly improved. For example, in LC A, the relative error of $\partial\mathbf{U}/\partial E_2$ dropped more than 100% from $p = 1$ and $\text{nspan} = 1$ to $p = 6$ and $\text{nspan} = 6$.

6.7 Summary

In the PGD separated representation, the 4-dimensional parametric space is separated as the tensor product of a 3D space and a 1D space. The parametric spaces are rep-

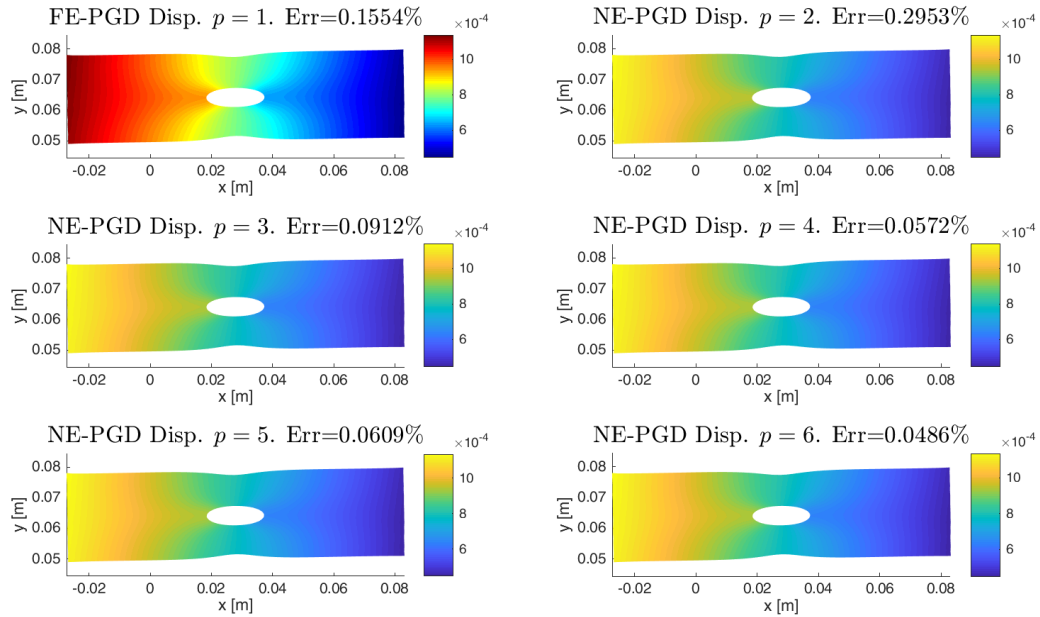


Figure 6.6: Displacement fields computed with PGD with different discretisations of parameter spaces for LC B.

resented with NURBS geometrically, and the parametric functions are approximated with B-spline shape functions accordingly. Therefore, k -refinement methodology from isogeometric analysis can be used.

It is shown that the PGD results are improved by B-spline shape functions with higher order, and the convergences of both displacement field and the sensitivities with respect to the parameters are investigated.

We would like to address that although FE is used for the spatial discretisation throughout this chapter, it is possible to replace it with NURBS shape functions as IGA does. Consequently, the NE-PGD framework could be fully enhanced with the higher continuity feature of NURBS shape functions.

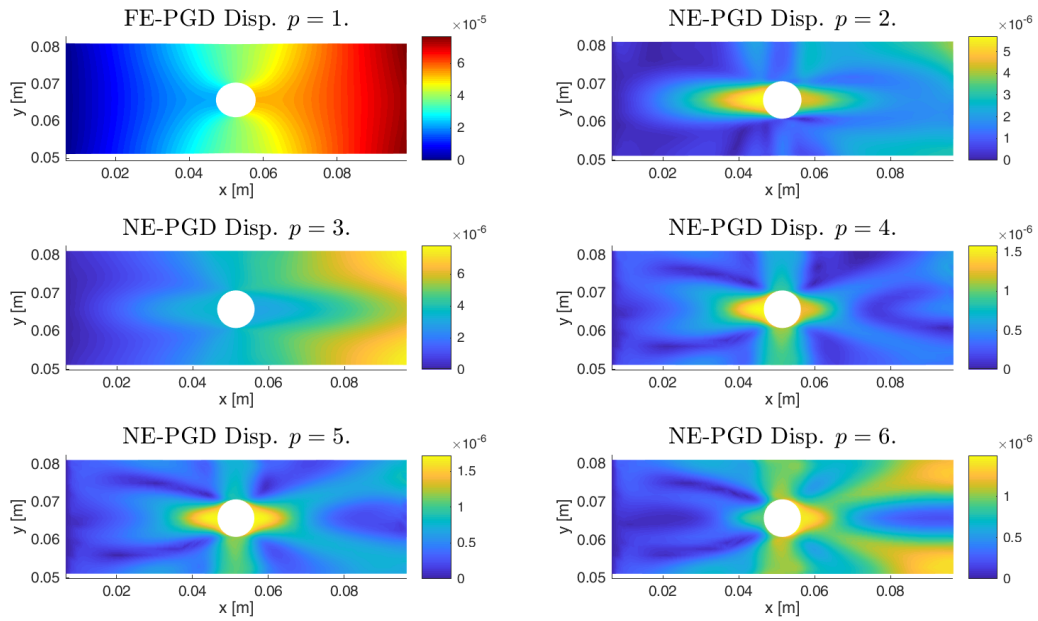


Figure 6.7: Differences of displacement fields computed with PGD with different discretizations of parameter spaces for LC A.

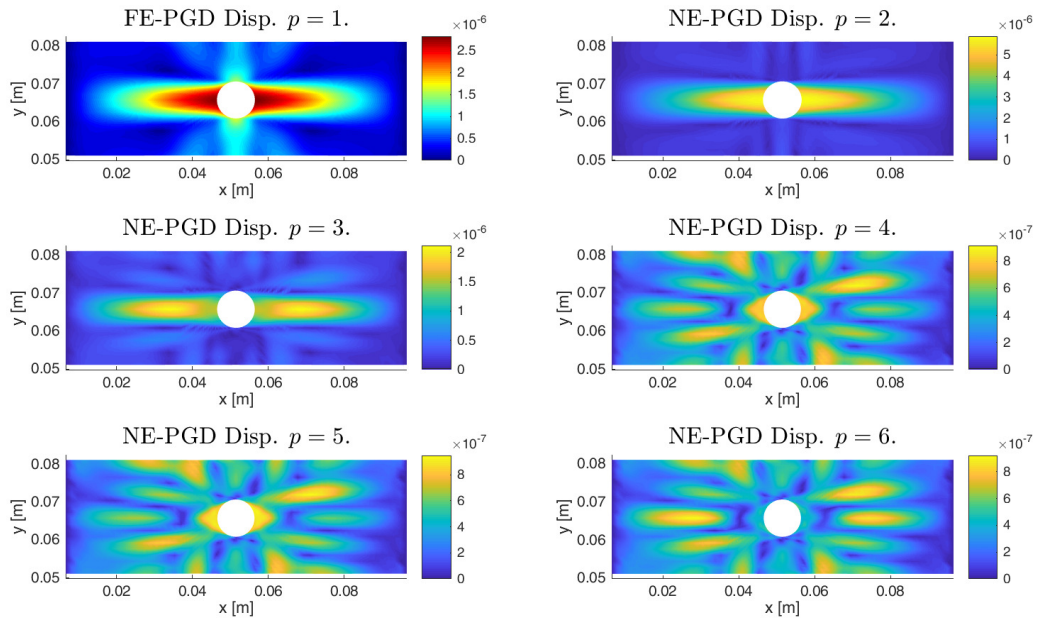


Figure 6.8: Differences of displacement fields computed with PGD with different discretizations of parameter spaces for LC B.

6. NURBS SHAPE FUNCTION DISCRETISATION OF PARAMETER SPACES

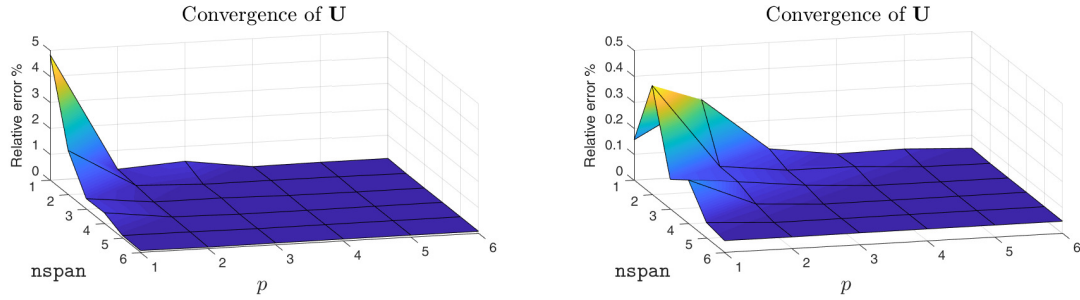


Figure 6.9: Convergence of the displacement field. Left: LC A. Right: LC B.

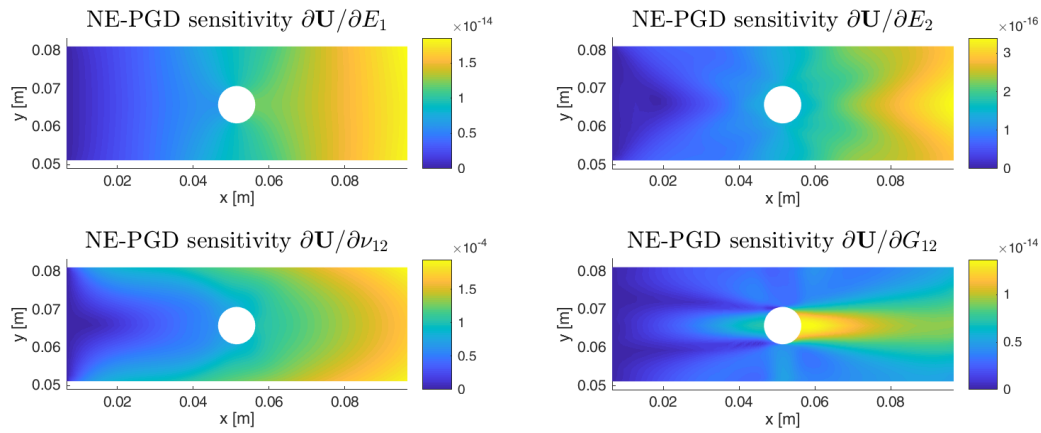


Figure 6.10: Reference sensitivity fields computed with PGD, parametric space polynomial order $p = 6$, number of knot span (element) $\mathbf{nspan} = 6$, LC A.

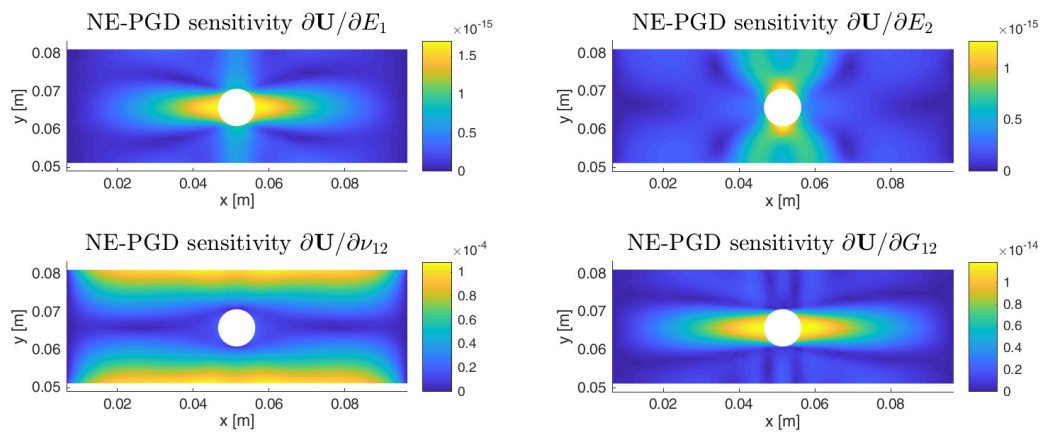


Figure 6.11: Reference sensitivity fields computed with PGD, parametric space polynomial order $p = 6$, number of knot span (element) $\mathbf{nspan} = 6$, LC B.

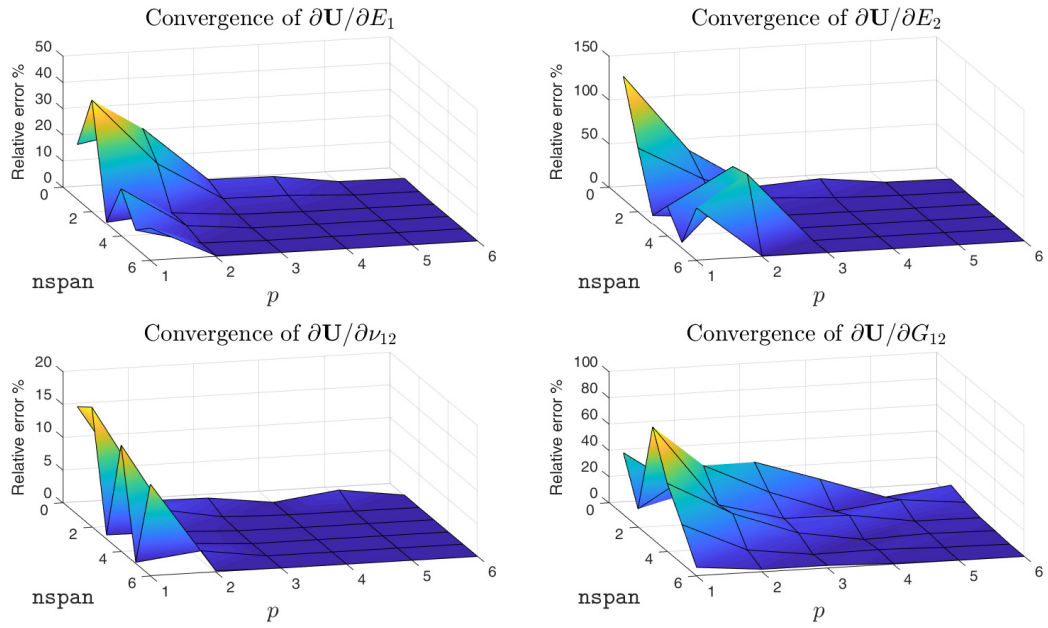


Figure 6.12: Convergence of the sensitivities. LC A.

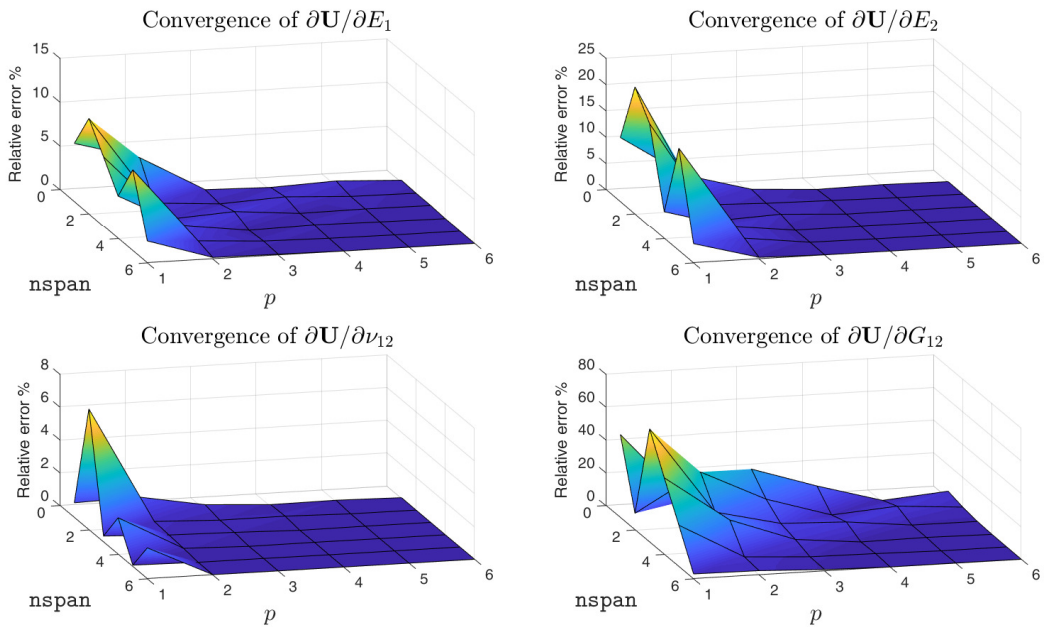


Figure 6.13: Convergence of the sensitivities. LC B.

Chapter 7

Nonlinear problems with PGD

In this chapter, we will enter a more colourful world of nonlinear problems. In biomechanics, many biological tissues have been observed nonlinear behaviours, such as hyperelastic large deformation and large strain, of mechanical response to loads. There are numerous constitutive models describing the stress-strain relationships for those nonlinear mechanisms, details can be found in numerous literature such as monographs by [Simo and Hughes \[1998\]](#), [Holzapfel \[2000\]](#), [Bonet and Wood \[2008\]](#).

As a first step towards the application of PGD framework for parametrised problems in nonlinear elasticity, this chapter is dedicated to the boundary value problems in large strain theory with St.Venant-Kirchhoff material model. There are several previous work on this topic, such as [Niroomandi et al. \[2013a,b\]](#), [Ladevèze \[2016\]](#). In those works, however, either high order nonlinear terms were discarded or additional techniques, such as the Asymptotic Numerical Method (ANM) or the large time increment (LATIN) method [Ladevèze \[1999\]](#), were employed. In this chapter we introduce a simple but effective PGD framework with Picard linearisation of the nonlinear operators. Both spatial and parametric domain are discretised with standard FE method. For the spatial domain, the total Lagrange formulation is used to write the FE equations.

Due to the nonlinearity of the problems, conventional PGD framework does not applicable any more. This requires linearisation of the formulas. In this case, Picard

linearisation is used since it is naturally compatible with the alternative direction iteration in the conventional PGD framework. To generalise to the nonlinear problem, a nonlinear term is added to the right-hand side, which makes the non-intrusive framework not available. Therefore, the implementation throughout this chapter has to be intrusive. In-house Matlab codes are programmed for the implementation.

Notations in this chapter are slightly modified from previous chapters to better adapt the nonlinear formulations.

7.1 A brief review of large strain theory

In this section, for better explanation in consistent notation and further extension to PGD formulation, we briefly review large strain theory in continuum mechanics under Lagrangian description.

7.1.1 Large strain kinematics

Let Ω_0 be a deformable continuum body in reference configuration with material coordinates denoted by $\mathbf{X} : \Omega_0 \rightarrow \mathbb{R}^3$, and its deformed or current configuration Ω with spatial coordinates denoted by $\mathbf{x} : \Omega \rightarrow \mathbb{R}^3$. The deformation is a map from the reference configuration to current configuration $\phi : \mathbf{X} \mapsto \mathbf{x} = \phi(\mathbf{X})$, as illustrated in [Figure 7.1](#). The deformation gradient $\mathbf{F} \in \mathbb{R}^{3 \times 3}$ is defined as

$$\mathbf{F} = \frac{\partial \phi(\mathbf{X})}{\partial \mathbf{X}} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} + \mathbf{I} = \nabla_0 \mathbf{u} + \mathbf{I}, \quad (7.1)$$

where $\nabla_0 := \partial/\partial \mathbf{X}$, whilst $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ denotes the identity tensor and \mathbf{u} the Lagrangian displacement of a material point

$$\mathbf{u}(\mathbf{X}) = \mathbf{x} - \mathbf{X} = \phi(\mathbf{X}) - \mathbf{X}. \quad (7.2)$$

For the convenience of establishing work-energy equilibrium, even the problem is time-independent, we define the virtual (Lagrangian) velocity as

$$\mathbf{v} = \dot{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial t}, \quad (7.3)$$

and rate of deformation gradient is derived with [Equation 7.1](#) and [7.3](#):

$$\dot{\mathbf{F}} = \nabla_0 \mathbf{v}. \quad (7.4)$$

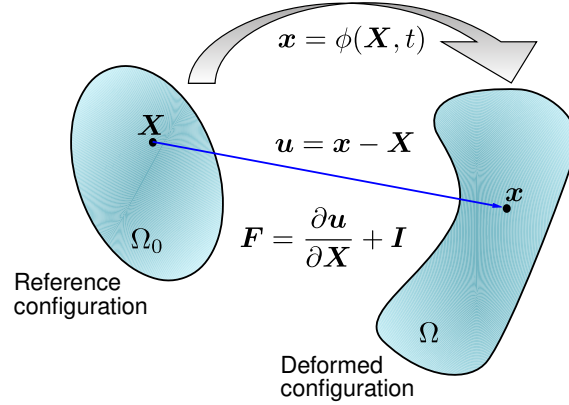


Figure 7.1: Deformation of a continuum body.

To measure the deformation under Lagrange description, we use Green-Lagrange strain which is defined as

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}). \quad (7.5)$$

7.1.2 Stress and equilibrium

Considering that we will be focused on St.Venant-Kirchhoff constitutive relationship, the second Piola-Kirchhoff stress, denoted by \mathbf{S} , is taken as the stress measure. It is well-known that Green-Lagrange strain and second Piola-Kirchhoff stress consists a work conjugate pair in large strain theory. In rate form, the inner virtual work of the system, denoted as δW_{int} , reads

$$\delta W_{\text{int}} = \int_{\Omega_0} \mathbf{S} : \delta \dot{\mathbf{E}} dV, \quad (7.6)$$

where the virtual strain rate is

$$\delta \dot{\mathbf{E}} = \frac{1}{2}(\delta \dot{\mathbf{F}}^T \mathbf{F} + \mathbf{F}^T \delta \dot{\mathbf{F}}), \quad (7.7)$$

and the variation of deformation gradient rate is

$$\delta \dot{\mathbf{F}} = \nabla_0 \delta \mathbf{v}. \quad (7.8)$$

Now let $\partial\Omega_0$ be the boundary of Ω_0 , we assume that

$$\partial\Omega_0 = \bar{\Gamma}_D \cup \bar{\Gamma}_N \quad \text{and} \quad \Gamma_D \cap \Gamma_N = \emptyset, \quad (7.9)$$

where Γ_D and Γ_N is the part of Γ_N with Dirichlet and Neumann boundary condition prescribed respectively. To solve the displacement field \mathbf{u} from a boundary value

problem, with prescribed Dirichlet boundary condition $\mathbf{u} = \mathbf{u}_D$ on Γ_D , we introduce the trial function space U and the test function space V as

$$\begin{aligned} U &:= \{\mathbf{u} \in H^1(\Omega_0) : \mathbf{u} = \mathbf{u}_D \text{ on } \Gamma_D\}, \\ V &:= \{\delta\mathbf{v} \in H^1(\Omega_0) : \delta\mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}. \end{aligned} \quad (7.10)$$

To simplify the formulas, we ignore all external body forces. The external virtual work, denoted as δW_{ext} , under surface traction loading $\mathbf{T} = T_0 \hat{\mathbf{N}}$, where T_0 is the amplitude and $\hat{\mathbf{N}}$ is unit normal vector on the surface, reads

$$\delta W_{\text{ext}} = \int_{\Gamma_N} \mathbf{T} \cdot \delta\mathbf{v} \, dA = \int_{\Gamma_N} T_0 \hat{\mathbf{N}} \cdot \delta\mathbf{v} \, dA. \quad (7.11)$$

It is convenient to define the abstract forms as:

$$\begin{aligned} a(\mathbf{u}, \delta\mathbf{v}) &:= \delta W_{\text{int}}(\mathbf{u}, \delta\mathbf{v}) = \int_{\Omega_0} \mathbf{S} : \delta\dot{\mathbf{E}} \, dV, \\ l(\delta\mathbf{v}) &:= \delta W_{\text{ext}}(\delta\mathbf{v}) = \int_{\Gamma_N} \mathbf{T} \cdot \delta\mathbf{v} \, dA. \end{aligned} \quad (7.12)$$

where $a(\mathbf{u}, \delta\mathbf{v})$ and $l(\delta\mathbf{v})$ are abstract forms that linearly depend on $\delta\mathbf{v}$, but the former is nonlinearly dependent on \mathbf{u} . Now the principle of virtual work can be written as

$$\begin{aligned} \delta W &= \delta W_{\text{int}} - \delta W_{\text{ext}} = 0 \\ \implies a(\mathbf{u}, \delta\mathbf{v}) &= l(\delta\mathbf{v}), \end{aligned} \quad (7.13)$$

which introduces the weak form of the problem: find $\mathbf{u} \in U$, such that [Equation 7.13](#) holds $\forall \delta\mathbf{v} \in V$.

7.2 PGD formulation with Picard linearisation

As mentioned before, PGD is a technique to build the reduced order model, especially powerful in dealing with PDE with multiple parameters that are stated in the equations. In particular, when dependency on the parameters is able to be represented in separated functions, the quantity of interest is assumed to have a decomposition of a certain set of reduced basis. Each reduced basis is a product of the separated functions, which we prefer to mention as modes. The modes are computed in the *offline* phase sequentially, generating the so-called PGD *vademecum*, which enables the efficient *online* computations. In this section the routine of PGD offline computation is reviewed, addressing that the alternative direction iteration, also known as Picard iteration, linearises the nonlinear form in [Equation 7.13](#).

7.2.1 Separated representation

Following the general notation introduced in [Chapter 3](#) and [Section 7.1](#), without loosing generality, we take the material property as the PGD extra-coordinate. Consider the range of shear modulus μ , denoted by I_μ , is known, as discussed before, displacement \mathbf{u} is considered as a function of both material location $\mathbf{X} \in \Omega_0$ and shear modulus $\mu \in I_\mu$, denoted as $\mathbf{u} = \mathbf{u}(\mathbf{X}, \mu) \in U \otimes L^2(I_\mu)$. Now PGD separated representation [Equation 3.13](#) can be written as

$$\mathbf{u}(\mathbf{X}, \mu) \approx \mathbf{u}^n(\mathbf{X}, \mu) = \sum_{i=1}^n \chi_i(\mathbf{X}) M_i(\mu), \quad (7.14)$$

where $\chi_i(\mathbf{X}) \in U$ is a function of material location referred to the i th spatial mode, and $M_i(\mu) \in L^2(I_\mu)$ a function of shear modulus referred to the i th parametric mode.

Following [Equation 3.11](#), we generalise the abstract forms [Equation 7.12](#) by integrating over I_μ :

$$\begin{aligned} \mathcal{A}(\mathbf{u}, \delta \mathbf{v}) &:= \int_{I_\mu} a(\mathbf{u}, \delta \mathbf{v}) d\mu, \\ \mathcal{L}(\delta \mathbf{v}) &:= \int_{I_\mu} l(\delta \mathbf{v}) d\mu. \end{aligned} \quad (7.15)$$

Note that the linearity of PGD generalised abstract forms $\mathcal{A}(\mathbf{u}, \delta \mathbf{v})$ and $\mathcal{L}(\delta \mathbf{v})$ depends on that of the original forms $a(\mathbf{u}, \delta \mathbf{v})$ and $l(\delta \mathbf{v})$ respectively.

7.2.2 Picard linearisation

As introduced previously, the PGD generalised abstract forms [Equation 7.15](#) are generally nonlinear on \mathbf{u} . Due to the nonlinearity, now unlike [Equation 3.15](#), we have:

$$\mathcal{A}(\chi_n M_n, \delta \mathbf{v}) \neq \mathcal{L}(\delta \mathbf{v}) - \mathcal{A}(\mathbf{u}^{n-1}, \delta \mathbf{v}). \quad (7.16)$$

To overcome this fatal problem which disables the alternative direction iteration, Picard linearisation is adopted since it is naturally consistent with this iteration scheme. As it is usually the case, we assume that it is possible to divide abstract form $a(\mathbf{u}, \delta \mathbf{v})$ into a linear part and a nonlinear part:

$$a(\mathbf{u}, \delta \mathbf{v}) = a_L(\mathbf{u}, \delta \mathbf{v}) + a_{NL}(\mathbf{u}, \delta \mathbf{v}), \quad (7.17)$$

where the linear part $a_L(\mathbf{u}, \delta \mathbf{v})$ is bilinear as we discussed before, while $a_{NL}(\mathbf{u}, \delta \mathbf{v})$ is nonlinear on \mathbf{u} . In iterations with Picard linearisation, the nonlinear part is approx-

imated with the formerly obtained results $\tilde{\mathbf{u}}$:

$$a_{\text{NL}}(\mathbf{u}, \delta \mathbf{v}) \approx a_{\text{NL}}(\tilde{\mathbf{u}}, \delta \mathbf{v}). \quad (7.18)$$

Then we obtain the linearised weak form of problem [Equation 7.13](#):

$$a_{\text{L}}(\mathbf{u}, \delta \mathbf{v}) \approx l(\delta \mathbf{v}) - a_{\text{NL}}(\tilde{\mathbf{u}}, \delta \mathbf{v}). \quad (7.19)$$

Accordingly, we the PGD generalised weak form is defined as following:

$$\begin{aligned} \mathcal{A}_{\text{L}}(\mathbf{u}, \delta \mathbf{v}) &:= \int_{I_\mu} a_{\text{L}}(\mathbf{u}, \delta \mathbf{v}) d\mu, \\ \mathcal{A}_{\text{NL}}(\mathbf{u}, \delta \mathbf{v}) &:= \int_{I_\mu} a_{\text{NL}}(\mathbf{u}, \delta \mathbf{v}) d\mu, \end{aligned} \quad (7.20)$$

and we have also $\mathcal{A}(\mathbf{u}, \delta \mathbf{v}) = \mathcal{A}_{\text{L}}(\mathbf{u}, \delta \mathbf{v}) + \mathcal{A}_{\text{NL}}(\mathbf{u}, \delta \mathbf{v})$. Now the linearised weak form reads

$$\mathcal{A}_{\text{L}}(\mathbf{u}, \delta \mathbf{v}) \approx \mathcal{L}(\delta \mathbf{v}) - \mathcal{A}_{\text{NL}}(\tilde{\mathbf{u}}, \delta \mathbf{v}). \quad (7.21)$$

Considering the PGD alternative direction iteration, when the first $n - 1$ modes are obtained, that is \mathbf{u}^{n-1} is known. The natural option we have is letting $\tilde{\mathbf{u}} = \mathbf{u}^{n-1}$ for the Picard linearisation, so that the sequential enrichment of the modes is enabled as

$$\mathbf{u}^n = \tilde{\mathbf{u}} + \chi_n M_n = \mathbf{u}^{n-1} + \chi_n M_n, \quad (7.22)$$

$$\mathcal{A}_{\text{L}}(\chi_n M_n, \delta \mathbf{v}) = \mathcal{L}(\delta \mathbf{v}) - \mathcal{A}_{\text{NL}}(\mathbf{u}^{n-1}, \delta \mathbf{v}) - \mathcal{A}_{\text{L}}(\mathbf{u}^{n-1}, \delta \mathbf{v}). \quad (7.23)$$

The alternative direction scheme now reads:

1. Assuming \mathbf{u}^{n-1} and M_n are known, let $\delta \mathbf{v} = M_n \delta \chi_n$, solve the mechanical problem for χ_n from

$$\mathcal{A}_{\text{L}}(\chi_n M_n, M_n \delta \chi_n) = \mathcal{L}(M_n \delta \chi_n) - \mathcal{A}_{\text{NL}}(\mathbf{u}^{n-1}, M_n \delta \chi_n) - \mathcal{A}_{\text{L}}(\mathbf{u}^{n-1}, M_n \delta \chi_n); \quad (7.24)$$

2. Assuming \mathbf{u}^{n-1} and χ_n are known, let $\delta \mathbf{v} = \chi_n \delta M_n$, solve the mechanical problem for M_n from

$$\mathcal{A}_{\text{L}}(\chi_n M_n, \chi_n \delta M_n) = \mathcal{L}(\chi_n \delta M_n) - \mathcal{A}_{\text{NL}}(\mathbf{u}^{n-1}, \chi_n \delta M_n) - \mathcal{A}_{\text{L}}(\mathbf{u}^{n-1}, \chi_n \delta M_n). \quad (7.25)$$

Remark 7.1 (On the choice of linearisation methods). *We choose Picard linearisation instead of Newton-Raphson method. Despite that Picard linearisation is consistent with PGD alternative direction iteration, it does not require rewriting the problem in an incremental formulation and computing the Jacobians. The price we have to pay is achieving a slower convergence rate.*

Remark 7.2 (Linearised alternative direction iteration). *By direct comparison of the alternative direction iteration schemes before and after Picard linearisation, it can be seen the latter scheme has only added a nonlinear term at the right-hand side. In fact, it is possible to rewrite Equation 7.23 as*

$$\mathcal{A}_L(\chi_n M_n, \delta \mathbf{v}) = \mathcal{L}(\delta \mathbf{v}) - \mathcal{A}(\mathbf{u}^{n-1}, \delta \mathbf{v}). \quad (7.26)$$

However, to make the formulation conceptually clear, this rewritten is not adopted in following contents.

7.3 Application to St.Venant-Kirchhoff material with discretised formulation

The macroscopic mechanical behaviour of a material under certain loadings is described by a corresponding constitutive law, or stress-strain relationship. For hyperelastic materials, strain energy density Ψ is used to characterise the energy stored during deformation of a material.

The simplest example of a hyperelastic material is the St.Venant-Kirchhoff model, which is defined by a strain energy function Ψ as

$$\Psi(\mathbf{E}) = \frac{1}{2} \lambda (\text{tr } \mathbf{E})^2 + \mu \mathbf{E} : \mathbf{E}. \quad (7.27)$$

where λ and μ are the Lamé coefficients, in particular, μ is the shear modulus. The second Piola-Kirchhoff stress is computed as

$$\mathbf{S} = \frac{\partial \Psi(\mathbf{E})}{\partial \mathbf{E}} = \lambda (\text{tr } \mathbf{E}) \mathbf{I} + 2\mu \mathbf{E}. \quad (7.28)$$

Note that the relationship between \mathbf{S} and \mathbf{E} is linear.

We now derive in detail the formulation. All the formulas is expanded and expressed by \mathbf{u} and $\delta\mathbf{v}$. Considering

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}) = \frac{1}{2}(\nabla_0 \mathbf{u} + \nabla_0^T \mathbf{u} + \nabla_0^T \mathbf{u} \nabla_0 \mathbf{u}) = \nabla_0^s \mathbf{u} + \frac{1}{2} \nabla_0^T \mathbf{u} \nabla_0 \mathbf{u}, \quad (7.29)$$

$$\begin{aligned} \delta \dot{\mathbf{E}} &= \frac{1}{2}(\delta \dot{\mathbf{F}}^T \mathbf{F} + \mathbf{F}^T \delta \dot{\mathbf{F}}) = \frac{1}{2} \left[\nabla_0^T \delta \mathbf{v} (\nabla_0 \mathbf{u} + \mathbf{I}) + (\nabla_0^T \mathbf{u} + \mathbf{I}) \nabla_0 \delta \mathbf{v} \right] \\ &= \nabla_0^s \delta \mathbf{v} + \frac{1}{2} (\nabla_0^T \delta \mathbf{v} \nabla_0 \mathbf{u} + \nabla_0^T \mathbf{u} \nabla_0 \delta \mathbf{v}), \end{aligned} \quad (7.30)$$

$$\text{tr } \mathbf{E} = \text{tr } \nabla_0^s \mathbf{u} + \text{tr } \frac{\nabla_0^T \mathbf{u} \nabla_0 \mathbf{u}}{2} = \text{tr } \nabla_0 \mathbf{u} + \frac{1}{2} \text{tr}(\nabla_0^T \mathbf{u} \nabla_0 \mathbf{u}), \quad (7.31)$$

where the symmetric gradient operator is defined as $\nabla_0^s = (\nabla_0 + \nabla_0^T)/2$. We may write the second Piola-Kirchhoff stress as a function of the displacement \mathbf{u} :

$$\mathbf{S} = \lambda(\text{tr } \nabla_0 \mathbf{u}) \mathbf{I} + 2\mu \nabla_0^s \mathbf{u} + \frac{\lambda \mathbf{I}}{2} \text{tr}(\nabla_0^T \mathbf{u} \nabla_0 \mathbf{u}) + \mu \nabla_0^T \mathbf{u} \nabla_0 \mathbf{u}. \quad (7.32)$$

Then with [Equation 7.30](#) and [7.32](#) we can explicitly calculate the abstract form $a(\mathbf{u}, \delta\mathbf{v})$ as

$$\begin{aligned} a(\mathbf{u}, \delta\mathbf{v}) &= \int_{\Omega_0} \mathbf{S} : \delta \dot{\mathbf{E}} dV = \underbrace{\int_{\Omega_0} [\lambda(\text{tr } \nabla_0 \mathbf{u})(\text{tr } \nabla_0 \delta \mathbf{v}) + 2\mu \nabla_0^s \mathbf{u} : \nabla_0 \delta \mathbf{v}] dV}_{\text{linear part}} \\ &\quad + \int_{\Omega_0} \lambda(\text{tr } \nabla_0 \mathbf{u}) \nabla_0 \mathbf{u} : \nabla_0 \delta \mathbf{v} dV \\ &\quad + \int_{\Omega_0} 2\mu \nabla_0 \mathbf{u} \nabla_0^s \mathbf{u} : \nabla_0 \delta \mathbf{v} dV \\ &\quad + \int_{\Omega_0} \frac{\lambda}{2} \text{tr}(\nabla_0^T \mathbf{u} \nabla_0 \mathbf{u}) (\nabla_0 \mathbf{u} + \mathbf{I}) : \nabla_0 \delta \mathbf{v} dV \\ &\quad + \int_{\Omega_0} \mu (\nabla_0 \mathbf{u} + \mathbf{I}) \nabla_0^T \mathbf{u} \nabla_0 \mathbf{u} : \nabla_0 \delta \mathbf{v} dV. \end{aligned} \quad (7.33)$$

Applying the Picard scheme, we divide $a(\mathbf{u}, \delta\mathbf{v})$ into a linear part and a nonlinear part

$$\begin{aligned} a_L(\mathbf{u}, \delta\mathbf{v}) &= \int_{\Omega_0} [\lambda(\text{tr } \nabla_0 \mathbf{u})(\text{tr } \nabla_0 \delta \mathbf{v}) + 2\mu \nabla_0^s \mathbf{u} : \nabla_0 \delta \mathbf{v}] dV, \\ a_{NL}(\mathbf{u}, \delta\mathbf{v}) &= \int_{\Omega_0} \lambda(\text{tr } \nabla_0 \mathbf{u}) \nabla_0 \mathbf{u} : \nabla_0 \delta \mathbf{v} dV + \int_{\Omega_0} 2\mu \nabla_0 \mathbf{u} \nabla_0^s \mathbf{u} : \nabla_0 \delta \mathbf{v} dV \\ &\quad + \int_{\Omega_0} \frac{\lambda}{2} \text{tr}(\nabla_0^T \mathbf{u} \nabla_0 \mathbf{u}) (\nabla_0 \mathbf{u} + \mathbf{I}) : \nabla_0 \delta \mathbf{v} dV \\ &\quad + \int_{\Omega_0} \mu (\nabla_0 \mathbf{u} + \mathbf{I}) \nabla_0^T \mathbf{u} \nabla_0 \mathbf{u} : \nabla_0 \delta \mathbf{v} dV. \end{aligned} \quad (7.34)$$

Accordingly, the generalised abstract forms can also be divided and calculated explicitly

$$\begin{aligned}
 \mathcal{A}_L(\mathbf{u}, \delta \mathbf{v}) &= \int_{I_\mu} \int_{\Omega_0} [\lambda(\text{tr } \nabla_0 \mathbf{u})(\text{tr } \nabla_0 \delta \mathbf{v}) + 2\mu \nabla_0^s \mathbf{u} : \nabla_0 \delta \mathbf{v}] dV d\mu, \\
 \mathcal{A}_{NL}(\mathbf{u}, \delta \mathbf{v}) &= \int_{I_\mu} \int_{\Omega_0} \lambda(\text{tr } \nabla_0 \mathbf{u}) \nabla_0 \mathbf{u} : \nabla_0 \delta \mathbf{v} dV d\mu \\
 &\quad + \int_{I_\mu} \int_{\Omega_0} 2\mu \nabla_0 \mathbf{u} \nabla_0^s \mathbf{u} : \nabla_0 \delta \mathbf{v} dV d\mu \\
 &\quad + \int_{I_\mu} \int_{\Omega_0} \frac{\lambda}{2} \text{tr}(\nabla_0^T \mathbf{u} \nabla_0 \mathbf{u})(\nabla_0 \mathbf{u} + \mathbf{I}) : \nabla_0 \delta \mathbf{v} d\mu \\
 &\quad + \int_{I_\mu} \int_{\Omega_0} \mu(\nabla_0 \mathbf{u} + \mathbf{I}) \nabla_0^T \mathbf{u} \nabla_0 \mathbf{u} : \nabla_0 \delta \mathbf{v} dV d\mu.
 \end{aligned} \tag{7.35}$$

7.3.1 Alternating direction iteration scheme

To derive the detailed alternative direction iteration scheme, we use LHS and RHS to denote left- and right-hand side respectively. The derivation of the mechanical problem [Equation 7.24](#) is only detailed, and the resultant formulas can also be used for the parametric problem after proper modifications.

For the mechanical problem [Equation 7.24](#), the left-hand side reads

$$\begin{aligned}
 \text{LHS} = \mathcal{A}_L(\chi_n M_n, M_n \delta \chi_n) &= \int_{I_\mu} \int_{\Omega_0} \lambda M_n^2 \text{tr}(\nabla_0 \chi_n) \text{tr}(\nabla_0 \delta \chi_n) dV d\mu \\
 &\quad + \int_{I_\mu} \int_{\Omega_0} 2\mu M_n^2 \nabla_0^s \chi_n : \nabla_0 \delta \chi_n dV d\mu \\
 &= \lambda \int_{I_\mu} M_n^2 d\mu \int_{\Omega_0} \text{tr}(\nabla_0 \chi_n) \text{tr}(\nabla_0 \delta \chi_n) dV \\
 &\quad + 2 \int_{I_\mu} \mu M_n^2 d\mu \int_{\Omega_0} \nabla_0^s \chi_n : \nabla_0 \delta \chi_n dV.
 \end{aligned} \tag{7.36}$$

As indicated by [Equation 7.23](#), there are three terms on the right-hand side. The first term deals with the prescribed loads, it reads

$$\text{RHS}_1 = \mathcal{L}(M_n \delta \chi_n) = \int_{I_\mu} \int_{\Gamma_N} \mathbf{T} \cdot M_n \delta \chi_n dA d\mu = \int_{I_\mu} M_n d\mu \int_{\Gamma_N} \mathbf{T} \cdot \delta \chi_n dA. \tag{7.37}$$

The second term is nonlinear, and it is the most complex term, which reads

$$\begin{aligned}
 \text{RHS}_2 &= \mathcal{A}_{\text{NL}} \left(\sum_{i=1}^{n-1} \boldsymbol{\chi}_i M_i, M_n \delta \boldsymbol{\chi}_n \right) \\
 &= \lambda \int_{I_\mu} \int_{\Omega_0} \left(\sum_{i=1}^{n-1} M_i \text{tr} \nabla_0 \boldsymbol{\chi}_i \right) M_n \left(\sum_{j=1}^{n-1} M_j \nabla_0 \boldsymbol{\chi}_j : \nabla_0 \delta \boldsymbol{\chi}_n \right) dV d\mu \\
 &\quad + 2 \int_{I_\mu} \int_{\Omega_0} \mu \left(\sum_{i=1}^{n-1} M_i \nabla_0 \boldsymbol{\chi}_i \right) \left(\sum_{j=1}^{n-1} M_j \nabla_0^s \boldsymbol{\chi}_j \right) : M_n \nabla_0 \delta \boldsymbol{\chi}_n dV d\mu \\
 &\quad + \frac{\lambda}{2} \int_{I_\mu} \int_{\Omega_0} \text{tr} \left[\left(\sum_{i=1}^{n-1} M_i \nabla_0^T \boldsymbol{\chi}_i \right) \left(\sum_{j=1}^{n-1} M_j \nabla_0 \boldsymbol{\chi}_j \right) \right] \sum_{k=1}^{n-1} M_k \nabla_0 \boldsymbol{\chi}_k : M_n \nabla_0 \delta \boldsymbol{\chi}_n dV d\mu \\
 &\quad + \frac{\lambda}{2} \int_{I_\mu} \int_{\Omega_0} \text{tr} \left[\left(\sum_{i=1}^{n-1} M_i \nabla_0^T \boldsymbol{\chi}_i \right) \left(\sum_{j=1}^{n-1} M_j \nabla_0 \boldsymbol{\chi}_j \right) \right] M_n \text{tr} (\nabla_0 \delta \boldsymbol{\chi}_n) dV d\mu \\
 &\quad + \int_{I_\mu} \int_{\Omega_0} \mu \left(\sum_{i=1}^{n-1} M_i \nabla_0 \boldsymbol{\chi}_i \right) \left(\sum_{j=1}^{n-1} M_j \nabla_0^T \boldsymbol{\chi}_j \right) \left(\sum_{k=1}^{n-1} M_k \nabla_0 \boldsymbol{\chi}_k \right) : M_n \nabla_0 \delta \boldsymbol{\chi}_n dV d\mu \\
 &\quad + \int_{I_\mu} \int_{\Omega_0} \mu \left(\sum_{j=1}^{n-1} M_j \nabla_0^T \boldsymbol{\chi}_j \right) \left(\sum_{k=1}^{n-1} M_k \nabla_0 \boldsymbol{\chi}_k \right) : M_n \text{tr} (\nabla_0 \delta \boldsymbol{\chi}_n) dV d\mu \\
 &= \lambda \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \int_{I_\mu} M_i M_j M_n d\mu \int_{\Omega_0} (\text{tr} \nabla_0 \boldsymbol{\chi}_i) (\nabla_0 \boldsymbol{\chi}_j : \nabla_0 \delta \boldsymbol{\chi}_n) dV \\
 &\quad + 2 \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \int_{I_\mu} \mu M_i M_j M_n d\mu \int_{\Omega_0} (\nabla_0 \boldsymbol{\chi}_i \nabla_0^s \boldsymbol{\chi}_j) : \nabla_0 \delta \boldsymbol{\chi}_n dV \\
 &\quad + \frac{\lambda}{2} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \sum_{k=1}^{n-1} \int_{I_\mu} M_i M_j M_k M_n d\mu \int_{\Omega_0} \text{tr} (\nabla_0^T \boldsymbol{\chi}_i \nabla_0 \boldsymbol{\chi}_j) (\nabla_0 \boldsymbol{\chi}_k : \nabla_0 \delta \boldsymbol{\chi}_n) dV \\
 &\quad + \frac{\lambda}{2} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \int_{I_\mu} M_i M_j M_n d\mu \int_{\Omega_0} \text{tr} (\nabla_0^T \boldsymbol{\chi}_i \nabla_0 \boldsymbol{\chi}_j) \text{tr} (\nabla_0 \delta \boldsymbol{\chi}_n) dV \\
 &\quad + \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \sum_{k=1}^{n-1} \int_{I_\mu} \mu M_i M_j M_k M_n d\mu \int_{\Omega_0} (\nabla_0 \boldsymbol{\chi}_i \nabla_0^T \boldsymbol{\chi}_j \nabla_0 \boldsymbol{\chi}_k) : \nabla_0 \delta \boldsymbol{\chi}_n dV \\
 &\quad + \sum_{j=1}^{n-1} \sum_{k=1}^{n-1} \int_{I_\mu} \mu M_j M_k M_n d\mu \int_{\Omega_0} (\nabla_0^T \boldsymbol{\chi}_j \nabla_0 \boldsymbol{\chi}_k) : \nabla_0 \delta \boldsymbol{\chi}_n dV.
 \end{aligned} \tag{7.38}$$

Now we consider the third term. In fact, it is similar to the left-hand side. Recall

Equation 7.14, and by mimicking Equation 7.36, we have

$$\begin{aligned}
 \text{RHS}_3 &= \mathcal{A}_L \left(\sum_{i=1}^{n-1} \boldsymbol{\chi}_i M_i, M_n \delta \boldsymbol{\chi}_n \right) \\
 &= \int_{I_\mu} \int_{\Omega_0} \lambda \left(\sum_{i=1}^{n-1} M_i \text{tr} \nabla_0 \boldsymbol{\chi}_i \right) M_n \text{tr} \nabla_0 \boldsymbol{\chi}_n dV d\mu \\
 &\quad + \int_{I_\mu} \int_{\Omega_0} 2\mu \left(\sum_{i=1}^{n-1} M_i \nabla_0^s \boldsymbol{\chi}_i \right) : M_n \nabla_0 \delta \boldsymbol{\chi}_n dV d\mu \\
 &= \lambda \sum_{i=1}^{n-1} \int_{I_\mu} M_i M_n d\mu \int_{\Omega_0} \text{tr}(\nabla_0 \boldsymbol{\chi}_i) \text{tr}(\nabla_0 \delta \boldsymbol{\chi}_n) dV \\
 &\quad + 2 \sum_{i=1}^{n-1} \int_{I_\mu} \mu M_i M_n d\mu \int_{\Omega_0} \nabla_0^s \boldsymbol{\chi}_i : \nabla_0 \delta \boldsymbol{\chi}_n dV.
 \end{aligned} \tag{7.39}$$

In total, the right-hand side is calculated as

$$\text{RHS} = \text{RHS}_1 - \text{RHS}_2 - \text{RHS}_3. \tag{7.40}$$

For the parametric problem, the formulation is similar and is omitted here. In the coming discretised formulation, implementation of both mechanical and parametric problem will be detailed.

7.3.2 Discretised formulation for parametric modes

First we introduce the discretised form of the parametric space Ω_μ . Since we discuss the case of only one parameter μ , the parametric space $\Omega_\mu = I_\mu$ is 1D. For this 1D case, any discretisation scheme should apply. We would choose finite element method for its robustness and optimality. The i th parametric mode $M_i(\mu)$ is then approximated by

$$M_i(\mu) \approx \sum_{a=1}^{\text{nnd1}} \tilde{N}_a(\mu) \hat{\omega}_a^i = \mathbf{N}^T \boldsymbol{\omega}_i, \tag{7.41}$$

where nnd1 is number of nodes used for the discretisation, $\mathbf{N} = (\tilde{N}_1, \tilde{N}_2, \dots, \tilde{N}_{\text{nnd1}})^T$ is a column vector composed by nodal shape functions $\tilde{N}_a(\mu)$ which is independent of any PGD mode, and $\boldsymbol{\omega}_i$ is a column vector containing the nodal DOFs $\hat{\omega}_a^i$.

For convenience, we define the following mass-like matrices

$$\begin{aligned}
 \mathbf{M} &:= \int_{I_\mu} \mathbf{N} \mathbf{N}^T d\mu, \\
 \mathbf{H} &:= \int_{I_\mu} \mu \mathbf{N} \mathbf{N}^T d\mu,
 \end{aligned} \tag{7.42}$$

and the following load-like vectors

$$\begin{aligned}
 \mathbf{Q} &:= \int_{I_\mu} \mathbf{N} d\mu, \\
 \mathbf{S}_1^{ij} &:= \int_{I_\mu} \mathbf{N}\mathbf{N}^T \boldsymbol{\omega}_i \boldsymbol{\omega}_j^T \mathbf{N} d\mu, \\
 \mathbf{S}_2^{ij} &:= \int_{I_\mu} \mu \mathbf{N}\mathbf{N}^T \boldsymbol{\omega}_i \boldsymbol{\omega}_j^T \mathbf{N} d\mu, \\
 \mathbf{S}_3^{ijk} &:= \int_{I_\mu} \mathbf{N}\mathbf{N}^T \boldsymbol{\omega}_i \boldsymbol{\omega}_j^T \mathbf{N}\mathbf{N}^T \boldsymbol{\omega}_k d\mu, \\
 \mathbf{S}_4^{ijk} &:= \int_{I_\mu} \mu \mathbf{N}\mathbf{N}^T \boldsymbol{\omega}_i \boldsymbol{\omega}_j^T \mathbf{N}\mathbf{N}^T \boldsymbol{\omega}_k d\mu,
 \end{aligned} \tag{7.43}$$

where \mathbf{M} and \mathbf{H} are constant mass-like matrices, \mathbf{Q} is constant vector while \mathbf{S}_1 , \mathbf{S}_2 , \mathbf{S}_3 , \mathbf{S}_4 are mode-dependent vectors. That is, the \mathbf{S} vectors are different from one PGD mode to another, because the nodal DOF vectors of parametric modes are included. In particular, \mathbf{S}_1 and \mathbf{S}_2 relate to two modes, whilst \mathbf{S}_3 and \mathbf{S}_4 relate to three modes.

Then we may approximate the necessary integrals in following discretised formulas:

$$\begin{aligned}
 \int_{I_\mu} M_n d\mu &\approx \boldsymbol{\omega}_n^T \mathbf{Q}, \\
 \int_{I_\mu} M_n^2 d\mu &\approx \boldsymbol{\omega}_n^T \mathbf{M} \boldsymbol{\omega}_n, \\
 \int_{I_\mu} \mu M_n^2 d\mu &\approx \boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_n, \\
 \int_{I_\mu} M_i M_j M_n d\mu &\approx \boldsymbol{\omega}_n^T \mathbf{S}_1^{ij}, \\
 \int_{I_\mu} \mu M_i M_j M_n d\mu &\approx \boldsymbol{\omega}_n^T \mathbf{S}_2^{ij}, \\
 \int_{I_\mu} M_i M_j M_k M_n d\mu &\approx \boldsymbol{\omega}_n^T \mathbf{S}_3^{ijk}, \\
 \int_{I_\mu} \mu M_i M_j M_k M_n d\mu &\approx \boldsymbol{\omega}_n^T \mathbf{S}_4^{ijk}.
 \end{aligned} \tag{7.44}$$

Typically Gaussian quadrature is used to compute the integrals numerically.

7.3.3 Discretised formulation for spatial modes

We now introduce the discretised form of the physical space in total Lagrange formulation. For convenience, we consider only one element. The i th spatial mode is typically discretised with 3D finite elements

$$\boldsymbol{\chi}_i(\mathbf{X}) \approx \sum_{a=1}^{\text{nnd}} N_a(\mathbf{X}) \hat{\boldsymbol{\varphi}}_a^i, \tag{7.45}$$

where $N_a(\mathbf{X})$ is the shape function and $\hat{\boldsymbol{\varphi}}_a^i$ is the nodal DOF vector. Number of nodes is denoted by **nnd**.

Gradient of the spatial mode with respect to material coordinates is computed as

$$\nabla_0 \boldsymbol{\chi}_i \approx \sum_{a=1}^{\text{nnd}} \hat{\boldsymbol{\varphi}}_a^i \otimes \frac{\partial N_a}{\partial \mathbf{X}}. \quad (7.46)$$

Usually the material coordinates \mathbf{X} is based on a standard isoparametric element, and the shape functions are represented by the parent coordinates $\boldsymbol{\xi} = (\xi, \eta, \zeta)$,

$$\mathbf{X} = \sum_{a=1}^{\text{nnd}} N_a(\boldsymbol{\xi}) \hat{\mathbf{X}}_a, \quad (7.47)$$

where $\hat{\mathbf{X}}_a$ denotes DOF vector in material coordinates.

Using the chain rule, we reach the shape function routine to obtain the gradients

$$\frac{\partial N_a}{\partial \mathbf{X}} = \frac{\partial N_a(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{X}} = \left(\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} \right)^{-T} \frac{\partial N_a(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}, \quad (7.48)$$

where the Jacobian is

$$\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} = \sum_{a=1}^{\text{nnd}} \hat{\mathbf{X}}_a \otimes \frac{\partial N_a(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}. \quad (7.49)$$

Trace of the gradient of i th spatial mode $\boldsymbol{\chi}_i$ is its divergence in material coordinates

$$\text{tr}(\nabla_0 \boldsymbol{\chi}_i) = \text{DIV} \boldsymbol{\chi}_i = \sum_{a=1}^{\text{nnd}} \hat{\boldsymbol{\varphi}}_a^i \cdot \frac{\partial N_a}{\partial \mathbf{X}}. \quad (7.50)$$

We define vectorised formulation following the Voigt convention

$$\mathbf{B}_0 := \left[\frac{\partial N_1}{\partial X_1}, \frac{\partial N_1}{\partial X_2}, \frac{\partial N_1}{\partial X_3}, \frac{\partial N_2}{\partial X_1}, \frac{\partial N_2}{\partial X_2}, \frac{\partial N_2}{\partial X_3}, \dots, \frac{\partial N_{\text{nnd}}}{\partial X_1}, \frac{\partial N_{\text{nnd}}}{\partial X_2}, \frac{\partial N_{\text{nnd}}}{\partial X_3} \right]^T, \quad (7.51)$$

$$\boldsymbol{\varphi}_i := \left[\hat{\boldsymbol{\varphi}}_1^T, \hat{\boldsymbol{\varphi}}_2^T, \dots, \hat{\boldsymbol{\varphi}}_{\text{nnd}}^T \right]_i^T, \quad (7.52)$$

where \mathbf{B}_0 is a vector composed by derivatives of shape functions, and $\boldsymbol{\varphi}_i$ is a rearranged vector of nodal DOFs. Now [Equation 7.50](#) can be written as

$$\text{tr}(\nabla_0 \boldsymbol{\chi}_i) = \mathbf{B}_0 \boldsymbol{\varphi}_i. \quad (7.53)$$

In addition, we define the following \mathbf{B} matrices

$$\mathbf{B}_1 := \begin{bmatrix} \frac{\partial N_1}{\partial X_1} & & \frac{\partial N_2}{\partial X_1} & & \cdots & \frac{\partial N_{\text{nd}}}{\partial X_1} & & & \\ & \frac{\partial N_1}{\partial X_1} & & \frac{\partial N_2}{\partial X_1} & & \cdots & & \frac{\partial N_{\text{nd}}}{\partial X_1} & \\ & & \frac{\partial N_1}{\partial X_1} & & \frac{\partial N_2}{\partial X_1} & \cdots & & & \frac{\partial N_{\text{nd}}}{\partial X_1} \\ \frac{\partial N_1}{\partial X_2} & & \frac{\partial N_2}{\partial X_2} & & \cdots & \frac{\partial N_{\text{nd}}}{\partial X_2} & & & \\ & \frac{\partial N_1}{\partial X_2} & & \frac{\partial N_2}{\partial X_2} & & \cdots & & \frac{\partial N_{\text{nd}}}{\partial X_2} & \\ & & \frac{\partial N_1}{\partial X_2} & & \frac{\partial N_2}{\partial X_2} & \cdots & & & \frac{\partial N_{\text{nd}}}{\partial X_2} \\ \frac{\partial N_1}{\partial X_3} & & \frac{\partial N_2}{\partial X_3} & & \cdots & \frac{\partial N_{\text{nd}}}{\partial X_3} & & & \\ & \frac{\partial N_1}{\partial X_3} & & \frac{\partial N_2}{\partial X_3} & & \cdots & & \frac{\partial N_{\text{nd}}}{\partial X_3} & \\ & & \frac{\partial N_1}{\partial X_3} & & \frac{\partial N_2}{\partial X_3} & \cdots & & & \frac{\partial N_{\text{nd}}}{\partial X_3} \end{bmatrix}, \quad (7.54)$$

$$\mathbf{B}_2 := \begin{bmatrix} \frac{\partial N_1}{\partial X_1} & & \frac{\partial N_2}{\partial X_1} & & \cdots & \frac{\partial N_{\text{nd}}}{\partial X_1} & & & \\ \frac{\partial N_1}{\partial X_2} & & \frac{\partial N_2}{\partial X_2} & & \cdots & \frac{\partial N_{\text{nd}}}{\partial X_2} & & & \\ \frac{\partial N_1}{\partial X_3} & & \frac{\partial N_2}{\partial X_3} & & \cdots & \frac{\partial N_{\text{nd}}}{\partial X_3} & & & \\ & \frac{\partial N_1}{\partial X_1} & & \frac{\partial N_2}{\partial X_1} & & \cdots & & \frac{\partial N_{\text{nd}}}{\partial X_1} & \\ & \frac{\partial N_1}{\partial X_2} & & \frac{\partial N_2}{\partial X_2} & & \cdots & & \frac{\partial N_{\text{nd}}}{\partial X_2} & \\ & \frac{\partial N_1}{\partial X_3} & & \frac{\partial N_2}{\partial X_3} & & \cdots & & \frac{\partial N_{\text{nd}}}{\partial X_3} & \\ & & \frac{\partial N_1}{\partial X_1} & & \frac{\partial N_2}{\partial X_1} & \cdots & & & \frac{\partial N_{\text{nd}}}{\partial X_1} \\ & & \frac{\partial N_1}{\partial X_2} & & \frac{\partial N_2}{\partial X_2} & \cdots & & & \frac{\partial N_{\text{nd}}}{\partial X_2} \\ & & \frac{\partial N_1}{\partial X_3} & & \frac{\partial N_2}{\partial X_3} & \cdots & & & \frac{\partial N_{\text{nd}}}{\partial X_3} \end{bmatrix}, \quad (7.55)$$

$$\mathbf{B}_s := \frac{1}{2}(\mathbf{B}_1 + \mathbf{B}_2), \quad (7.56)$$

and dimension of all these matrices is $9 \times 3\text{nd}$. With the help of the \mathbf{B} matrices, the gradients can be simplified as

$$\nabla_0 \chi_i := \mathbf{B}_1 \varphi_i, \quad (7.57)$$

$$\nabla_0^T \chi_i := \mathbf{B}_2 \varphi_i, \quad (7.58)$$

$$\nabla_0^s \chi_i := \mathbf{B}_s \varphi_i. \quad (7.59)$$

For convenience, we introduce the following stiffness-like matrices \mathbf{L}_1 and \mathbf{L}_2 :

$$\begin{aligned} \mathbf{L}_1 &:= \int_{\Omega_0} \mathbf{B}_0 \mathbf{B}_0^T dV, \\ \mathbf{L}_2 &:= \int_{\Omega_0} \mathbf{B}_1^T \mathbf{B}_s dV, \end{aligned} \quad (7.60)$$

and the following load-like vectors:

$$\begin{aligned}
 \mathbf{R}_1^{ij} &:= \int_{\Omega_0} (\mathbf{B}_0^T \boldsymbol{\varphi}_i) (\mathbf{B}_1^T \mathbf{B}_1 \boldsymbol{\varphi}_j) dV, \\
 \mathbf{R}_2^{ij} &:= \int_{\Omega_0} \mathbf{B}_1^T [(\mathbf{B}_1 \boldsymbol{\varphi}_i) \odot (\mathbf{B}_s \boldsymbol{\varphi}_j)] dV, \\
 \mathbf{R}_3^{ijk} &:= \int_{\Omega_0} (\boldsymbol{\varphi}_i^T \mathbf{B}_1^T \mathbf{B}_1 \boldsymbol{\varphi}_j) (\mathbf{B}_1^T \mathbf{B}_1 \boldsymbol{\varphi}_k) dV, \\
 \mathbf{R}_4^{ij} &:= \int_{\Omega_0} (\boldsymbol{\varphi}_i^T \mathbf{B}_1^T \mathbf{B}_1 \boldsymbol{\varphi}_j) \mathbf{B}_0 dV, \\
 \mathbf{R}_5^{ijk} &:= \int_{\Omega_0} \mathbf{B}_1^T [(\mathbf{B}_1 \boldsymbol{\varphi}_i) \odot (\mathbf{B}_2 \boldsymbol{\varphi}_j) \odot (\mathbf{B}_1 \boldsymbol{\varphi}_k)] dV, \\
 \mathbf{R}_6^{ij} &:= \int_{\Omega_0} \mathbf{B}_1^T [(\mathbf{B}_2 \boldsymbol{\varphi}_i) \odot (\mathbf{B}_1 \boldsymbol{\varphi}_j)] dV,
 \end{aligned} \tag{7.61}$$

where \mathbf{L}_1 and \mathbf{L}_2 are constant matrices, while the \mathbf{R} vectors are mode-dependent. Note that operator \odot means converting Voigt vectors back to matrices, perform matrix multiplication and then transform the product matrix to Voigt vector.

Now we may simplify the necessary integrals in following discretised formulas:

$$\int_{\Omega_0} \text{tr}(\nabla_0 \boldsymbol{\chi}_n) \text{tr}(\nabla_0 \delta \boldsymbol{\chi}_n) dV \approx \delta \boldsymbol{\varphi}_n^T \mathbf{L}_1 \boldsymbol{\varphi}_n, \tag{7.62}$$

$$\int_{\Omega_0} \nabla_0^s \boldsymbol{\chi}_n : \nabla_0 \delta \boldsymbol{\chi}_n dV \approx \delta \boldsymbol{\varphi}_n^T \mathbf{L}_2 \boldsymbol{\varphi}_n, \tag{7.63}$$

$$\int_{\Omega_0} (\text{tr} \nabla_0 \boldsymbol{\chi}_i) (\nabla_0 \boldsymbol{\chi}_j : \nabla_0 \delta \boldsymbol{\chi}_n) dV \approx \delta \boldsymbol{\varphi}_n^T \mathbf{R}_1^{ij}, \tag{7.64}$$

$$\int_{\Omega_0} (\nabla_0 \boldsymbol{\chi}_i \nabla_0^s \boldsymbol{\chi}_j) : \nabla_0 \delta \boldsymbol{\chi}_n dV \approx \delta \boldsymbol{\varphi}_n^T \mathbf{R}_2^{ij}, \tag{7.65}$$

$$\int_{\Omega_0} \text{tr}(\nabla_0^T \boldsymbol{\chi}_i \nabla_0 \boldsymbol{\chi}_j) (\nabla_0 \boldsymbol{\chi}_k : \nabla_0 \delta \boldsymbol{\chi}_n) dV \approx \delta \boldsymbol{\varphi}_n^T \mathbf{R}_3^{ijk}, \tag{7.66}$$

$$\int_{\Omega_0} \text{tr}(\nabla_0^T \boldsymbol{\chi}_i \nabla_0 \boldsymbol{\chi}_j) \text{tr}(\nabla_0 \delta \boldsymbol{\chi}_n) dV \approx \delta \boldsymbol{\varphi}_n^T \mathbf{R}_4^{ij}, \tag{7.67}$$

$$\int_{\Omega_0} (\nabla_0 \boldsymbol{\chi}_i \nabla_0^T \boldsymbol{\chi}_j \nabla_0 \boldsymbol{\chi}_k) : \nabla_0 \delta \boldsymbol{\chi}_n dV \approx \delta \boldsymbol{\varphi}_n^T \mathbf{R}_5^{ijk}, \tag{7.68}$$

$$\int_{\Omega_0} (\nabla_0^T \boldsymbol{\chi}_j \nabla_0 \boldsymbol{\chi}_k) : \nabla_0 \delta \boldsymbol{\chi}_n dV \approx \delta \boldsymbol{\varphi}_n^T \mathbf{R}_6^{ij}. \tag{7.69}$$

7.3.4 Discretised formulation of PGD alternative direction iteration scheme

Now we may rewrite the discretised PGD generalised weak form in the discretised formulation. Again, we take the mechanical problem for example, as the resultant formulation could be modified for the parametric problem.

- Consider Equation 7.36, we have

$$\text{LHS} = \delta \boldsymbol{\varphi}_n^T \left[\lambda (\boldsymbol{\omega}_n^T \mathbf{M} \boldsymbol{\omega}_n) \mathbf{L}_1 + 2 (\boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_n) \mathbf{L}_2 \right] \boldsymbol{\varphi}_n. \quad (7.70)$$

- Consider Equation 7.37, for simplicity, we assume \mathbf{T} is a known constant

$$\text{RHS}_1 = \int_{I_\mu} M_n d\mu \int_{\Gamma_N} \mathbf{T} \cdot \delta \boldsymbol{\chi}_n dA \approx \delta \boldsymbol{\varphi}_n^T \mathbf{F} (\boldsymbol{\omega}_n^T \mathbf{Q}), \quad (7.71)$$

where \mathbf{F} is the well-known nodal force vector.

- Consider Equation 7.39, it is similar to the LHS

$$\text{RHS}_3 \approx \delta \boldsymbol{\varphi}_n^T \sum_{i=1}^{n-1} \left[\lambda (\boldsymbol{\omega}_n^T \mathbf{M} \boldsymbol{\omega}_i) \mathbf{L}_1 + 2 (\boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_i) \mathbf{L}_2 \right] \boldsymbol{\varphi}_i. \quad (7.72)$$

- Consider the large RHS₂ Equation 7.38, it can be written in matrix formulation as

$$\begin{aligned} \text{RHS}_2 = \delta \boldsymbol{\varphi}_n^T \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} & \left[\lambda (\boldsymbol{\omega}_n^T \mathbf{S}_1^{ij}) \mathbf{R}_1^{ij} + 2 (\boldsymbol{\omega}_n^T \mathbf{S}_2^{ij}) \mathbf{R}_2^{ij} + \frac{\lambda}{2} (\boldsymbol{\omega}_n^T \mathbf{S}_1^{ij}) \mathbf{R}_4^{ij} + (\boldsymbol{\omega}_n^T \mathbf{S}_2^{ij}) \mathbf{R}_6^{ij} \right. \\ & \left. + \sum_{k=1}^{n-1} \left(\frac{\lambda}{2} (\boldsymbol{\omega}_n^T \mathbf{S}_3^{ijk}) \mathbf{R}_3^{ijk} + (\boldsymbol{\omega}_n^T \mathbf{S}_4^{ijk}) \mathbf{R}_5^{ijk} \right) \right]. \end{aligned} \quad (7.73)$$

Finally, we obtain the discretised formulation for PGD alternative direction iteration:

1. Mechanical problem: Assuming known $\boldsymbol{\varphi}_i, \boldsymbol{\omega}_i, (i = 1, \dots, n-1)$ and $\boldsymbol{\omega}_n$, solve $\boldsymbol{\varphi}_n$ by LHS = RHS, where

$$\begin{aligned} \text{LHS} &= \left[\lambda (\boldsymbol{\omega}_n^T \mathbf{M} \boldsymbol{\omega}_n) \mathbf{L}_1 + 2 (\boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_n) \mathbf{L}_2 \right] \boldsymbol{\varphi}_n, \\ \text{RHS} &= \mathbf{F} (\boldsymbol{\omega}_n^T \mathbf{Q}) - \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \left[\lambda (\boldsymbol{\omega}_n^T \mathbf{S}_1^{ij}) \mathbf{R}_1^{ij} + 2 (\boldsymbol{\omega}_n^T \mathbf{S}_2^{ij}) \mathbf{R}_2^{ij} + \frac{\lambda}{2} (\boldsymbol{\omega}_n^T \mathbf{S}_1^{ij}) \mathbf{R}_4^{ij} \right. \\ & \quad \left. + (\boldsymbol{\omega}_n^T \mathbf{S}_2^{ij}) \mathbf{R}_6^{ij} + \sum_{k=1}^{n-1} \left(\frac{\lambda}{2} (\boldsymbol{\omega}_n^T \mathbf{S}_3^{ijk}) \mathbf{R}_3^{ijk} + (\boldsymbol{\omega}_n^T \mathbf{S}_4^{ijk}) \mathbf{R}_5^{ijk} \right) \right] \\ & \quad - \sum_{i=1}^{n-1} \left[\lambda (\boldsymbol{\omega}_n^T \mathbf{M} \boldsymbol{\omega}_i) \mathbf{L}_1 + 2 (\boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_i) \mathbf{L}_2 \right] \boldsymbol{\varphi}_i; \end{aligned} \quad (7.74)$$

2. Parametric problem: Assuming known $\boldsymbol{\varphi}_i, \boldsymbol{\omega}_i, (i = 1, \dots, n - 1)$ and $\boldsymbol{\varphi}_n$, solve $\boldsymbol{\omega}_n$ by LHS = RHS, where

$$\begin{aligned}
\text{LHS} &= \left[\lambda(\boldsymbol{\varphi}_n^T \mathbf{L}_1 \boldsymbol{\varphi}_n) \mathbf{M} + 2(\boldsymbol{\varphi}_n^T \mathbf{L}_2 \boldsymbol{\varphi}_n) \mathbf{H} \right] \boldsymbol{\omega}_n, \\
\text{RHS} &= \mathbf{Q}(\boldsymbol{\varphi}_n^T \mathbf{F}) - \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \left[\lambda(\boldsymbol{\varphi}_n^T \mathbf{R}_1^{ij}) \mathbf{S}_1^{ij} + 2(\boldsymbol{\varphi}_n^T \mathbf{R}_2^{ij}) \mathbf{S}_2^{ij} + \frac{\lambda}{2} (\boldsymbol{\varphi}_n^T \mathbf{R}_4^{ij}) \mathbf{S}_1^{ij} \right. \\
&\quad \left. + (\boldsymbol{\varphi}_n^T \mathbf{R}_6^{ij}) \mathbf{S}_2^{ij} + \sum_{k=1}^{n-1} \left(\frac{\lambda}{2} (\boldsymbol{\varphi}_n^T \mathbf{R}_3^{ijk}) \mathbf{S}_3^{ijk} + (\boldsymbol{\varphi}_n^T \mathbf{R}_5^{ijk}) \mathbf{S}_4^{ijk} \right) \right] \\
&\quad - \sum_{i=1}^{n-1} \left[\lambda(\boldsymbol{\varphi}_n^T \mathbf{L}_1 \boldsymbol{\varphi}_i) \mathbf{M} + 2(\boldsymbol{\varphi}_n^T \mathbf{L}_2 \boldsymbol{\varphi}_i) \mathbf{H} \right] \boldsymbol{\omega}_i.
\end{aligned} \tag{7.75}$$

Remark 7.3 (On the monolithic solver). *Comparing the finally obtained formulations and the conventional PGD alternative direction iteration scheme, it can be seen that the Picard linearisation works quite implicitly on the right-hand side. It implies that the proposed scheme can be implemented based on a conventional PGD code after adding to the right-hand side a linearised term, achieving a monolithic PGD solver with Picard linearisation.*

Remark 7.4 (On the mode-dependent vectors). *For both parametric and mechanical problems, there are mode-dependent vectors (the \mathbf{S} and \mathbf{R} vectors) which occur only in the second term on the right-hand side of each problem. As we already discussed, the second right-hand side term comes from Picard linearisation of the nonlinear problem. It is interesting to point out that this linearisation essentially turns the nonlinearity of the problem into interactions of PGD modes.*

Remark 7.5 (On the efficiency of practical implementation). *It is obvious that the computational cost of proposed scheme would grow exponentially with the number of required modes, because in the second right-hand side term the sums include mode-dependent vectors which have to be updated during the iterations. However, the vectors depend only on modes that should have been obtained in previous modal enrichment, which enables a more efficient parallel implementation in the code.*

7.4 Numerical example

In this section, a numerical example is presented with implementation of the Picard linearised PGD framework. To simplify the problem, only one material parameter is

used as the extra coordinate.

7.4.1 Material properties

The range of Young Modulus is assumed to be $E \in [10, 100]$ MPa. If we fix the Poisson's ratio $\nu = 0.49$, the possible range of the Lamé constants λ and μ can be obtained as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \in [164.43, 1644.3] \text{ MPa},$$
$$\mu = \frac{E}{2(1+\nu)} \in [3.3557, 33.557] \text{ MPa}.$$

Since St.Venant-Kirchhoff constitutive model is used, to adapt the problem with only one variable material parameter, we fix $\lambda = 400$ MPa and specify range $\mu \in [4, 40]$ MPa. For reference, in fact, we have set:

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \in [11.96, 116.36] \text{ MPa},$$
$$\nu = \frac{\lambda}{2(\lambda + \mu)} \in [0.454, 0.495].$$

7.4.2 Mechanical model

The classical mechanical model, plate with a round open hole in the centre with unidirectional tension applied, is again used for this example. However, here the plate is square, 3D hexahedral mesh is used, and only one eighth of the plate is modelled with proper symmetric boundary conditions. The FE mesh is illustrated in [Figure 7.2](#).

Symmetric boundary conditions have been applied, and nodal tension forces are applied on nodes lying on the right end in x direction. The total load magnitude is 24 N, which is uniformly distributed on the right end.

7.4.3 SVD analysis

To assess the parametric dependency of the model, we use standard nonlinear FE methods to compute a series of results $\mathbf{U}(\mu_i)$ with specified samples of parameters μ_i , and then perform SVD on the matrix composed by obtained results of nodal displacement DOF vectors. From [Figure 7.3](#) it can be seen that norm of the displacement

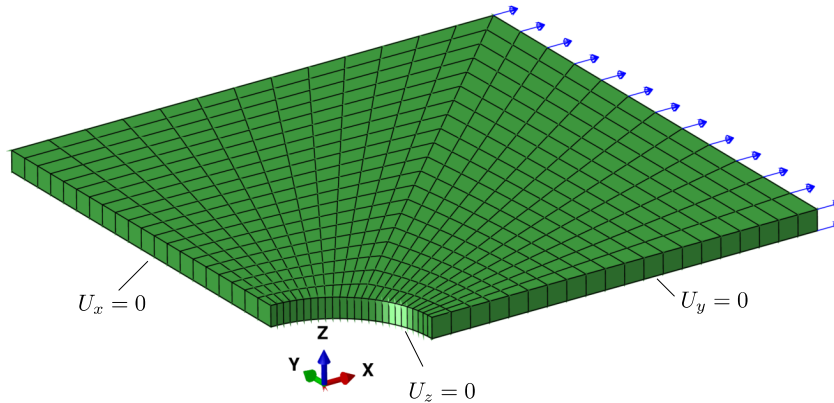


Figure 7.2: Illustration of the model (1/8 model due to symmetry). Edge length both 50 mm, hole radius 10 mm, model thickness 2 mm.

DOF \mathbf{U} is nonlinearly dependent on the shear modulus μ . The FE results forms a matrix $[\mathbf{U}(\mu_i)]$, where each column corresponds a DOF vector computed with specified shear modulus μ_i . The singular values for this matrix is plotted in [Figure 7.4](#), from which we can see the first 10 singular values drop significantly as much as 10^{10} times. This indicates that 10 PGD modes is accurate enough to approximate the high-fidelity results.

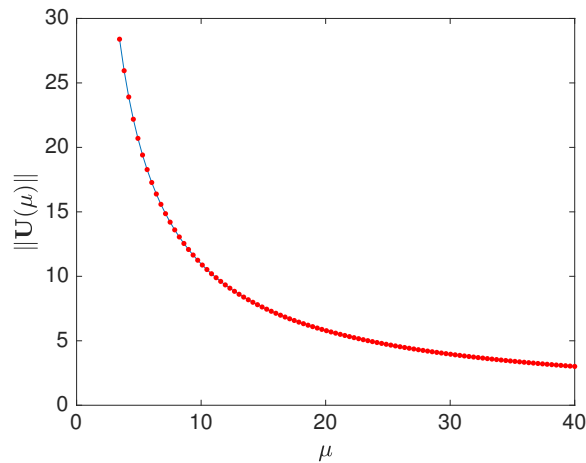


Figure 7.3: Dependency of displacement norm $\|\mathbf{U}\|$ on parameter μ .

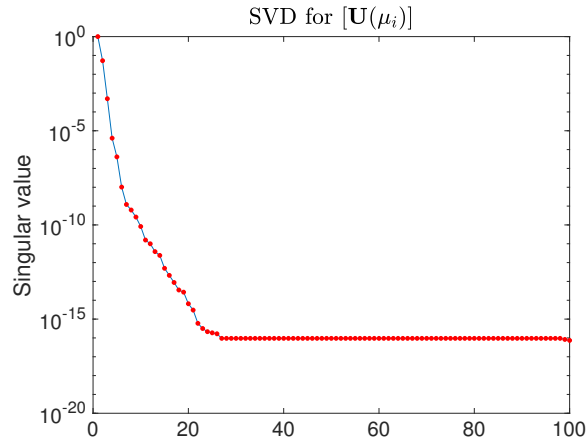


Figure 7.4: Singular value spectrum for each parameter.

7.4.4 PGD results

We compute the global nodal displacement vector using the proposed Picard linearised PGD scheme, which reads:

$$\mathbf{U}_{\text{PGD}}^n(\mu) = \sum_{i=1}^n M_i(\mu) \boldsymbol{\varphi}_i = \sum_{i=1}^n \boldsymbol{\omega}_i^T \mathbf{N}(\mu) \boldsymbol{\varphi}_i, \quad (7.76)$$

where $\boldsymbol{\varphi}_i$ and $\boldsymbol{\omega}_i$ denote the spatial and parametric DOF vectors respectively. As it is discussed, 10 PGD modes is computed to compose the PGD vademecum. [Figure 7.5](#) shows the amplitudes and computational costs. The computational costs, as discussed before, grow quasi-exponentially. However, with this price paid, the acceleration in the online phase is fascinating, as shown in [Table 7.1](#). The offline computation for the PGD vademecum is performed with Matlab R2012a on a workstation (CPU: 4×Intel Xeon E7540 2.00 GHz, 24 cores in total, 64 GB RAM), whilst the online computation and reference FE solution are performed with Matlab R2017b on a desktop PC (CPU: Intel Core i5-3470 3.20 GHz, 4 cores in total, 16 GB RAM). No explicit parallelisation has been used.

Table 7.1: Acceleration of PGD in a typical online computation

PGD CPU time	FE CPU time	Acceleration ratio
Offline 13.0 h	—	—
Online 0.001024 s	5.19553 s	4784.1

Since PGD vademecum has enabled extremely fast computation of the displacement DOFs, it is possible to perform a post-compression on the approximated results

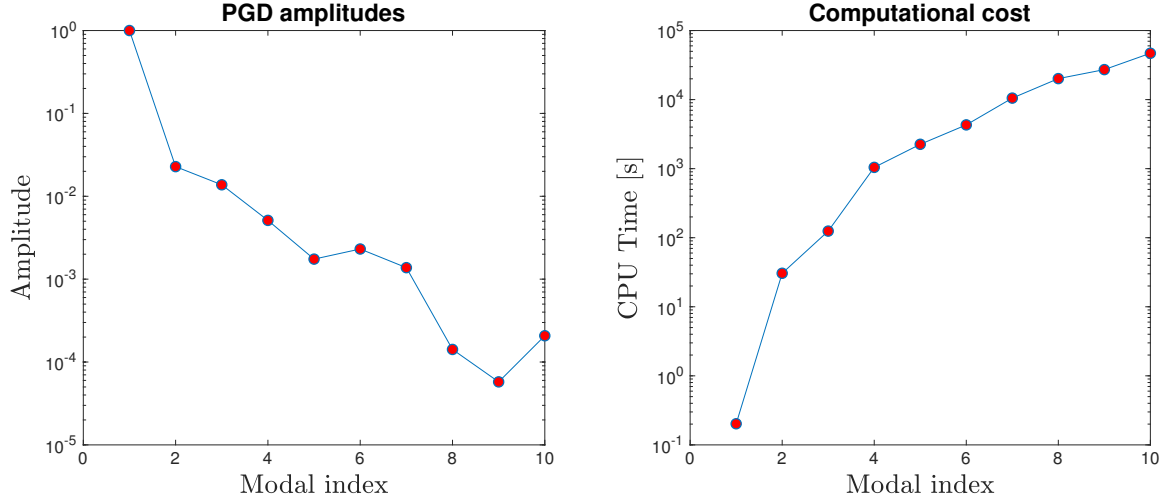


Figure 7.5: Amplitude of the mode $\|\varphi_i\| \cdot \|\omega_i\|$ decreases with modal index i , while computational cost increases when more modes are computed.

using again the SVD technique. That is, perform SVD for $[\mathbf{U}_{\text{PGD}}(\mu_i)]$ instead of on $[\mathbf{U}(\mu_i)]$. It is interesting to compare the normalised amplitudes between PGD modes and previously computed singular values, as shown in Figure 7.6. It can be seen that from the convergence rate point of view, PGD results is less optimal than SVD results. However, after the post-compression, PGD results could approximate the SVD results.

The computed 10 parametric modes are plotted in Figure 7.7, and the spatial modes are illustrated in Figure 7.8.

To assess the accuracy, the relative errors are computed on selected parameter values, using FE results as the reference. The relative errors are computed as $\|\mathbf{U}_{\text{PGD}} - \mathbf{U}_{\text{FE}}\| / \|\mathbf{U}_{\text{FE}}\|$ and listed in Table 7.2.

Table 7.2: Relative error for selected parameters

No.	μ [MPa]	Relative error
1	4.0	0.079%
2	20.0	0.017%
3	40.0	0.031%

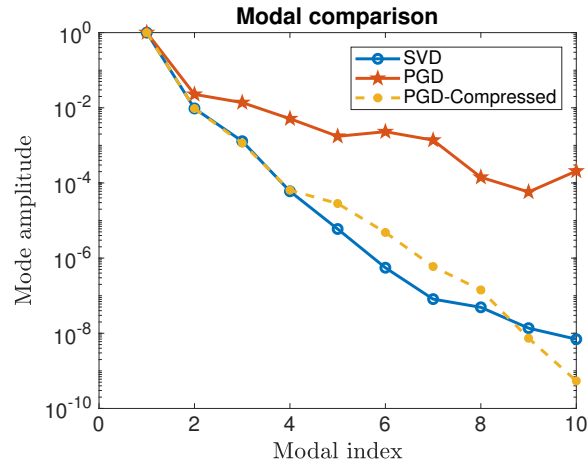


Figure 7.6: Compare between SVD singular values and PGD amplitudes. It shows the less optimality of PGD before compression.

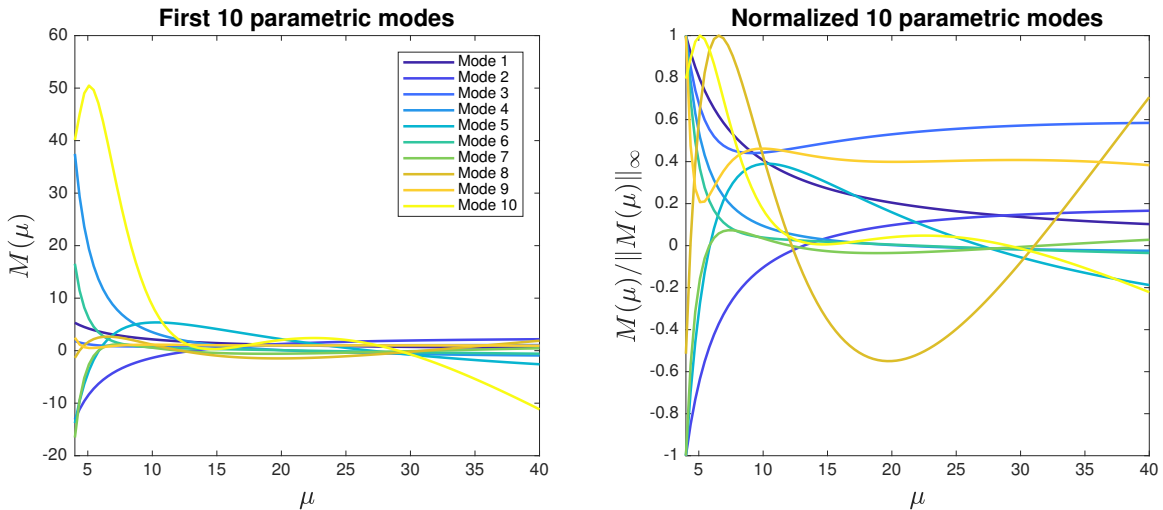


Figure 7.7: First 10 PGD parametric modes.

7.5 Summary

As a preliminary step towards applying a consistent PGD framework in nonlinear problem, we have introduced a Picard linearised formulation under Lagrange description of large deformation solid mechanics. The key is adding a nonlinear term at the right-hand side of the PGD generalised weak form, which could increase the computational cost to the offline phase. However, considerable acceleration during the online phase is obtained, enabling extremely fast simulations for this nonlinear problem up to about 1 kHz.

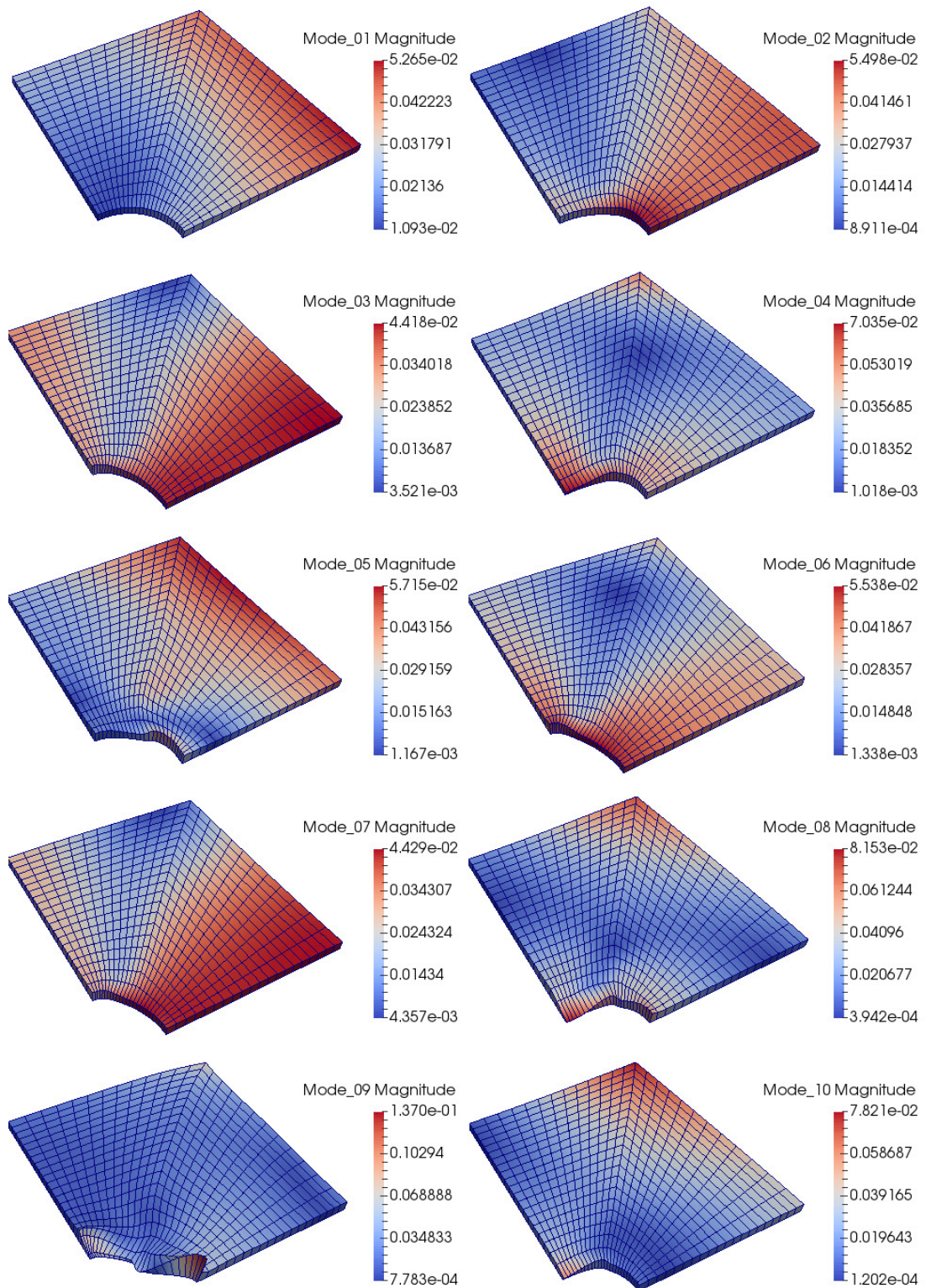


Figure 7.8: First 10 PGD spatial modes.

Chapter 8

Conclusions and future work

This thesis has been dedicated to the exploration of simulation tools/methods for biomechanical applications with the intention of applying reduced order modelling techniques. The multidisciplinary research involves biomechanics, medical image processing, numerical methods for PDEs, reduced order modelling techniques, non-intrusive coupling of computer codes, and even machine learning algorithms from data science. In this final chapter we are about to make some conclusive remarks on the presented work, and try to make some reasonable predictions on the possible future works.

8.1 Conclusive remarks

For the foundation of a biomechanical application, in [Chapter 2](#) we have established an FE model of human proximal femur based on CT images. Previously, *in vitro* experimental tests have been performed, providing data for the validation of the FE model. The model is capable of considering both homogeneous and inhomogeneous material property distributions under isotropic assumption and simple calibration. It is demonstrated that the strain results extracted from both experiment and simulation on same locations match well. The validated FE model with inhomogeneous material properties is further used for PGD application in [Chapter 5](#).

A non-intrusive PGD scheme is proposed in [Chapter 3](#). We used in-house developed Matlab code to control the two-loop alternative direction iteration flow, and

called Abaqus as an external FE solver for each fictitious mechanical problem. It is known that commercial FE packages, such as Abaqus, is widely used in industry. This scheme is able to act as a bridge connecting academic research products, such as PGD, to industrial applications. FE discretisation is used for both spatial model and the parameter space, considering material properties or loading locations as extra coordinates. Numerical benchmark example showed that this non-intrusive PGD scheme is working flawlessly under the Matlab-Abaqus implementation.

Using the FE model built in [Chapter 2](#), it is observed that due to some physical or geometric limitations, the parameter space could be subjected to certain constraints. Those constraints could cause conventional PGD separation of variables fail for the parameters, because they are no more Cartesian. To cure this problem, [Chapter 4](#) proposed a collective strategy to separate the parametric functions in selectively, that is, instead of separating the full parameter space into independent 1D subspaces, we keep the most correlated parameters unseparated in 2D/3D subspaces, and solve the parametric modes in a parameter space in 2D/3D. This strategy is demonstrated by numerical examples and it is shown that the convergence rate is improved, which could possibly compensate the increased computational cost. It is also noted that the non-intrusive scheme is also applicable for the parametric problem, leading to a fully non-intrusive PGD implementation.

To take advantage of the proposed tools into the real biomechanical problem, we used the non-intrusive PGD scheme plus the collective separation strategy for constrained parameter space to investigate the simulation of the femur mechanics in [Chapter 5](#). As a typical biomechanical problem, the material properties is unknown, and it is solved from an identification problem utilising the PGD vademecum. In addition, the exact loading location on the hip joint is also unknown. A further PGD vademecum is generated with variable loading location, which has the ability of providing real-time simulation of mechanical response of the femur under arbitrary loads.

It is well-known that NURBS shape functions, such as the B-splines, are able to provide exact descriptions of smooth geometries and high-order continuities. For many practical applications, such as parameter identification problems, high-order continuity improves the parameter sensitivity of the quantity of interest. Having studied in [Chapter 6](#), we demonstrated this using a classical example for 2D orthotropic materials with four constitutive parameters. Applying the collective separation strat-

egy proposed in [Chapter 4](#), we separated the parameter space into a 3D subspace and a 1D space, both of which are discretised with B-spline shape functions. We used k -refinement for high-order discretisations, inspired by the IGA methodologies. As a result, significant improvements of the sensitivities have been observed.

To further explore the capability of the PGD framework, in [Chapter 7](#) we have developed a consistent PGD framework for nonlinear solid mechanics involving the large strain theory. The nonlinearity in the abstract form prohibits the conventional alternative direction iteration. To solve this problem, we separated the nonlinear form on the left-hand side into a linear part and a nonlinear part, and then used Picard linearisation for the nonlinear part. Consequently, a new nonlinear term is added to the right-hand side, increasing the computational cost considerably. The increased cost is because the computational complexity caused by nonlinearity is transformed into interaction terms of known PGD modes. The more modes are obtained, the higher the computational cost will be. Therefore, the offline phase could be accelerated using parallelisation techniques. Finally, the framework is demonstrated with a classical example, using St.Venant-Kirchhoff constitutive model.

8.2 Future works

All the findings and methods presented in this thesis forms a foundation for further research developments, especially for projects towards more complex applications in biomechanical problems. In this final section, we discuss some possible directions for future works that could be carried out.

Although the PGD-based complete patient-specific real-time simulation framework for the femur has been established, a final product is to be created for practical clinical applications. This product could be a software package, highlighting the on-line phase of the reduced order modelling. To meet its fast-response and multi-query requirements, proper computational visualisation libraries such as `OpenGL` and `VTK`, are necessary for the implementation. Modern simulation Apps, similar to those in literature [Quesada et al. \[2016b\]](#), [Aguado et al. \[2017\]](#), could be developed on different platforms of portable devices such as iOS and Android.

The non-intrusive PGD scheme could be implemented using other high-efficiency programming languages such as C/C++ and FORTRAN. Alternative to Abaqus, other commercial FEM software, such as ANSYS (Ansys Inc., USA), NASTRAN

(MSC software, USA), are also able to be used as the external solver.

Considering the NURBS shape functions, there are many techniques from the IGA community could improve the efficiency of the discretisations of parameter spaces. For instance, in current implementations, standard Gaussian quadrature is used for the numerical integration of necessary mass-like matrices, and this could be very expensive when high-order B-splines are used. More efficient quadrature rules, such as Gauss-Lobatto, could be implemented [Hughes et al. \[2010\]](#). In addition, the spatial discretisation can also use NURBS in the standard IGA approach when necessary. Further techniques, such collocation method [Auricchio et al. \[2010\]](#) are likely to improve the solution of spatial/mechanical problems.

Geometric parametrisation for the shape of biological tissues is a well-known open question. Attempts have been made in the literature such as [Rozza et al. \[2013\]](#), [Iapichino et al. \[2016\]](#) with RB and [Ammar et al. \[2014\]](#), [Zlotnik et al. \[2015b\]](#), [González et al. \[2018\]](#) with PGD. The dimensionality of complex geometry can be reduced using NURBS with high-accuracy description as presented in [Al Akhras et al. \[2017\]](#). It is believed combining NURBS description with PGD could facilitate the reduction of the model, and IGA would be naturally introduced for the spatial problem.

To further investigate PGD application on nonlinear problems, the Picard scheme could be improved by using parallelisation for the offline computation. Although Lagrange description in large strain theory has been used, current implementation is limited to St.Venant-Kirchhoff constitutive model thanks to its simplicity. Since other hyperelastic material models, such as Neo-Hookean and Mooney-Rivlin, are more commonly used in biomechanics, it is necessary to generate the nonlinear PGD framework to those cases.

Bibliography

- J. V. Aguado, D. Borzacchiello, C. Ghnatios, F. Lebel, R. Upadhyay, C. Binetruy, and F. Chinesta. A Simulation App based on reduced order modeling for manufacturing optimization of composite outlet guide vanes. *Advanced Modeling and Simulation in Engineering Sciences*, 4(1):1, Dec. 2017. ISSN 2213-7467. doi:[10.1186/s40323-017-0087-y](https://doi.org/10.1186/s40323-017-0087-y).
- M. Agus, A. Giachetti, E. Gobetti, G. Zanetti, and A. Zorcolo. Real-Time Haptic and Visual Simulation of Bone Dissection. *Presence: Teleoperators and Virtual Environments*, 12(1):110–122, Feb. 2003. ISSN 1054-7460. doi:[10.1162/105474603763835378](https://doi.org/10.1162/105474603763835378).
- M. Ainsworth and J. T. Oden. *A posteriori error estimation in finite element analysis*. Wiley, 2000. ISBN 9780471294115. URL <https://www.wiley.com/en-gb/A+Posteriori+Error+Estimation+in+Finite+Element+Analysis-p-9780471294115>.
- H. Al Akhras, T. Elguedj, A. Gravouil, and M. Rochette. Towards an automatic isogeometric analysis suitable trivariate models generation—Application to geometric parametric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:623–645, 2017. ISSN 00457825. doi:[10.1016/j.cma.2016.09.030](https://doi.org/10.1016/j.cma.2016.09.030). URL <http://www.sciencedirect.com/science/article/pii/S0045782516312221>.
- I. Alfaro, D. González, S. Zlotnik, P. Díez, E. Cueto, and F. Chinesta. An error estimator for real-time simulators based on model order reduction. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):30, Nov. 2015. ISSN 2213-7467. doi:[10.1186/s40323-015-0050-8](https://doi.org/10.1186/s40323-015-0050-8). URL <http://www.amses-journal.com/content/2/1/30>.
- P.-E. Allier, L. Chamoin, and P. Ladevèze. Proper Generalized Decomposition computational methods on a benchmark problem: introducing a new strategy based on Constitutive Relation Error minimization. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):17, July 2015. ISSN 2213-7467. doi:[10.1186/s40323-015-0038-4](https://doi.org/10.1186/s40323-015-0038-4).

- A. Ammar. The proper generalized decomposition: A powerful tool for model reduction. *International Journal of Material Forming*, 3:89–102, 2010. ISSN 19606206. doi:[10.1007/s12289-009-0647-x](https://doi.org/10.1007/s12289-009-0647-x).
- A. Ammar, F. Chinesta, P. Diez, and A. Huerta. An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering*, 199(25-28):1872–1880, 2010. ISSN 00457825. doi:[10.1016/j.cma.2010.02.012](https://doi.org/10.1016/j.cma.2010.02.012). URL <http://www.sciencedirect.com/science/article/pii/S0045782510000708>.
- A. Ammar, A. Huerta, F. Chinesta, E. Cueto, and A. Leygue. Parametric solutions involving geometry: A step towards efficient shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268:178–193, Jan. 2014. ISSN 00457825. doi:[10.1016/j.cma.2013.09.003](https://doi.org/10.1016/j.cma.2013.09.003). URL <http://www.sciencedirect.com/science/article/pii/S0045782513002284>.
- L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, and D. a. Steinman. An image-based modeling framework for patient-specific computational hemodynamics. *Medical and Biological Engineering and Computing*, 46(11):1097–1112, Nov. 2008. ISSN 01400118. doi:[10.1007/s11517-008-0420-1](https://doi.org/10.1007/s11517-008-0420-1). URL <http://www.ncbi.nlm.nih.gov/pubmed/19002516>.
- F. Auricchio, L. B. 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, Nov. 2010. ISSN 0218-2025. doi:[10.1142/S0218202510004878](https://doi.org/10.1142/S0218202510004878).
- I. Babuška and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615, Jan. 1978. ISSN 0029-5981. doi:[10.1002/nme.1620121010](https://doi.org/10.1002/nme.1620121010).
- J. Berger, N. Mendes, S. Guernouti, M. Woloszyn, and F. Chinesta. Review of Reduced Order Models for Heat and Moisture Transfer in Building Physics with Emphasis in PGD Approaches. *Archives of Computational Methods in Engineering*, 24(3):655–667, July 2017. ISSN 1134-3060. doi:[10.1007/s11831-016-9184-1](https://doi.org/10.1007/s11831-016-9184-1).
- B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, and A. Poitou. Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity. *Computer Methods in Applied Mechanics and Engineering*, 201-204:1–12, Jan. 2012. ISSN 00457825. doi:[10.1016/j.cma.2011.08.025](https://doi.org/10.1016/j.cma.2011.08.025). URL <http://www.sciencedirect.com/science/article/pii/S0045782511002891>.
- J. Bonet and R. C. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, Cambridge, 2nd edition, 2008. ISBN 978-0521838702. URL <http://www.cambridge.org/catalogue/catalogue.asp?isbn=0511389132>.

- M. Bro-Nielsen and S. Cotin. Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation. *Computer Graphics Forum*, 15(3):57–66, Aug. 1996. ISSN 0167-7055. doi:[10.1111/1467-8659.1530057](https://doi.org/10.1111/1467-8659.1530057).
- L. Chamoin and P. Díez, editors. *Verifying Calculations - Forty Years On*. Springer-Briefs in Applied Sciences and Technology. Springer International Publishing, Cham, 2016. ISBN 978-3-319-20552-6. doi:[10.1007/978-3-319-20553-3](https://doi.org/10.1007/978-3-319-20553-3).
- L. Chamoin, F. Pled, P.-E. Allier, and P. Ladevèze. A posteriori error estimation and adaptive strategy for PGD model reduction applied to parametrized linear parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 327:118–146, Dec. 2017. ISSN 00457825. doi:[10.1016/j.cma.2017.08.047](https://doi.org/10.1016/j.cma.2017.08.047). URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782517306357>.
- F. Chinesta and E. Cueto. *PGD-Based Modeling of Materials, Structures and Processes*. ESAFORM Bookseries on Material Forming. Springer International Publishing, Cham, 2014. ISBN 978-3-319-06181-8. doi:[10.1007/978-3-319-06182-5](https://doi.org/10.1007/978-3-319-06182-5).
- F. Chinesta and P. Ladevèze, editors. *Separated Representations and PGD-Based Model Reduction*, volume 554 of *CISM International Centre for Mechanical Sciences*. Springer Vienna, Vienna, 2014. ISBN 978-3-7091-1793-4. doi:[10.1007/978-3-7091-1794-1](https://doi.org/10.1007/978-3-7091-1794-1).
- F. Chinesta, A. Ammar, and E. Cueto. Recent Advances and New Challenges in the Use of the Proper Generalized Decomposition for Solving Multidimensional Models. *Archives of Computational Methods in Engineering*, 17(4):327–350, 2010. ISSN 11343060. doi:[10.1007/s11831-010-9049-y](https://doi.org/10.1007/s11831-010-9049-y).
- F. Chinesta, A. Ammar, A. Leygue, and R. Keunings. An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11):578–592, June 2011a. ISSN 03770257. doi:[10.1016/j.jnnfm.2010.12.012](https://doi.org/10.1016/j.jnnfm.2010.12.012). URL <http://www.sciencedirect.com/science/article/pii/S0377025711000061>.
- F. Chinesta, P. Ladevèze, and E. Cueto. A Short Review on Model Order Reduction Based on Proper Generalized Decomposition. *Archives of Computational Methods in Engineering*, 18(4):395–404, Oct. 2011b. ISSN 1134-3060. doi:[10.1007/s11831-011-9064-7](https://doi.org/10.1007/s11831-011-9064-7).
- F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, and A. Huerta. PGD-Based Computational Vademecum for Efficient Design, Optimization and Control. *Archives of Computational Methods in Engineering*, 20(1):31–59, 2013. ISSN 11343060. doi:[10.1007/s11831-013-9080-x](https://doi.org/10.1007/s11831-013-9080-x).

- F. Chinesta, R. Keunings, and A. Leygue. *The Proper Generalized Decomposition for Advanced Numerical Simulations*. SpringerBriefs in Applied Sciences and Technology. Springer International Publishing, Cham, 2014. ISBN 978-3-319-02864-4. doi:[10.1007/978-3-319-02865-1](https://doi.org/10.1007/978-3-319-02865-1).
- T. Colin, A. Iollo, D. Lombardi, and O. Saut. System Identification in Tumor Growth Modeling Using Semi-empirical Eigenfunctions. *Mathematical Models and Methods in Applied Sciences*, 22(06):1250003, June 2012. ISSN 0218-2025. doi:[10.1142/S0218202512500030](https://doi.org/10.1142/S0218202512500030).
- A. Cong, J. O. D. Buijs, and D. Dragomir-Daescu. In situ parameter identification of optimal density-elastic modulus relationships in subject-specific finite element models of the proximal femur. *Medical Engineering and Physics*, 33(2):164–173, Mar. 2011. ISSN 13504533. doi:[10.1016/j.medengphy.2010.09.018](https://doi.org/10.1016/j.medengphy.2010.09.018). URL <http://www.sciencedirect.com/science/article/pii/S1350453310002146>.
- S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics*, 5:62–73, 1999. ISSN 10772626. doi:[10.1109/2945.764872](https://doi.org/10.1109/2945.764872).
- A. Courard, D. Néron, P. Ladevèze, and L. Ballere. Integration of PGD-virtual charts into an engineering design process. *Computational Mechanics*, 57(4):637–651, Dec. 2015. ISSN 0178-7675. doi:[10.1007/s00466-015-1246-y](https://doi.org/10.1007/s00466-015-1246-y).
- T. F. Cox and M. A. A. Cox. *Multidimensional scaling*. Chapman & Hall/CRC, 2001. ISBN 1420036122.
- E. Cueto, D. González, and I. Alfaro. *Proper Generalized Decompositions*. SpringerBriefs in Applied Sciences and Technology. Springer International Publishing, Cham, 2016. ISBN 978-3-319-29993-8. doi:[10.1007/978-3-319-29994-5](https://doi.org/10.1007/978-3-319-29994-5).
- E. Dall’Ara, B. Luisier, R. Schmidt, F. Kainberger, P. Zysset, and D. Pahr. A nonlinear QCT-based finite element model validation study for the human femur tested in two configurations in vitro. *Bone*, 52(1):27–38, Jan. 2013. ISSN 87563282. doi:[10.1016/j.bone.2012.09.006](https://doi.org/10.1016/j.bone.2012.09.006). URL <http://www.sciencedirect.com/science/article/pii/S8756328212012306>.
- C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, July 1972. ISSN 0021-9045. doi:[10.1016/0021-9045\(72\)90080-9](https://doi.org/10.1016/0021-9045(72)90080-9). URL <https://www.sciencedirect.com/science/article/pii/0021904572900809>.
- C. de Boor. Package for Calculating with B-Splines. *SIAM Journal on Numerical Analysis*, 14(3):441–472, June 1977. ISSN 0036-1429. doi:[10.1137/0714026](https://doi.org/10.1137/0714026).
- P. Demartines and J. Herault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural*

-
- Networks*, 8(1):148–154, 1997. ISSN 10459227. doi:[10.1109/72.554199](https://doi.org/10.1109/72.554199). URL <http://ieeexplore.ieee.org/document/554199/>.
- D. Devaud and G. Rozza. Certified Reduced Basis Method for Affinely Parametric Isogeometric Analysis NURBS Approximation. In H. J. Bittencourt M., Dumont N., editor, *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pages 41–62. Springer Cham, 2017. ISBN 978-3-319-65870-4. doi:[10.1007/978-3-319-65870-4_3](https://doi.org/10.1007/978-3-319-65870-4_3).
- P. Díez, S. Zlotnik, and A. Huerta. Generalized parametric solutions in Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 326:223–240, Nov. 2017. ISSN 00457825. doi:[10.1016/j.cma.2017.07.016](https://doi.org/10.1016/j.cma.2017.07.016). URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782516316309>.
- M. Duval, J.-C. Passieux, M. Salaün, and S. Guinard. Non-intrusive Coupling: Recent Advances and Scalable Nonlinear Domain Decomposition. *Archives of Computational Methods in Engineering*, 23(1):17–38, Mar. 2016. ISSN 1886-1784. doi:[10.1007/s11831-014-9132-x](https://doi.org/10.1007/s11831-014-9132-x).
- A. Ern, S. Perotto, and A. Veneziani. Hierarchical Model Reduction for Advection-Diffusion-Reaction Problems. In *Numerical Mathematics and Advanced Applications*, pages 703–710. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi:[10.1007/978-3-540-69777-0_84](https://doi.org/10.1007/978-3-540-69777-0_84).
- Y.-C. Fung. *Biomechanics*. Springer New York, New York, NY, 1993. ISBN 978-1-4419-3104-7. doi:[10.1007/978-1-4757-2257-4](https://doi.org/10.1007/978-1-4757-2257-4).
- R. García-Blanco, P. Díez, D. Borzacchiello, and F. Chinesta. Algebraic and Parametric Solvers for the Power Flow Problem: Towards Real-Time and Accuracy-Guaranteed Simulation of Electric Systems. *Archives of Computational Methods in Engineering*, pages 1–24, Apr. 2017. ISSN 1134-3060. doi:[10.1007/s11831-017-9223-6](https://doi.org/10.1007/s11831-017-9223-6).
- E. Giner, B. Bognet, J. J. Ródenas, A. Leygue, F. J. Fuenmayor, and F. Chinesta. The Proper Generalized Decomposition (PGD) as a numerical procedure to solve 3D cracked plates in linear elastic fracture mechanics. *International Journal of Solids and Structures*, 50(10):1710–1720, May 2013. ISSN 00207683. doi:[10.1016/j.ijsolstr.2013.01.039](https://doi.org/10.1016/j.ijsolstr.2013.01.039). URL <http://www.sciencedirect.com/science/article/pii/S0020768313000541>.
- D. González, E. Cueto, F. Chinesta, P. Díez, and A. Huerta. Streamline upwind/Petrov–Galerkin-based stabilization of proper generalized decompositions for high-dimensional advection–diffusion equations. *International Journal for Numerical Methods in Engineering*, 94(13):1216–1232, 2013. ISSN 1097-0207. doi:[10.1002/nme.4493](https://doi.org/10.1002/nme.4493).

- D. González, E. Cueto, and F. Chinesta. Computational Patient Avatars for Surgery Planning. *Annals of biomedical engineering*, 44(1):35–45, June 2015. ISSN 1573-9686. doi:[10.1007/s10439-015-1362-z](https://doi.org/10.1007/s10439-015-1362-z). URL <http://www.ncbi.nlm.nih.gov/pubmed/26101033>.
- D. González, J. V. Aguado, E. Cueto, E. Abisset-Chavanne, and F. Chinesta. kPCA-Based Parametric Solutions Within the PGD Framework. *Archives of Computational Methods in Engineering*, 25(1):69–86, Jan. 2018. ISSN 1134-3060. doi:[10.1007/s11831-016-9173-4](https://doi.org/10.1007/s11831-016-9173-4).
- H. Hatze. The meaning of the term 'biomechanics'. *Journal of Biomechanics*, 7(2):189–190, Mar. 1974. ISSN 00219290. doi:[10.1016/0021-9290\(74\)90060-8](https://doi.org/10.1016/0021-9290(74)90060-8). URL <http://www.ncbi.nlm.nih.gov/pubmed/4837555>.
- J. Hernández, M. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer Methods in Applied Mechanics and Engineering*, 313:687–722, Jan. 2017. ISSN 0045-7825. doi:[10.1016/J.CMA.2016.10.022](https://doi.org/10.1016/J.CMA.2016.10.022). URL <https://www.sciencedirect.com/science/article/pii/S004578251631355X>.
- G. A. Holzapfel. *Nonlinear solid mechanics : a continuum approach for engineering*. Wiley, 2000. ISBN 0471823198. URL <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471823198.html>.
- M. Horák, D. Ryckelynck, and S. Forest. Hyper-reduction of generalized continua. *Computational Mechanics*, 59(5):753–778, May 2017. ISSN 0178-7675. doi:[10.1007/s00466-016-1371-2](https://doi.org/10.1007/s00466-016-1371-2).
- T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Books, 1989. ISBN 9780486411811.
- 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(39):4135–4195, 2005. ISSN 0045-7825. doi:[10.1016/j.cma.2004.10.008](https://doi.org/10.1016/j.cma.2004.10.008). URL <http://www.sciencedirect.com/science/article/pii/S0045782504005171>.
- T. J. R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):301–313, Jan. 2010. ISSN 00457825. doi:[10.1016/j.cma.2008.12.004](https://doi.org/10.1016/j.cma.2008.12.004). URL <http://www.sciencedirect.com/science/article/pii/S0045782508004295>.
- L. Iapichino, A. Quarteroni, and G. Rozza. Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries. *Computers & Mathematics with Applications*, 71(1):408–430, 2016. ISSN 0898-1221. doi:[10.1016/j.camwa.2015.12.001](https://doi.org/10.1016/j.camwa.2015.12.001). URL <https://www.sciencedirect.com/science/article/pii/S0898122115005696>.

- T. M. Keaveny, X. E. Guo, E. F. Wachtel, T. A. McMahon, and W. C. Hayes. Trabecular bone exhibits fully linear elastic behavior and yields at low strains. *Journal of Biomechanics*, 27(9):1127–1136, 1994. ISSN 00219290. doi:[10.1016/0021-9290\(94\)90053-1](https://doi.org/10.1016/0021-9290(94)90053-1).
- T. S. Keller. Predicting the compressive mechanical behavior of bone. *Journal of biomechanics*, 27(9):1159–68, Sept. 1994. ISSN 0021-9290. URL <http://www.ncbi.nlm.nih.gov/pubmed/7929465>.
- J. Keyak and H. Skinner. Three-dimensional finite element modelling of bone: effects of element size. *Journal of Biomedical Engineering*, 14(6):483–489, Nov. 1992. ISSN 01415425. doi:[10.1016/0141-5425\(92\)90100-Y](https://doi.org/10.1016/0141-5425(92)90100-Y). URL <http://www.sciencedirect.com/science/article/pii/014154259290100Y>.
- J. H. Keyak, I. Y. Lee, and H. B. Skinner. Correlations between orthogonal mechanical properties and density of trabecular bone: Use of different densitometric measures. *Journal of Biomedical Materials Research*, 28(11):1329–1336, Nov. 1994. ISSN 00219304. doi:[10.1002/jbm.820281111](https://doi.org/10.1002/jbm.820281111). URL <http://www.ncbi.nlm.nih.gov/pubmed/7829563>.
- J. H. Keyak, S. A. Rossi, K. A. Jones, and H. B. Skinner. Prediction of femoral fracture load using automated finite element modeling. *Journal of Biomechanics*, 31(2):125–133, May 1998. ISSN 00219290. doi:[10.1016/S0021-9290\(97\)00123-1](https://doi.org/10.1016/S0021-9290(97)00123-1). URL <http://www.sciencedirect.com/science/article/pii/S0021929097001231>.
- J. B. Kruskal and M. Wish. *Multidimensional scaling*. Sage Publications, 1978. ISBN 0803909403.
- P. Ladevèze. *Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation*. Springer, 1999. ISBN 9781461271413. doi:[10.1007/978-1-4612-1432-8](https://doi.org/10.1007/978-1-4612-1432-8).
- P. Ladevèze. On reduced models in nonlinear solid mechanics. *European Journal of Mechanics, A/Solids*, 60:227–237, Nov. 2016. ISSN 09977538. doi:[10.1016/j.euromechsol.2016.08.005](https://doi.org/10.1016/j.euromechsol.2016.08.005). URL <http://www.sciencedirect.com/science/article/pii/S0997753816301036?via%3Dihub>.
- P. Ladevèze and D. Leguillon. Error Estimate Procedure in the Finite Element Method and Applications. *SIAM Journal on Numerical Analysis*, 20(3):485–509, June 1983. ISSN 0036-1429. doi:[10.1137/0720033](https://doi.org/10.1137/0720033).
- M. G. Larson. A Posteriori and a Priori Error Analysis for Finite Element Approximations of Self-Adjoint Elliptic Eigenvalue Problems. *SIAM Journal on Numerical Analysis*, 38(2):608–625, Jan. 2000. ISSN 0036-1429. doi:[10.1137/S0036142997320164](https://doi.org/10.1137/S0036142997320164).

- T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Boundary control and shape optimization for the robust design of bypass anastomoses under uncertainty. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(4):1107–1131, June 2013a. ISSN 0764-583X. doi:[10.1051/m2an/2012059](https://doi.org/10.1051/m2an/2012059).
- T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. A reduced computational and geometrical framework for inverse problems in hemodynamics. *International journal for numerical methods in biomedical engineering*, 29(7):741–76, July 2013b. ISSN 2040-7947. doi:[10.1002/cnm.2559](https://doi.org/10.1002/cnm.2559). URL <http://www.ncbi.nlm.nih.gov/pubmed/23798318>.
- J. A. Lee and M. Verleysen, editors. *Nonlinear Dimensionality Reduction*. Information Science and Statistics. Springer, 2007. ISBN 978-0-387-39350-6. doi:[10.1007/978-0-387-39351-3](https://doi.org/10.1007/978-0-387-39351-3).
- R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, 2007. ISBN 978-0-898716-29-0.
- A. Leygue and E. Verron. A First Step Towards the Use of Proper General Decomposition Method for Structural Optimization. *Archives of Computational Methods in Engineering*, 17(4):465–472, 2010. ISSN 11343060. doi:[10.1007/s11831-010-9052-3](https://doi.org/10.1007/s11831-010-9052-3).
- Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu. Proper orthogonal decomposition and its applications—Part I: Theory. *Journal of Sound and Vibration*, 252(3):527–544, May 2002a. ISSN 0022460X. doi:[10.1006/jsvi.2001.4041](https://doi.org/10.1006/jsvi.2001.4041). URL <http://www.sciencedirect.com/science/article/pii/S0022460X01940416>.
- Y. Liang, W. Lin, H. Lee, S. Lim, K. Lee, and H. Sun. Proper orthogonal decomposition and its applications – Part II: Model reduction for mems dynamical analysis. *Journal of Sound and Vibration*, 256(3):515–532, Sept. 2002b. ISSN 0022460X. doi:[10.1006/jsvi.2002.5007](https://doi.org/10.1006/jsvi.2002.5007). URL <http://www.sciencedirect.com/science/article/pii/S0022460X02950078>.
- B. Luisier, E. Dall’Ara, and D. H. Pahr. Orthotropic HR-pQCT-based FE models improve strength predictions for stance but not for side-way fall loading compared to isotropic QCT-based FE models of human femurs. *Journal of the mechanical behavior of biomedical materials*, 32:287–99, Apr. 2014. ISSN 1878-0180. doi:[10.1016/j.jmbbm.2014.01.006](https://doi.org/10.1016/j.jmbbm.2014.01.006). URL <http://www.sciencedirect.com/science/article/pii/S1751616114000071>.
- C. Maccone. A simple introduction to the KLT (Karhunen—Loève Transform). In *Deep Space Flight and Communications*, pages 151–179. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:[10.1007/978-3-540-72943-3_10](https://doi.org/10.1007/978-3-540-72943-3_10).

-
- A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *International Journal for Numerical Methods in Biomedical Engineering*, 28(6-7):604–625, 2012a. ISSN 2040-7947. doi:[10.1002/cnm.1465](https://doi.org/10.1002/cnm.1465).
- A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *International Journal for Numerical Methods in Fluids*, 70(5):646–670, Oct. 2012b. ISSN 02712091. doi:[10.1002/flid.2712](https://doi.org/10.1002/flid.2712).
- U. Meier, O. López, C. Monserrat, M. C. Juan, and M. Alcañiz. Real-time deformable models for surgery simulation: a survey. *Computer methods and programs in biomedicine*, 77(3):183–97, Mar. 2005. ISSN 0169-2607. doi:[10.1016/j.cmpb.2004.11.002](https://doi.org/10.1016/j.cmpb.2004.11.002). URL <http://www.sciencedirect.com/science/article/pii/S0169260704002093>.
- A. Mena, D. Bel, I. Alfaro, D. González, E. Cueto, and F. Chinesta. Towards a pancreatic surgery simulator based on model order reduction. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):31, Nov. 2015. ISSN 2213-7467. doi:[10.1186/s40323-015-0049-1](https://doi.org/10.1186/s40323-015-0049-1). URL <http://www.amses-journal.com/content/2/1/31>.
- L. Meng, P. Breikopf, B. Raghavan, G. Mauvoisin, O. Bartier, and X. Hernot. Identification of material properties using indentation test and shape manifold learning approach. *Computer Methods in Applied Mechanics and Engineering*, 297:239–257, Dec. 2015. ISSN 0045-7825. doi:[10.1016/J.CMA.2015.09.004](https://doi.org/10.1016/J.CMA.2015.09.004). URL <https://www.sciencedirect.com/science/article/pii/S0045782515002984>.
- R. Milani, A. Quarteroni, and G. Rozza. Reduced basis method for linear elasticity problems with many parameters. *Computer Methods in Applied Mechanics and Engineering*, 197(51-52):4812–4829, Oct. 2008. ISSN 00457825. doi:[10.1016/j.cma.2008.07.002](https://doi.org/10.1016/j.cma.2008.07.002). URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782508002521>.
- R. Milk, S. Rave, and F. Schindler. pyMOR – Generic Algorithms and Interfaces for Model Order Reduction. *SIAM Journal on Scientific Computing*, 38(5):S194–S216, 2016. doi:[10.1137/15M1026614](https://doi.org/10.1137/15M1026614). URL <https://doi.org/10.1137/15M1026614>.
- E. Nadal, F. Chinesta, P. Díez, F. Fuenmayor, and F. Denia. Real time parameter identification and solution reconstruction from experimental data using the Proper Generalized Decomposition. *Computer Methods in Applied Mechanics and Engineering*, 296:113–128, Nov. 2015a. ISSN 00457825. doi:[10.1016/j.cma.2015.07.020](https://doi.org/10.1016/j.cma.2015.07.020). URL <http://www.sciencedirect.com/science/article/pii/S0045782515002327>.

- E. Nadal, A. Leygue, F. Chinesta, M. Beringhier, J. J. Ródenas, and F. J. Fuenmayor. A separated representation of an error indicator for the mesh refinement process under the proper generalized decomposition framework. *Computational Mechanics*, 55(2):251–266, Feb. 2015b. ISSN 1432-0924. doi:[10.1007/s00466-014-1097-y](https://doi.org/10.1007/s00466-014-1097-y).
- S. Niroomandi, I. Alfaro, E. Cueto, and F. Chinesta. Real-time deformable models of non-linear tissues by model reduction techniques. *Computer methods and programs in biomedicine*, 91(3):223–31, Sept. 2008. ISSN 0169-2607. doi:[10.1016/j.cmpb.2008.04.008](https://doi.org/10.1016/j.cmpb.2008.04.008). URL <http://www.ncbi.nlm.nih.gov/pubmed/18534712>.
- S. Niroomandi, I. Alfaro, E. Cueto, and F. Chinesta. Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models. *Computer methods and programs in biomedicine*, 105(1):1–12, Jan. 2012a. ISSN 1872-7565. doi:[10.1016/j.cmpb.2010.06.012](https://doi.org/10.1016/j.cmpb.2010.06.012). URL <http://www.ncbi.nlm.nih.gov/pubmed/20739090>.
- S. Niroomandi, I. Alfaro, D. González, E. Cueto, and F. Chinesta. Real-time simulation of surgery by reduced-order modeling and X-FEM techniques. *International Journal for Numerical Methods in Biomedical Engineering*, 28(5):574–588, 2012b. ISSN 2040-7947. doi:[10.1002/cnm.1491](https://doi.org/10.1002/cnm.1491).
- S. Niroomandi, I. Alfaro, D. González, E. Cueto, and F. Chinesta. Model order reduction in hyperelasticity: a proper generalized decomposition approach. *International Journal for Numerical Methods in Engineering*, 96(3):129–149, 2013a. ISSN 1097-0207. doi:[10.1002/nme.4531](https://doi.org/10.1002/nme.4531).
- S. Niroomandi, D. González, I. Alfaro, F. Bordeu, A. Leygue, E. Cueto, and F. Chinesta. Real-time simulation of biological soft tissues: A PGD approach. *International Journal for Numerical Methods in Biomedical Engineering*, 29(5):586–600, 2013b. ISSN 20407939. doi:[10.1002/cnm.2544](https://doi.org/10.1002/cnm.2544).
- J. Oden and S. Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Computers & Mathematics with Applications*, 41(5-6):735–756, Mar. 2001. ISSN 0898-1221. doi:[10.1016/S0898-1221\(00\)00317-5](https://doi.org/10.1016/S0898-1221(00)00317-5). URL <https://www.sciencedirect.com/science/article/pii/S0898122100003175>.
- S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, Nov. 1988. ISSN 00219991. doi:[10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2). URL <http://www.sciencedirect.com/science/article/pii/0021999188900022>.
- M. Papini, R. Zdero, E. H. Schemitsch, and P. Zalzal. The biomechanics of human femurs in axial and torsional loading: comparison of finite element analysis, human

- cadaveric femurs, and synthetic femurs. *Journal of biomechanical engineering*, 129(1):12–9, Feb. 2007. ISSN 0148-0731. doi:[10.1115/1.2401178](https://doi.org/10.1115/1.2401178). URL <http://www.ncbi.nlm.nih.gov/pubmed/17227093>.
- A. Peano. Gauss-Lobatto integration of high precision tetrahedral elements. *International Journal for Numerical Methods in Engineering*, 18(2):311–313, Feb. 1982. ISSN 0029-5981. doi:[10.1002/nme.1620180211](https://doi.org/10.1002/nme.1620180211).
- S. Perotto and A. Zilio. Hierarchical Model Reduction: Three Different Approaches. In *Numerical Mathematics and Advanced Applications ...*, pages 1–8. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-33134-3. doi:[10.1007/978-3-642-33134-3_88](https://doi.org/10.1007/978-3-642-33134-3_88).
- S. Perotto, A. Ern, and A. Veneziani. Hierarchical Local Model Reduction for Elliptic Problems: A Domain Decomposition Approach. *Multiscale Modeling & Simulation*, 8(4):1102–1127, Jan. 2010. ISSN 1540-3459. doi:[10.1137/090756624](https://doi.org/10.1137/090756624).
- L. Piegl and W. Tiller. *The NURBS Book*. Monographs in Visual Communication. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. ISBN 978-3-540-61545-3. doi:[10.1007/978-3-642-59223-2](https://doi.org/10.1007/978-3-642-59223-2).
- A. Quarteroni and G. Rozza, editors. *Reduced Order Methods for Modeling and Computational Reduction*. Springer International Publishing, Cham, 2014. ISBN 978-3-319-02089-1. doi:[10.1007/978-3-319-02090-7](https://doi.org/10.1007/978-3-319-02090-7).
- A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*, volume 23 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994. ISBN 978-3-540-85267-4. doi:[10.1007/978-3-540-85268-1](https://doi.org/10.1007/978-3-540-85268-1).
- A. Quarteroni and A. Veneziani. Analysis of a Geometrical Multiscale Model Based on the Coupling of ODE and PDE for Blood Flow Simulations. *Multiscale Modeling & Simulation*, 1(2):173–195, Jan. 2003. ISSN 1540-3459. doi:[10.1137/S1540345902408482](https://doi.org/10.1137/S1540345902408482).
- A. Quarteroni, M. Taveri, and A. Veneziani. Computational vascular fluid dynamics: Problems, models and methods. *Computing and Visualization in Science*, 2(4):163–197, Mar. 2000. ISSN 14330369. doi:[10.1007/s007910050039](https://doi.org/10.1007/s007910050039).
- A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(1):3, June 2011. ISSN 2190-5983. doi:[10.1186/2190-5983-1-3](https://doi.org/10.1186/2190-5983-1-3). URL <http://www.mathematicsinindustry.com/content/1/1/3>.
- A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations*, volume 92 of *UNITEXT*. Springer International Publishing, Cham, 2016. ISBN 978-3-319-15430-5. doi:[10.1007/978-3-319-15431-2](https://doi.org/10.1007/978-3-319-15431-2).

- C. Quesada, D. González, I. Alfaro, E. Cueto, and F. Chinesta. Computational vademecums for real-time simulation of surgical cutting in haptic environments. *International Journal for Numerical Methods in Engineering*, 108(10):1230–1247, 2016a. ISSN 1097-0207. doi:[10.1002/nme.5252](https://doi.org/10.1002/nme.5252).
- C. Quesada, D. González, I. Alfaro, E. Cueto, A. Huerta, and F. Chinesta. Real-time simulation techniques for augmented learning in science and engineering. *Visual Computer*, 32(11):1465–1479, May 2016b. ISSN 01782789. doi:[10.1007/s00371-015-1134-7](https://doi.org/10.1007/s00371-015-1134-7).
- A. Ramos and J. A. Simões. Tetrahedral versus hexahedral finite elements in numerical modelling of the proximal femur. *Medical Engineering & Physics*, 28(9):916–924, 2006. ISSN 1350-4533. doi:[10.1016/j.medengphy.2005.12.006](https://doi.org/10.1016/j.medengphy.2005.12.006). URL <http://www.sciencedirect.com/science/article/pii/S1350453305002675>.
- S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science (New York, N.Y.)*, 290(5500):2323–6, Dec. 2000. ISSN 0036-8075. doi:[10.1126/science.290.5500.2323](https://doi.org/10.1126/science.290.5500.2323). URL <http://www.ncbi.nlm.nih.gov/pubmed/11125150>.
- G. Rozza. Reduced-basis methods for elliptic equations in sub-domains with a posteriori error bounds and adaptivity. *Applied Numerical Mathematics*, 55(4):403–424, Dec. 2005. ISSN 01689274. doi:[10.1016/j.apnum.2004.11.004](https://doi.org/10.1016/j.apnum.2004.11.004). URL <http://linkinghub.elsevier.com/retrieve/pii/S0168927404002478>.
- G. Rozza, D. B. P. Huynh, and a. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3):229–275, May 2008. ISSN 11343060. doi:[10.1007/s11831-008-9019-9](https://doi.org/10.1007/s11831-008-9019-9).
- G. Rozza, D. B. P. Huynh, and A. Manzoni. Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants. *Numerische Mathematik*, 125(1):115–152, Mar. 2013. ISSN 0029-599X. doi:[10.1007/s00211-013-0534-8](https://doi.org/10.1007/s00211-013-0534-8).
- D. Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *International Journal for Numerical Methods in Engineering*, 77(1):75–89, Jan. 2009. ISSN 00295981. doi:[10.1002/nme.2406](https://doi.org/10.1002/nme.2406).
- D. Ryckelynck, F. Chinesta, E. Cueto, and A. Ammar. On the a priori model reduction: Overview and recent developments. *Archives of Computational Methods in Engineering*, 13(1):91–128, Mar. 2006. ISSN 1134-3060. doi:[10.1007/BF02905932](https://doi.org/10.1007/BF02905932).
- F. Salmoiraghi, F. Ballarin, L. Heltai, and G. Rozza. Isogeometric analysis-based reduced order modelling for incompressible linear viscous flows in parametrized

- shapes. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):21, Dec. 2016. ISSN 2213-7467. doi:[10.1186/s40323-016-0076-6](https://doi.org/10.1186/s40323-016-0076-6).
- O. Samko, A. Marshall, and P. Rosin. Selection of the optimal parameter value for the Isomap algorithm. *Pattern Recognition Letters*, 27(9):968–979, July 2006. ISSN 01678655. doi:[10.1016/j.patrec.2005.11.017](https://doi.org/10.1016/j.patrec.2005.11.017). URL <http://www.sciencedirect.com/science/article/pii/S0167865505003648>.
- A. Saxena and A. Gupta. Non-linear dimensionality reduction by locally linear isomaps. In N. R. Pal, N. Kasabov, R. K. Mudi, S. Pal, and S. K. Parui, editors, *Neural Information Processing*, pages 1038–1043, Calcutta, 2004. Springer Berlin Heidelberg. doi:[10.1007/978-3-540-30499-9_161](https://doi.org/10.1007/978-3-540-30499-9_161). URL <http://www.springerlink.com/index/14754g3k7gp4lt2y.pdf>.
- J. Simo and T. Hughes. *Computational Inelasticity*, volume 7 of *Interdisciplinary Applied Mathematics*. Springer-Verlag, New York, Feb. 1998. ISBN 0387975209. doi:[10.1016/S0898-1221\(99\)90413-3](https://doi.org/10.1016/S0898-1221(99)90413-3). URL <http://linkinghub.elsevier.com/retrieve/pii/S0898122199904133>.
- Simulia Corp. *Abaqus Analysis User's Guide*. Dassault Systèmes, 2013.
- J. S. Siopack and H. E. Jergesen. Total hip arthroplasty. *The Western journal of medicine*, 162(3):243–9, Mar. 1995. ISSN 0093-0415. URL <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1022709&tool=pmcentrez&rendertype=abstract>.
- E. Stein, editor. *The History of Theoretical, Material and Computational Mechanics - Mathematics Meets Mechanics and Engineering*, volume 1 of *Lecture Notes in Applied Mathematics and Mechanics*. Springer-Verlag, 2014. ISBN 978-3-642-39904-6.
- F. Taddei, A. Pancanti, and M. Viceconti. An improved method for the automatic mapping of computed tomography numbers onto finite element models. *Medical Engineering and Physics*, 26(1):61–69, Jan. 2004. ISSN 13504533. doi:[10.1016/S1350-4533\(03\)00138-3](https://doi.org/10.1016/S1350-4533(03)00138-3). URL <http://www.sciencedirect.com/science/article/pii/S1350453303001383>.
- F. Taddei, L. Cristofolini, S. Martelli, H. S. Gill, and M. Viceconti. Subject-specific finite element models of long bones: An in vitro evaluation of the overall accuracy. *Journal of biomechanics*, 39(13):2457–67, Jan. 2006. ISSN 0021-9290. doi:[10.1016/j.jbiomech.2005.07.018](https://doi.org/10.1016/j.jbiomech.2005.07.018). URL <http://www.sciencedirect.com/science/article/pii/S0021929005003568>.
- J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.)*, 290(5500):2319–23, Dec. 2000. ISSN 0036-8075. doi:[10.1126/science.290.5500.2319](https://doi.org/10.1126/science.290.5500.2319). URL <http://www.sciencemag.org/content/290/5500/2319.full>.

- N. Trabelsi, Z. Yosibash, and C. Milgrom. Validation of subject-specific automated p-FE analysis of the proximal femur. *Journal of biomechanics*, 42(3):234–41, Mar. 2009. ISSN 0021-9290. doi:[10.1016/j.jbiomech.2008.10.039](https://doi.org/10.1016/j.jbiomech.2008.10.039). URL <http://www.sciencedirect.com/science/article/pii/S0021929008005514>.
- N. Trabelsi, Z. Yosibash, C. Wutte, P. Augat, and S. Eberle. Patient-specific finite element analysis of the human femur—a double-blinded biomechanical validation. *Journal of biomechanics*, 44(9):1666–72, June 2011. ISSN 1873-2380. doi:[10.1016/j.jbiomech.2011.03.024](https://doi.org/10.1016/j.jbiomech.2011.03.024). URL <http://www.ncbi.nlm.nih.gov/pubmed/21497354>.
- H. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education Limited, second edition edition, 2007. ISBN 978-0-13-127498-3.
- P. Vidal, L. Gallimard, and O. Polit. Composite beam finite element based on the Proper Generalized Decomposition. *Computers & Structures*, 102-103:76–86, July 2012. ISSN 00457949. doi:[10.1016/j.compstruc.2012.03.008](https://doi.org/10.1016/j.compstruc.2012.03.008). URL <http://www.sciencedirect.com/science/article/pii/S0045794912000752>.
- P. Vidal, L. Gallimard, and O. Polit. Shell finite element based on the Proper Generalized Decomposition for the modeling of cylindrical composite structures. *Computers & Structures*, 132:1–11, Feb. 2014. ISSN 00457949. doi:[10.1016/j.compstruc.2013.10.015](https://doi.org/10.1016/j.compstruc.2013.10.015). URL <http://www.sciencedirect.com/science/article/pii/S0045794913002861>.
- Z. Yosibash, R. Padan, L. Joskowicz, and C. Milgrom. A CT-Based High-Order Finite Element Analysis of the Human Proximal Femur Compared to In-vitro Experiments. *Journal of Biomechanical Engineering*, 129(3):297–309, Nov. 2006. ISSN 0148-0731. doi:[10.1115/1.2720906](https://doi.org/10.1115/1.2720906).
- Z. Yosibash, N. Trabelsi, and C. Milgrom. Reliable simulations of the human proximal femur by high-order finite element analysis validated by experimental observations. *Journal of Biomechanics*, 40(16):3688–3699, 2007. ISSN 0021-9290. doi:[10.1016/j.jbiomech.2007.06.017](https://doi.org/10.1016/j.jbiomech.2007.06.017). URL <http://www.sciencedirect.com/science/article/pii/S0021929007002916>.
- P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *NeuroImage*, 31(3):1116–28, July 2006. ISSN 1053-8119. doi:[10.1016/j.neuroimage.2006.01.015](https://doi.org/10.1016/j.neuroimage.2006.01.015). URL <http://www.sciencedirect.com/science/article/pii/S1053811906000632>.
- K. Zaw, G. R. Liu, B. Deng, and K. B. C. Tan. Rapid identification of elastic modulus of the interface tissue on dental implants surfaces using reduced-basis method and

- a neural network. *Journal of biomechanics*, 42(5):634–41, Mar. 2009. ISSN 1873-2380. doi:[10.1016/j.jbiomech.2008.12.001](https://doi.org/10.1016/j.jbiomech.2008.12.001). URL <http://www.ncbi.nlm.nih.gov/pubmed/19171345>.
- O. Zienkiewicz, , R. Taylor, , and J. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, Oxford, seventh edition edition, 2013. ISBN 978-1-85617-633-0. doi:[10.1016/B978-1-85617-633-0.00019-8](https://doi.org/10.1016/B978-1-85617-633-0.00019-8). URL <https://www.sciencedirect.com/science/article/pii/B9781856176330000198>.
- S. Zlotnik, P. Díez, D. Gonzalez, E. Cueto, and A. Huerta. Effect of the separated approximation of input data in the accuracy of the resulting PGD solution. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):28, Nov. 2015a. ISSN 2213-7467. doi:[10.1186/s40323-015-0052-6](https://doi.org/10.1186/s40323-015-0052-6). URL <http://www.amses-journal.com/content/2/1/28>.
- S. Zlotnik, P. Díez, D. Modesto, and A. Huerta. Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *International Journal for Numerical Methods in Engineering*, 103(10):737–758, Sept. 2015b. ISSN 00295981. doi:[10.1002/nme.4909](https://doi.org/10.1002/nme.4909).

Appendix A

PGD formulation for static condensation

This appendix introduces the PGD formulation for linear elasticity problems using static condensation to apply boundary conditions. Three cases have been considered:

1. Taking Dirichlet boundary condition as the extra coordinate.
2. Taking material properties as extra coordinates, with prescribed Dirichlet boundary condition.
3. Taking material properties as extra coordinates, with prescribed Dirichlet and Neumann boundary conditions.

A.1 Variable Dirichlet boundary condition

Consider the following standard static condensation

$$\begin{bmatrix} \mathbf{K}_d & \mathbf{K}_{df} \\ \mathbf{K}_{df}^T & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \mathbf{U}_d \\ \mathbf{U}_f \end{bmatrix} = \begin{bmatrix} \mathbf{R}_d \\ \mathbf{F}_f \end{bmatrix}, \quad (\text{A.1})$$

where the stiffness matrix \mathbf{K} is partitioned, with subscript d denotes Dirichlet boundary, and subscript f denotes the free boundary. The displacement DOF \mathbf{U} and load vector \mathbf{F} are also partitioned accordingly. In this case, prescribed displacement \mathbf{U}_d is

variable, while the external load \mathbf{F}_f is known. We can solve the displacement of free nodes and the reaction force by

$$\begin{aligned}\mathbf{U}_f &= \mathbf{K}_f^{-1} \left(\mathbf{F}_f - \mathbf{K}_{df}^T \mathbf{U}_d \right), \\ \mathbf{R}_d &= \mathbf{K}_d \mathbf{U}_d + \mathbf{K}_{df} \mathbf{U}_f.\end{aligned}\tag{A.2}$$

Consider the following PGD separated representation of the free displacement:

$$\mathbf{U}_f^n(g) = \sum_{i=1}^n \mathbf{U}_i \omega_i(g),\tag{A.3}$$

the corresponding weak form is

$$\mathbf{K}_f \int \omega_n \mathbf{U}_f^n(g) dg = \int \left(\mathbf{F}_f - \mathbf{K}_{df}^T \mathbf{U}_d \right) \omega_n dg.\tag{A.4}$$

With the assumption of

$$\mathbf{U}_d = g \cdot \mathbf{I}_u,\tag{A.5}$$

where $\mathbf{I}_u = [1, 1, \dots, 1]^T$, the weak form turns into

$$\mathbf{K}_f \sum_{i=1}^n \mathbf{U}_i \int \omega_n \omega_i dg = \int \mathbf{F}_f \omega_n dg - \mathbf{K}_{df}^T \mathbf{I}_u \int g \omega_n dg.\tag{A.6}$$

Then the PGD alternative direction iteration reads:

1. Assume ω_i are known, solve \mathbf{U}_n from

$$\mathbf{K}_f \mathbf{U}_n \int \omega_n^2 dg = \int \mathbf{F}_f \omega_n dg - \mathbf{K}_{df}^T \mathbf{I}_u \int g \omega_n dg - \mathbf{K}_f \sum_{i=1}^{n-1} \int \omega_n \omega_i dg;\tag{A.7}$$

2. Assume \mathbf{U}_i are known, solve ω_n from

$$\begin{aligned}\mathbf{U}_n^T \mathbf{K}_f \mathbf{U}_n \int \omega_n^* \omega_n dg &= \mathbf{U}_n^T \int \mathbf{F}_f \omega_n^* dg - \mathbf{U}_n^T \mathbf{K}_{df}^T \mathbf{I}_u \int g \omega_n^* dg \\ &\quad - \mathbf{U}_n^T \mathbf{K}_f \sum_{i=1}^{n-1} \mathbf{U}_i \int \omega_n^* \omega_i dg.\end{aligned}\tag{A.8}$$

Discretise $\omega(g)$ with FE method as

$$\omega_i(g) \approx \mathbf{N}^T(g) \mathbf{d}_i,\tag{A.9}$$

where \mathbf{N} is the shape function vector, and \mathbf{d}_i is the DOF vector.

We define the following matrices for convenience:

$$\mathbf{M} = \int \mathbf{N}\mathbf{N}^T dg, \quad (\text{A.10})$$

$$\mathbf{P} = \int g\mathbf{N} dg, \quad (\text{A.11})$$

$$\mathbf{Q} = \int \mathbf{N} dg. \quad (\text{A.12})$$

The discretised formulation reads:

1. Assume \mathbf{d}_i are known, solve \mathbf{U}_n from

$$\mathbf{K}_f \mathbf{U}_n (\mathbf{d}_n^T \mathbf{M} \mathbf{d}_n) = \mathbf{F}_f (\mathbf{d}_n^T \mathbf{Q}) - \mathbf{K}_{df}^T \mathbf{I}_u (\mathbf{d}_n^T \mathbf{P}) - \mathbf{K}_f \sum_{i=1}^{n-1} \mathbf{U}_i (\mathbf{d}_n^T \mathbf{M} \mathbf{d}_i); \quad (\text{A.13})$$

2. Assume \mathbf{U}_i are known, solve \mathbf{d}_n from

$$(\mathbf{U}_n^T \mathbf{K}_f \mathbf{U}_n) \mathbf{M} \mathbf{d}_n = (\mathbf{U}_n^T \mathbf{F}_f) \mathbf{Q} - (\mathbf{U}_n^T \mathbf{K}_{df}^T \mathbf{I}_u) \mathbf{P} - \mathbf{U}_n^T \mathbf{K}_f \sum_{i=1}^{n-1} \mathbf{U}_i (\mathbf{M} \mathbf{d}_i). \quad (\text{A.14})$$

A.2 Prescribed Dirichlet BC with variable material properties

In case of only displacement boundary condition is prescribed, with no external force, consider the following static condensation

$$\begin{bmatrix} \mathbf{K}_d & \mathbf{K}_{df} \\ \mathbf{K}_{df}^T & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \mathbf{U}_d \\ \mathbf{U}_f \end{bmatrix} = \begin{bmatrix} \mathbf{R}_d \\ \mathbf{0} \end{bmatrix}, \quad (\text{A.15})$$

we can solve

$$\begin{aligned} \mathbf{U}_f &= -\mathbf{K}_f^{-1} \mathbf{K}_{df}^T \mathbf{U}_d, \\ \mathbf{R}_d &= \mathbf{K}_d \mathbf{U}_d + \mathbf{K}_{df} \mathbf{U}_f = \left(\mathbf{K}_d - \mathbf{K}_{df} \mathbf{K}_f^{-1} \mathbf{K}_{df}^T \right) \mathbf{U}_d. \end{aligned} \quad (\text{A.16})$$

Suppose the material property μ is unknown, consider the following PGD separated representation:

$$\mathbf{U}_f^n(\mu) = \sum_{i=1}^n \mathbf{U}_i \omega_i(\mu), \quad (\text{A.17})$$

the corresponding weak form is

$$\int \omega_n(\mu) \mathbf{K}_f(\mu) \mathbf{U}_f^n(\mu) d\mu = - \int \omega_n(\mu) \mathbf{K}_{df}^T(\mu) \mathbf{U}_d d\mu, \quad (\text{A.18})$$

which can be rewritten as

$$\sum_{i=1}^n \left(\int \mathbf{K}_f(\mu) \omega_n(\mu) \omega_i(\mu) d\mu \right) \mathbf{U}_i = - \left(\int \mathbf{K}_{df}^T(\mu) \omega_n(\mu) d\mu \right) \mathbf{U}_d. \quad (\text{A.19})$$

The PGD alternative direction iteration reads:

1. Assume ω_i are known, solve \mathbf{U}_n from

$$\left(\int \mathbf{K}_f(\mu) \omega_n \omega_n d\mu \right) \mathbf{U}_n = - \left(\int \mathbf{K}_{df}^T(\mu) \omega_n d\mu \right) \mathbf{U}_d - \sum_{i=1}^{n-1} \left(\int \mathbf{K}_f(\mu) \omega_n \omega_i d\mu \right) \mathbf{U}_i; \quad (\text{A.20})$$

2. Assume \mathbf{U}_i are known, solve ω_n from

$$\begin{aligned} \mathbf{U}_n^T \left(\int \mathbf{K}_f(\mu) \omega_n \omega_n d\mu \right) \mathbf{U}_n &= - \mathbf{U}_n^T \left(\int \mathbf{K}_{df}^T(\mu) \omega_n d\mu \right) \mathbf{U}_d \\ &\quad - \mathbf{U}_n^T \sum_{i=1}^{n-1} \left(\int \mathbf{K}_f(\mu) \omega_n \omega_i d\mu \right) \mathbf{U}_i. \end{aligned} \quad (\text{A.21})$$

Discretise $\omega_i(\mu)$ with FE method as

$$\omega_i(\mu) \approx \mathbf{N}^T(\mu) \boldsymbol{\omega}_i, \quad (\text{A.22})$$

and suppose that

$$\mathbf{K}(\mu) = \mu \hat{\mathbf{K}}, \quad (\text{A.23})$$

we define the following matrices for convenience:

$$\mathbf{H} = \int \mu \mathbf{N} \mathbf{N}^T d\mu, \quad (\text{A.24})$$

$$\mathbf{P} = \int \mu \mathbf{N} d\mu. \quad (\text{A.25})$$

The discretised formulation reads:

1. Assume $\boldsymbol{\omega}_i$ are known, solve \mathbf{U}_n from

$$\hat{\mathbf{K}}_f \mathbf{U}_n (\boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_n) = - \hat{\mathbf{K}}_{df}^T \mathbf{U}_d (\boldsymbol{\omega}_n^T \mathbf{P}) - \hat{\mathbf{K}}_f \sum_{i=1}^{n-1} \mathbf{U}_i (\boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_i); \quad (\text{A.26})$$

2. Assume \mathbf{U}_i are known, solve $\boldsymbol{\omega}_n$ from

$$(\mathbf{U}_n^T \hat{\mathbf{K}}_f \mathbf{U}_n) \mathbf{H} \boldsymbol{\omega}_n = - (\mathbf{U}_n^T \hat{\mathbf{K}}_{df}^T \mathbf{U}_d) \mathbf{P} - \mathbf{U}_n^T \hat{\mathbf{K}}_f \sum_{i=1}^{n-1} \mathbf{U}_i (\mathbf{H} \boldsymbol{\omega}_i). \quad (\text{A.27})$$

A.3 Prescribed BC with variable material properties

In case of both displacement boundary condition and external force are prescribed, consider the following static condensation

$$\begin{bmatrix} \mathbf{K}_d & \mathbf{K}_{df} \\ \mathbf{K}_{df}^T & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \mathbf{U}_d \\ \mathbf{U}_f \end{bmatrix} = \begin{bmatrix} \mathbf{R}_d \\ \mathbf{F}_f \end{bmatrix}, \quad (\text{A.28})$$

we can solve

$$\begin{aligned} \mathbf{U}_f &= \mathbf{K}_f^{-1}(\mathbf{F}_f - \mathbf{K}_{df}^T \mathbf{U}_d), \\ \mathbf{R}_d &= \mathbf{K}_d \mathbf{U}_d + \mathbf{K}_{df} \mathbf{U}_f = \left(\mathbf{K}_d - \mathbf{K}_{df} \mathbf{K}_f^{-1} \mathbf{K}_{df}^T \right) \mathbf{U}_d + \mathbf{K}_{df} \mathbf{K}_f^{-1} \mathbf{F}_f. \end{aligned} \quad (\text{A.29})$$

Suppose the material property μ is unknown, consider the following PGD separated representation:

$$\mathbf{U}_f^n(\mu) = \sum_{i=1}^n \mathbf{U}_i \omega_i(\mu). \quad (\text{A.30})$$

the corresponding weak form is

$$\int \omega_n(\mu) \mathbf{K}_f(\mu) \mathbf{U}_f^n(\mu) d\mu = \int \omega_n(\mu) \mathbf{F}_f d\mu - \int \omega_n(\mu) \mathbf{K}_{df}^T(\mu) \mathbf{U}_d d\mu, \quad (\text{A.31})$$

which can be rewritten as

$$\sum_{i=1}^n \left(\int \mathbf{K}_f(\mu) \omega_n(\mu) \omega_i(\mu) d\mu \right) \mathbf{U}_i = \int \mathbf{F}_f \omega_n(\mu) d\mu - \left(\int \mathbf{K}_{df}^T(\mu) \omega_n(\mu) d\mu \right) \mathbf{U}_d. \quad (\text{A.32})$$

The PGD alternative direction iteration reads:

1. Assume ω_i are known, solve \mathbf{U}_n from

$$\begin{aligned} \left(\int \mathbf{K}_f(\mu) \omega_n \omega_n d\mu \right) \mathbf{U}_n &= \int \mathbf{F}_f \omega_n d\mu - \left(\int \mathbf{K}_{df}^T(\mu) \omega_n d\mu \right) \mathbf{U}_d \\ &\quad - \sum_{i=1}^{n-1} \left(\int \mathbf{K}_f(\mu) \omega_n \omega_i d\mu \right) \mathbf{U}_i; \end{aligned} \quad (\text{A.33})$$

2. Assume \mathbf{U}_i are known, solve ω_n from

$$\begin{aligned} \mathbf{U}_n^T \left(\int \mathbf{K}_f(\mu) \omega_n \omega_n d\mu \right) \mathbf{U}_n &= \mathbf{U}_n^T \int \mathbf{F}_f \omega_n d\mu - \mathbf{U}_n^T \left(\int \mathbf{K}_{df}^T(\mu) \omega_n d\mu \right) \mathbf{U}_d \\ &\quad - \mathbf{U}_n^T \sum_{i=1}^{n-1} \left(\int \mathbf{K}_f(\mu) \omega_n \omega_i d\mu \right) \mathbf{U}_i. \end{aligned} \quad (\text{A.34})$$

Discretise $\omega_i(\mu)$ with FE method as

$$\omega_i(\mu) \approx \mathbf{N}^T(\mu) \boldsymbol{\omega}_i, \quad (\text{A.35})$$

and suppose that

$$\mathbf{K}(\mu) = \mu \hat{\mathbf{K}}, \quad (\text{A.36})$$

we define the following matrices for convenience:

$$\mathbf{H} = \int \mu \mathbf{N} \mathbf{N}^T d\mu, \quad (\text{A.37})$$

$$\mathbf{P} = \int \mu \mathbf{N} d\mu, \quad (\text{A.38})$$

$$\mathbf{Q} = \int \mathbf{N} d\mu. \quad (\text{A.39})$$

The discretised formulation reads:

1. Assume $\boldsymbol{\omega}_i$ are known, solve \mathbf{U}_n from

$$\hat{\mathbf{K}}_f \mathbf{U}_n (\boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_n) = \mathbf{F}_f (\boldsymbol{\omega}_n^T \mathbf{Q}) - \hat{\mathbf{K}}_{df}^T \mathbf{U}_d (\boldsymbol{\omega}_n^T \mathbf{P}) - \hat{\mathbf{K}}_f \sum_{i=1}^{n-1} \mathbf{U}_i (\boldsymbol{\omega}_n^T \mathbf{H} \boldsymbol{\omega}_i); \quad (\text{A.40})$$

2. Assume \mathbf{U}_i are known, solve $\boldsymbol{\omega}_n$ from

$$(\mathbf{U}_n^T \hat{\mathbf{K}}_f \mathbf{U}_n) \mathbf{H} \boldsymbol{\omega}_n = (\mathbf{U}_n^T \mathbf{F}_f) \mathbf{Q} - (\mathbf{U}_n^T \hat{\mathbf{K}}_{df}^T \mathbf{U}_d) \mathbf{P} - \mathbf{U}_n^T \hat{\mathbf{K}}_f \sum_{i=1}^{n-1} \mathbf{U}_i (\mathbf{H} \boldsymbol{\omega}_i). \quad (\text{A.41})$$

Appendix B

PGD formulation for Young modulus and Poisson's ratio

This appendix is dedicated to the derivation of PGD formulation taking both Young modulus E and Poisson's ratio ν as the extra coordinates. The 2D case of plane stress is considered for instance. However, the resultant formulation can be generalised to plane strain and 3D case with proper modifications.

B.1 Basics of plane stress elasticity

For plane stress problems, there are two independent parameters of the material property: E, ν . Assume the ranges for the properties are given.

The stress-strain relationship reads:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix}. \quad (\text{B.1})$$

B.2 PGD separated representation

With nodal displacement vector \mathbf{U} as the unknown, we have

$$\mathbf{U}_{\text{PGD}}^m(E, \nu) = \sum_{i=1}^m \boldsymbol{\chi}_i \alpha_i(E) \omega_i(\nu). \quad (\text{B.2})$$

The PGD alternative direction iteration reads:

1. Assume $\alpha_n(E)$, $\omega_n(\nu)$ and $\mathbf{U}_{\text{PGD}}^{n-1}$ are know, solve $\boldsymbol{\chi}_n$ from

$$\int_{I_E} \int_{I_\nu} \mathbf{K} \boldsymbol{\chi}_n \alpha_n^2 \omega_n^2 dE d\nu = \int_{I_E} \int_{I_\nu} \left[\mathbf{F} - \sum_{i=1}^{n-1} \mathbf{K} \boldsymbol{\chi}_i \alpha_i \omega_i \right] \alpha_n \omega_n dE d\nu; \quad (\text{B.3})$$

2. Assume $\boldsymbol{\chi}_n$, $\omega_n(\nu)$ and $\mathbf{U}_{\text{PGD}}^{n-1}$ are know, solve $\alpha_n(E)$ from

$$\alpha_n \int_{I_\nu} \boldsymbol{\chi}_n^T \mathbf{K} \boldsymbol{\chi}_n \omega_n^2 d\nu = \int_{I_\nu} \boldsymbol{\chi}_n^T \left[\mathbf{F} - \sum_{i=1}^{n-1} \mathbf{K} \boldsymbol{\chi}_i \alpha_i \omega_i \right] \omega_n d\nu; \quad (\text{B.4})$$

3. Assume $\alpha_n(E)$, $\boldsymbol{\chi}_n$ and $\mathbf{U}_{\text{PGD}}^{n-1}$ are know, solve $\omega_n(\nu)$ from

$$\omega_n \int_{I_E} \boldsymbol{\chi}_n^T \mathbf{K} \boldsymbol{\chi}_n \alpha_n^2 dE = \int_{I_E} \boldsymbol{\chi}_n^T \left[\mathbf{F} - \sum_{i=1}^{n-1} \mathbf{K} \boldsymbol{\chi}_i \alpha_i \omega_i \right] \alpha_n dE. \quad (\text{B.5})$$

B.3 Decomposition of elemental stiffness matrix

With definition of

$$\mathbf{D}_1 := \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D}_2 := \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.6})$$

we may write:

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} = \frac{E}{2(1-\nu)} \mathbf{D}_1 + \frac{E}{2(1+\nu)} \mathbf{D}_2 \quad (\text{B.7})$$

The element stiffness matrix can be decomposed by

$$\mathbf{K} = \int_{\Omega_e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV = \frac{E}{2(1-\nu)} \mathbf{K}_1 + \frac{E}{2(1+\nu)} \mathbf{K}_2, \quad (\text{B.8})$$

where $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n]$ is the strain-displacement matrix for n-node elements with shape functions N_i , and the stiffness-like matrices are defined by

$$\begin{aligned} \mathbf{K}_1 &:= \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_1 \mathbf{B} dV, \\ \mathbf{K}_2 &:= \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_2 \mathbf{B} dV. \end{aligned} \quad (\text{B.9})$$

Now we may rewrite, for instance, the LHS for the mechanical problem as:

$$\begin{aligned} \int_{I_E} \int_{I_\nu} \mathbf{K} \boldsymbol{\chi}_n \alpha_n^2(E) \omega_n^2(\nu) dE d\nu &= \mathbf{K}_1 \boldsymbol{\chi}_n \int_{I_E} E \alpha_n^2(E) dE \int_{I_\nu} \frac{1}{2(1-\nu)} \omega_n^2(\nu) d\nu \\ &\quad + \mathbf{K}_2 \boldsymbol{\chi}_n \int_{I_E} E \alpha_n^2(E) dE \int_{I_\nu} \frac{1}{2(1+\nu)} \omega_n^2(\nu) d\nu. \end{aligned} \quad (\text{B.10})$$

B.3.1 Discretised formulation

Consider the following FE discretisation:

$$\alpha_i(E) = \mathbf{N}_E^T(E) \boldsymbol{\alpha}_i, \quad (\text{B.11})$$

$$\omega_i(\nu) = \mathbf{N}_\nu^T(\nu) \boldsymbol{\omega}_i, \quad (\text{B.12})$$

for convenience, we define the following matrices

$$\begin{aligned} \mathbf{H} &:= \int_{I_E} E \mathbf{N}_E \mathbf{N}_E^T dE, \\ \mathbf{G}_1 &:= \frac{1}{2} \int_{I_\nu} \frac{1}{1-\nu} \mathbf{N}_\nu \mathbf{N}_\nu^T d\nu, \\ \mathbf{G}_2 &:= \frac{1}{2} \int_{I_\nu} \frac{1}{1+\nu} \mathbf{N}_\nu \mathbf{N}_\nu^T d\nu. \end{aligned} \quad (\text{B.13})$$

and the following vectors

$$\begin{aligned} \mathbf{Q} &:= \int_{I_E} \mathbf{N}_E dE, \\ \mathbf{P} &:= \int_{I_\nu} \mathbf{N}_\nu d\nu. \end{aligned} \quad (\text{B.14})$$

The integrals can be obtained by:

$$\int_{I_E} E \alpha_n^2(E) dE = \boldsymbol{\alpha}_n^T \mathbf{H} \boldsymbol{\alpha}_n, \quad (\text{B.15})$$

$$\int_{I_\nu} \frac{1}{2(1-\nu)} \omega_n^2(\nu) d\nu = \boldsymbol{\omega}_n^T \mathbf{G}_1 \boldsymbol{\omega}_n, \quad (\text{B.16})$$

$$\int_{I_\nu} \frac{1}{2(1+\nu)} \omega_n^2(\nu) d\nu = \boldsymbol{\omega}_n^T \mathbf{G}_2 \boldsymbol{\omega}_n, \quad (\text{B.17})$$

$$\int_{I_E} \alpha_n(E) dE = \boldsymbol{\alpha}_n^T \mathbf{Q}, \quad (\text{B.18})$$

$$\int_{I_\nu} \omega_n(\nu) d\nu = \boldsymbol{\omega}_n^T \mathbf{P}. \quad (\text{B.19})$$

Finally the discretised PGD alternative direction iteration scheme reads

1. Assume α_n , ω_n and $\mathbf{U}_{\text{PGD}}^{n-1}$ are know, solve χ_n from LHS = RHS, where

$$\begin{aligned} \text{LHS} &= (\alpha_n^T \mathbf{H} \alpha_n) \left[(\omega_n^T \mathbf{G}_1 \omega_n) \mathbf{K}_1 + (\omega_n^T \mathbf{G}_2 \omega_n) \mathbf{K}_2 \right] \chi_n, \\ \text{RHS} &= \mathbf{F}(\alpha_n^T \mathbf{Q})(\omega_n^T \mathbf{P}) - \sum_{i=1}^{n-1} (\alpha_n^T \mathbf{H} \alpha_i) \left[(\omega_n^T \mathbf{G}_1 \omega_i) \mathbf{K}_1 + (\omega_n^T \mathbf{G}_2 \omega_i) \mathbf{K}_2 \right] \chi_i; \end{aligned} \quad (\text{B.20})$$

2. Assume χ_n , ω_n and $\mathbf{U}_{\text{PGD}}^{n-1}$ are know, solve α_n from LHS = RHS, where

$$\begin{aligned} \text{LHS} &= \left[(\omega_n^T \mathbf{G}_1 \omega_n)(\chi_n^T \mathbf{K}_1 \chi_n) + (\omega_n^T \mathbf{G}_2 \omega_n)(\chi_n^T \mathbf{K}_2 \chi_n) \right] \mathbf{H} \alpha_n, \\ \text{RHS} &= (\chi_n^T \mathbf{F}) \mathbf{Q}(\omega_n^T \mathbf{P}) - \sum_{i=1}^{n-1} \left[(\omega_n^T \mathbf{G}_1 \omega_i)(\chi_n^T \mathbf{K}_1 \chi_i) + (\omega_n^T \mathbf{G}_2 \omega_i)(\chi_n^T \mathbf{K}_2 \chi_i) \right] \mathbf{H} \alpha_i; \end{aligned} \quad (\text{B.21})$$

3. Assume α_n , χ_n and $\mathbf{U}_{\text{PGD}}^{n-1}$ are know, solve ω_n from LHS = RHS, where

$$\begin{aligned} \text{LHS} &= (\alpha_n^T \mathbf{H} \alpha_n) \left[(\chi_n^T \mathbf{K}_1 \chi_n) \mathbf{G}_1 + (\chi_n^T \mathbf{K}_2 \chi_n) \mathbf{G}_2 \right] \omega_n, \\ \text{RHS} &= \mathbf{P}(\alpha_n^T \mathbf{Q})(\chi_n^T \mathbf{F}) - \sum_{i=1}^{n-1} (\alpha_n^T \mathbf{H} \alpha_i) \left[(\chi_n^T \mathbf{K}_1 \chi_i) \mathbf{G}_1 + (\chi_n^T \mathbf{K}_2 \chi_i) \mathbf{G}_2 \right] \omega_i. \end{aligned} \quad (\text{B.22})$$

Appendix C

Nonlinear dimensionality reduction for geometric parameters

In order to find an appropriate approach for geometric parametrisation of biomechanical tissues, it is necessary to reduce the high dimensionality of the images which contains geometric description of the shape. Present appendix is designated for a brief discussion on the dimensionality reduction which is based on manifold learning method which comes from machine learning techniques.

C.1 Nonlinear dimensionality reduction with Isomap

Science and engineering researchers often work with large volumes of data involving large dimensions. In this case, the data we are working with has a 512×512 resolution for each image in each CT image set that contains 456 images, so the total dimension of each sample of data is 119 537 664. It is believed the data must contain much redundancy, it is necessary to find meaningful low-dimensional structures hidden in their high-dimensional representations [Lee and Verleysen \[2007\]](#).

Consider a group of data, in which each sample can be flattened to a m -dimensional vector $\mathbf{x} \in \mathbb{R}^m$. Mathematically, it is assumed that the data live in a m -dimensional manifold $\mathcal{M} \subseteq \mathbb{R}^m$. For analyses involving large volumes of data, SVD or PCA can be used as it is discussed in [Chapter 1](#). However, in most cases, the global structure of \mathcal{M} is nonlinear, and SVD or PCA works only for linear manifolds or locally on nonlinear ones. Various techniques have been developed to find the implicit connections between local behaviour and global structures on nonlinear manifolds, such as curvilinear component analysis (CCA) [Demartines and Herault \[1997\]](#), kernel PCA (kPCA), locally linear embedding (LLE) [Roweis and Saul \[2000\]](#), etc.

As mentioned in [Chapter 1](#), it is not trivial to reduce the dimensionality of the geometric shape space to a reasonable low order due to its nonlinearity. Inspired by latest literature such as [Meng et al. \[2015\]](#), [González et al. \[2018\]](#), it is worth trying to investigate the power of nonlinear dimension reduction techniques. In this appendix, we use Isomap [Tenenbaum et al. \[2000\]](#) to perform the dimensionality reduction on the nonlinear manifold of geometric shape. Isomap can be regarded as a type of kPCA, using the geodesic distance matrix as the kernel, it can also be interpreted as an extension from the classical multi-dimensional scaling (MDS) [Kruskal and Wish \[1978\]](#), [Cox and Cox \[2001\]](#). Isomap can also be embedded into the LLE framework [Saxena and Gupta \[2004\]](#).

The idea of Isomap is to find a global pairwise distance matrix of data samples, and perform spectral decomposition on the distance matrix, obtaining the low-dimensional manifold \mathcal{P} through truncation on the dominant eigenvalues and eigenvectors. The key to the construction of the distance matrix is, instead of computing Euclidean distance globally, the global distance is accumulated by local Euclidean distances through a weighted graph of the high-dimensional data. The procedure is as follows:

1. Input a set of n data points $\mathbf{x}_i \in \mathcal{M} \subseteq \mathbb{R}^m$, ($i = 1, 2, \dots, n$), usually $m \gg n$;
2. Build a graph of the data points with K -nearest neighbour (KNN) or ϵ -ball method;
3. Weight the graph by labelling each edge with Euclidean length

$$d(i, j) = \|\mathbf{x}_i - \mathbf{x}_j\|_E; \tag{C.1}$$

4. Compute all pairwise graph distance matrix \mathbf{D} using a *shortest path* algorithm:

- Floyd-Warshall, $O(n^3)$: $D_{ij} = \min\{d(i, j), d(i, k) + d(k, j)\}$, ($k = 1, 2, \dots, K$),
- Dijkstra's algorithm, $O(Kn^2 \log n)$;

5. Convert the matrix of distances \mathbf{D} into a Gram matrix \mathbf{S} by double centering

$$\begin{aligned} S_{ij} &= -\frac{1}{2}H_{ik}(D_{kl})^2H_{lj}, \\ H_{ij} &= \delta_{ij} - \frac{1}{n}; \end{aligned} \tag{C.2}$$

6. Compute spectral decomposition

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T; \tag{C.3}$$

7. A P -dimensional representation $\hat{\mathbf{Y}}$ is obtained by taking the first P rows of \mathbf{Y}

$$\begin{aligned} \mathbf{Y} &= \mathbf{\Lambda}^{1/2}\mathbf{U}^T, \\ \mathbf{S} &= \mathbf{Y}^T\mathbf{Y}; \end{aligned} \tag{C.4}$$

8. Output a quasi-isometric, P -dimensional embedding ($P = 2$ for a 2D plot) with data points $\hat{\mathbf{y}}_i \in \mathcal{P} \subseteq \mathbb{R}^P$, ($i = 1, 2, \dots, n$).

Typically, there are two approaches to estimate the error of Isomap results:

- **Cost function** approach: compute the pairwise spatial distance matrix $\hat{\mathbf{D}}$ in the lower-dimensional manifold, and the corresponding Gram matrix $\hat{\mathbf{S}}$, then the cost function is

$$E = \|\mathbf{S} - \hat{\mathbf{S}}\|_{L^2}, \tag{C.5}$$

where $\|\mathbf{A}\|_{L^2} = \sqrt{\sum_{i,j} A_{ij}^2}$ is the Frobenius norm.

- **Residual variance** approach: compute the Pearson product-moment correlation coefficient between the flattened pairwise spatial distance matrices in each space, and then the residual variance is

$$r = 1 - \rho_{\mathbf{D}, \hat{\mathbf{D}}}^2, \tag{C.6}$$

where $\rho_{\mathbf{D}, \hat{\mathbf{D}}} = \frac{\text{cov}(\mathbf{D}, \hat{\mathbf{D}})}{\sigma_{\mathbf{D}}\sigma_{\hat{\mathbf{D}}}}$ is the Pearson correlation coefficient.

The only free parameter for Isomap is the nearest neighbour number K or the radius of the neighbourhood ball ϵ . Taking K for example, an optimal choice is chosen following [Samko et al. \[2006\]](#):

1. Find S_K : the minimas of cost functions $E(K)$ calculated with selected range of K ;
2. Find the optimal $K_{\text{opt}} = \arg \min_{K \in S_K}(r) = \arg \min_{K \in S_K} (1 - \rho_{\mathbf{D}, \hat{\mathbf{D}}}^2)$.

In practice implementation of Isomap, we used a Python function provided by the machine learning package `Scikit Learn` <http://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html>. The main input parameters for this Isomap function are:

- neighbours algorithm: K -nearest or ϵ -ball.
- number of neighbours K or ball radius ϵ .
- number of lower dimensional space P ($P = 2$ by default).
- shortest path method: FW (Floyd-Warshall) or D (Dijkstra).

C.2 Geometric shape analysis of proximal femur

To analysis the geometric shape of human proximal femur, we have collected 25 samples of CT image sets as illustrated in [Figure C.1](#). The resolution of all CT images are the same, but the number of slices is different from one set to another. To obtain a uniform dimension of description of the data, all the CT images are segmented into STL surfaces, and then level sets are created based on the distance fields to the femur surfaces, as shown in [Figure C.2](#). The level sets have a uniform dimension of $25 \times 21 \times 61$, and thus the dimension of each sample is 32 025.

Applying the Isomap algorithm on the collected data, the 2D and 3D embedding is plotted in [Figure C.3](#) and [C.4](#) respectively. From the embedding plots we can observe the essential parameter could be the length and radius of the shaft. The residual variance for the dimension of embedded low-dimensional space is plotted in [Figure C.5](#). It can be seen the residual variance drops to below 5×10^{-4} when $n \geq 3$. However, unlike PCA analysis, Isomap cannot provide the eigenshapes corresponding to each low-dimensional embedding.

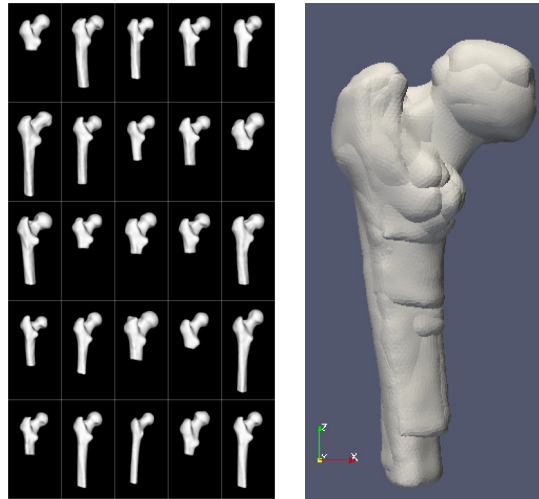


Figure C.1: Collected femur samples. Left: individual view. Right: Pile-up view.

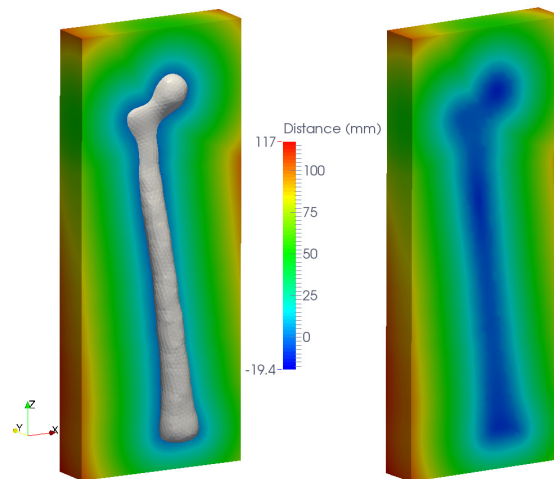


Figure C.2: A level set example.

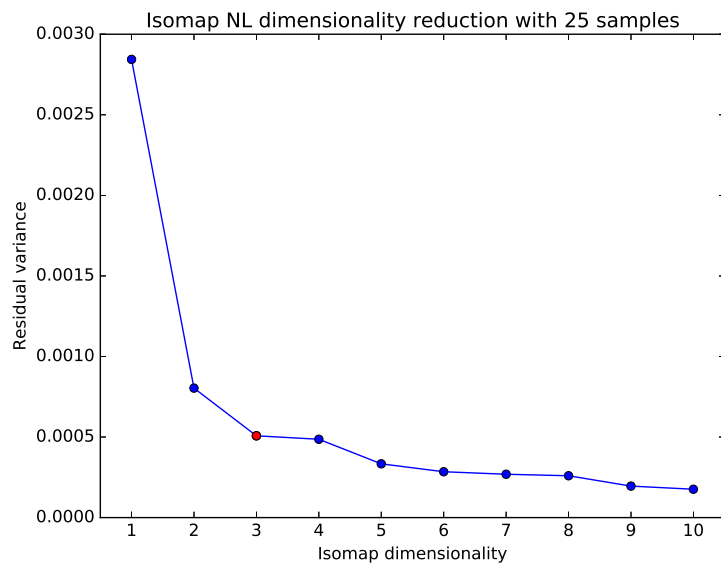


Figure C.5: Residual variance of Isomap result.

