Modelling of structures using the Isogeometric Method

By
Dimitrios Tsapetis

School of Civil Engineering
Institute of Structural Analysis and Antiseismic Research

National Technical University of Athens

Supervisor: Emeritus Professor Manolis Papadrakakis

A thesis submitted for the degree of
Doctor of Philosophy

January, 2021

APPROVAL

PhD THESIS EXAMINATION COMMITTEE:

Emeritus Professor Manolis Papadrakakis
(Supervisor and Principal Advisor of the Committee)
National Technical University of Athens
School of Civil Engineering

Professor Konstantinos Spiliopoulos
(Member Advisor of the Committee)
National Technical University of Athens
School of Civil Engineering

Associate Professor Vissarion Papadopoulos
(Member Advisor of the Committee)
National Technical University of Athens
School of Civil Engineering

Professor Christofer Provatidis
(Member of the Examination Committee)
National Technical University of Athens
School of Mechanical Engineering

Professor Charis Gantes
(Member of the Examination Committee)
National Technical University of Athens
School of Civil Engineering

Assistant Professor Michalis Fragiadakis
(Member of the Examination Committee)
National Technical University of Athens
School of Civil Engineering

Assistant Professor Savvas Triantafyllou
(Member of the Examination Committee)
National Technical University of Athens
School of Civil Engineering

To my parents.

Acknowledgments

I would like to thank everyone who contributed in any way to successful completion of this dissertation.

First, I would like to express my heartfelt thanks to Professor Manolis Papadrakakis, not only for the opportunity he gave me to deal with Isogeometric Analysis and research in National Technical University of Athens, but also for the supervision of this thesis.

Besides my advisor, I would like to thank all members of my thesis committee, Prof. K. Spiliopoulos, Assoc. Prof. V. Papadopoulos, for their insightful comments, time and support throughout the preparation and review of this dissertation.

This thesis owes a great deal to Dr. George Stavroulakis, who introduced me to the world of programming, that completely changed my perspective regarding the Civil Engineering profession and allowed to enter the world of computational mechanics. His guidance was pivotal in both academic and scientific matters, as well as subjects beyond the scope of this thesis.

In addition, I would like to express my sincere gratitude to all member of the MGroup and its scientific advisor Associate Professor Vissarion Papadopoulos. Our cooperation allowed me to delve into many different subjects of the computational mechanics field, as well as cultivate my programming skills that lead to the creation of MSolve open-source Finite Element Platform.

I gratefully acknowledge the funding received towards my Ph.D from the European Council Advanced Grant MASTER - *Mastering the computational challenges in numerical modeling and optimum design of CNT reinforced composites*, from Alexander S. Onassis Foundation for my Ph.D. studies scholarship and from the European Regional Development Fund and Greek national funds under the Grant HEAT - *Optimal multiscale design of innovative materials for heat exchange applications.* In addition, I would like to thank Ms. Stella Lioliousi-Tsoukatou for proofreading this thesis.

Last but not least, I would to thank my parents for always supporting in any way possible during all adversities of the past five years. Their continuous and unconditional support and love is the driving force that shows me the way through hard times.

Modelling of structures using the Isogeometric Method

Abstract

Isogeometric Analysis was proposed as an alternative spatial discretization method, that addresses the need for an integrated pipeline of Computer Aided Design and Computer Aided Engineering industries, towards the development of efficient and reliable structures. To this end, CAD shape functions ranging from Bézier to T-Splines are utilized, that allow the exact geometrical representation of arbitrarily complex geometries, while at the same time rendering the mesh generation procedure of other spatial discretization techniques such as FEM obsolete. Due to the high smoothness of the shape functions over conventional approaches, IGA has showcased significant advantages in various computational mechanics fields such as structural dynamics, fluid mechanics and optimization problems. In addition, the description of intricate geometries with highly continuous shape functions, enables IGA to be effectively used for shell theories such as Kirchhoff-Love thin shells, that required special treatment in the case of FEM. The latter resulted in the introduction of various shell formulations, for miscellaneous materials. Unfortunately, the strongest asset of isogeometric methods, which is the increased interelement continuity of the shape functions, constitutes at the same time its greatest weakness. Despite resulting in a smooth variation of the analysis characteristics and enhanced accuracy, a significant computational burden is added to the formation and solution of the resulting linear systems due to their increased bandwidth and reduced sparsity patterns. This rendering the introduction of efficient solution schemes a necessity for the establishment of isogeometric methods.

To this end, this dissertation introduces a family of methods, to address the efficient implementation of solution schemes for the purpose of isogeometric Galerkin and collocation methods. Specifically, for the case of isogeometric Galerkin method, an appropriate modification of the overlapping nature of NURBS shape functions is introduced, in the form of truncated shape functions. This modification generates a non-overlapping equivalent of the initial model that retains the same geometry, yet has reduced accuracy. As a result,with the aid of IETI domain decomposition method, applied to the interface among adjacent subdomains, the non-overlapping model can serve as an efficient preconditioner for the PCG iterative method. In a similar fashion, a non-overlapping decomposition of the non-symmetric matrices derived from isogeometric collocation methods is proposed, that allows the development of a GMRES preconditioner based on P-FETI-DP domain decomposition method. Finally, in order to unify all existing thin shell isogeometric formulations, a framework is proposed that examines isogeometric

shells under the prism of semi-concurrent multiscale analysis. To this end, a nested IGA-FEM analysis scheme is introduced, where macroscale shell modeling is performed with isogeometric Kirchhoff-Love shell elements, while Representative Volume Elements are discretized with solid finite elements. The required plane-stress constitutive law is then extracted from a homogenization process, thus leading to the introduction of a framework that can efficiently address composite materials of arbitrary microstructural topology.

ΠΕΡΙΛΗΨΗ ΤΗΣ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

με τίτλο

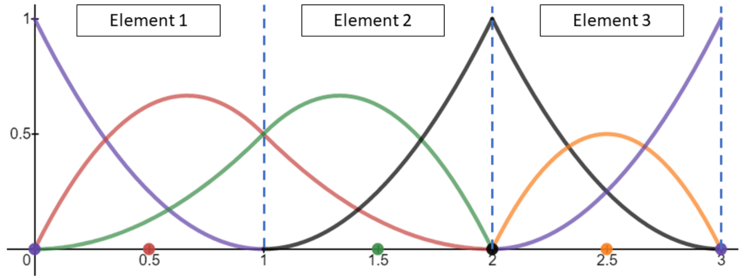
‘Προσομοίωση Κατασκευών με την Ισογεωμετρική Μέθοδο’

Κύριος σκοπός των υπολογιστικών μεθόδων είναι μελέτη και η ερμηνεία των φυσικών φαινομένων μέσω της μαθηματικής τους μοντελοποίσης. Η διαδικασία αυτή συχνά οδηγεί στη μετάφραση ενός φυσικού φαινομένου σε όρους μερικών διαφορικών εξισώσεων, η λύση των οποίων αναζητείται από μαθηματικούς και επιστήμονες. Η εύρεση αναλυτικής λύσης σε αυτά τα προβλήματα δεν ειναι πάντα εφικτή γεγονός που οδήγησε τους επιστήμονες στην ανάπτυξη υπολογιστικών μεθόδων. Μια από τις πλέον διαδεδομένες υπολογιστικές μεθόδους είναι αυτή των Πεπερασμένων Στοιχείων, η οποία διακριτοποιεί τον φορέα και προσφέρει μια προσεγγιστική λύση της διαφορικής εξίσωσης σε ένα πεπερασμένο πλήθος σημείων του. Κάθως όμως η πολυπλοκότητα των φορέων προς επίλυση αυξάνει, δημιουργείται η ανάγκη μιας στενότερης σύνδεσης της ανάλυσης με τη σχεδίαση του προσομοιώματος.

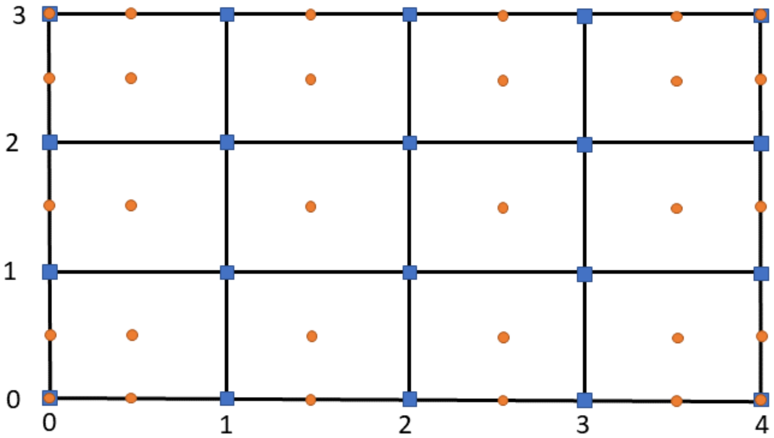
Η σχεδίαση με τη βοήθεια υπολογιστή (Computer Aided Design) είναι το επιστημονικό πεδίο που έχει ως βασικό στόχο τη διευκόλυνση της διαδικασίας δημιουργίας λεπτομερών σχεδίων, εκμεταλλευόμενη την τεχνολογική πρόοδο των υπολογιστών. Μια από τις πρώτες συνεισφορές σε αυτό το αντικείμενο προτάθηκε από τον Ferguson σαν μια εξέλιξη των πολυωνυμικών καμπυλών. Οι καμπύλες Ferguson αποτελούν μια ειδική κατηγορία πολυωνυμικών καμπυλών τρίτου βαθμού, οι οποίες υπολογίζονται δεδομένης της θέσης και της εφαπτομένης της καμπύλης, στο αρχικό και τελικό της σημείο. Η γενίκευση της πρότασης αυτής για οποιοδήποτε πολυωνυμικό βαθμό, οδήγησε στη δημιουργία των καμπυλών Hermite, όπου για τον το υπολογισμό μιας καμπύλης βαθμού $(2k+1)$, χρησιμποιούνται k τιμές σε κάθε άκρο της. Παρά της σαφή τους μαθηματική αναπαράσταση, οι πολυωνυμικές καμπύλες εγκαταλείφθηκαν νωρίς από τη σχεδιαστική κοινότητα αφού δεν θεωρούνται πρακτικές για τους σκοπούς της σχεδίασης. Το γεγονός αυτό αυτό οφείλεται στην μαθηματική φύση των υφιστάμενων αλγορίθμων για τον υπολογισμό τους, οι οποίοι επηρεάζονται σημαντικά από σφάλματα αποκοπής, ειδικά σε ακραίες τιμές των πολυωνυμικών συντελεστών. Με σκοπό να αμβλύνουν το παραπάνω μειονέκτημα των πολυωνυμικών καμπυλών, προτάθηκαν οι καμπύλες Bézier, οι οποίες θεωρούνται ισοδύναμες με τις πολυωνυμικές καμπύλες αλλά αντιμετωπίζουν τα μειονεκτήματα τους. Πλεόν το σχήμα της καμπύλης εξαρτάται μόνο από τα σημεία ελέγχου (control points), τα οποία είναι σημεία στον καρτεσιανό τρισδιάστατο χώρο. Η αλλαγή αυτή στη διαδικασία σχεδίασης της προσδίδει άμεσο φυσικό νόημα, ενώ οι μεταβολές στην καμπύλη υλοποιούνται ευκολότερα. Επιπρόσθετα, ο αριθμός των σημείων ελέγχου για τη δημιουργία μιας καμπύλης Bézier συνδέεται άμεσα με τον πολυωνυμικό της βαθμό και οδηγεί στη δημιουργία καμπυλών με αυξημένη ομαλότητα. Δυστυχώς, αυτό το πλεονέκτημα των καμπυλών Bézier, αποτελεί ταυτόχρονα και το μεγαλύτερο μειονέκτημα τους, καθώς η συνεχόμενη αύξηση του πολυωνυμικού βαθμού δημιουργεί αστάθειες των αλγορίθμων, ενώ η επιρροή κάθε σημείου ελέγχου σε ολόκληρο το μήκος της καμπύλης, δυσχαιρένει την τμηματική αλλαγή της. Για το λόγο αυτό, εισήχθησαν οι B-Splines, οι οποίες αποτελούν μια γενίκευση των καμπυλών Bézier ενός παραμετρικού διαστήματος σε περισσότερα. Η επέκταση αυτή απεμπλέκει τον αριθμό των σημείων ελέγχου από τον πολυωνυμικό βαθμό της καμπύλης και δίνει τη δυνατότητα στους σχεδιαστές να ελέγξουν μεμονομένα τμήματα της καμπύλης, ενώ η εισαγωγή κόμβων (knots) επιτρέπει απότομες αλλαγές στη συνέχεια της γεωμετρίας. Παρ' όλα αυτά, η αδυναμία της τεχνολογίας των B-Splines να αναπαραστήσει το σύνολο των κωνικών τομών, οδήγησε στην ανάγκη ανάπτυξης μιας νές τεχνολογίας σχεδίασης που θα λύνει το συγκεκριμένο πρόβλημα. Έτσι δημιουργήθηκαν οι NURBS. Η διαφορά τους με την προηγούμενη τεχνολογία των B-Splines, έγκειται στην εισαγωγή μιας τιμής βάρους σε κάθε σημείο ελέγχου της καμπύλης. Με τον τρόπο αυτό, δημιουργείται μια προβολή από τον χώρο $\mathbb{R}^{d+1}$ στον $\mathbb{R}^{d}$, όπου d οι διαστάσεις της γεωμετρίας, που επιτρέπει την αναπαράσταση πολυπλοκότερων σχεδίων και κωνικών τομών. Οι περιορισμοί των NURBS εμφανίζονται σε γεωμετρίες ανωτέρων διαστάσεων, όπως οι επιφάνειες και οι όγκοι καθώς η αναπαράσταση μιας γεωμετρίας στον παραμετρικό της χώρο, πρέπει να αντιστοιχεί σε ένα ευθύγραμμο τμήμα, ορθογώνιο ή κυβοειδές σχήμα, γεγονός που περιορίζει τις δυνατότητες αναπαράστασης εξαιρετικά πολύπλοκων γεωμετριών. Η πιο πρόσφατη τεχνολογία γεωμετρικής σχεδίασης που δημιουργήθηκε, είναι οι T-Splines, οι οποίες καταργούν την αναπαράσταση του παραμετρικού χώρου της γεωμετρίας ως τανυστικά γινόμενα μονοδιάστατων αξόνων όπως συνέβαινε στα NURBS, επιτρέποντας έτσι τη γεωμετρική απεικόνιση εξαιρετικά πολύπλοκων γεωμετριών και για πρώτη φορά την τοπική πύκνωση του δικτύου της γεωμετρίας.

Παράλληλα με την ανάπτυξη της τεχνολογίας σχεδιασμού με υπολογιστή (CAD), αναπτύχθηκε η επιστημονική περιοχή της ανάλυσης φορέων μηχανικού με υπολογιστή (CAE). Η δημιουργία γεωμετριών συνεχώς αυξανόμενης πολυπλοκότητας από τους σχεδιαστές, δημιουργεί το επιπρόσθετο βάρος στους μηχανικούς, της μετατροπής της γεωμετρικής πληροφορίας σε δεδομένα κατάλληλα για τα λογισμικά ανάλυσης. Αυτό το χάσμα μεταξύ σχεδίασης και ανάλυσης, δημιούργησε σταδιακά μια αυξανόμενη απόκλιση μεταξύ των δύο τεχνολογιών. Η Ισογεωμετρική Ανάλυση προτάθηκε ως μια τεχνολογία χωρικής διακριτοποίησης, η οποία αντιμετωπίζει την ανάγκη για μια κοινή εξέλιξη των τεχνολογιών σχεδίασης και ανάλυσης φορέων, με σκοπό να αμβλύνει τα προβλήματα χρήσης προσεγγιστικής γεωμετρίας κατά την ανάλυση. Αυτό το επιτυγχάνει με τη χρήση των συναρτήσεων σχεδίασης σαν κοινή βάση και για την ανάλυση των φορέων. Η ισογεωμετρική ανάλυση εμφανίζει πολλές ομοιότητες με την προκάτοχο τεχνολογία της, τα Πεπερασμένα Στοιχεία, καθώς και οι δυο αποτελούν ισοπαραμετρικές υλοποιήσεις της μεθόδου $Galerkin$ και διατηρούν παρόμοια ροή του κώδικα και ιδιότητες, όπως το εύρος διαγωνίου των παραγόμενων μητρώων. Από μια άλλη οπτική, πολλές πτυχές της μεθόδου των πεπερασμένων στοιχείων παύουν να ισχύουν. Για παράδειγμα, η τεχνική διακριτοποίησης της γεωμετρίας καταργείται, αφού πλέον χρησιμοποιείται η αρχική γεωμετρία για την ανάλυση, ενώ οι κόμβοι, που είναι τα σημεία ελέγχου, δεν ανήκουν πλέον κατά κανόνα πάνω στη γεωμετρία.

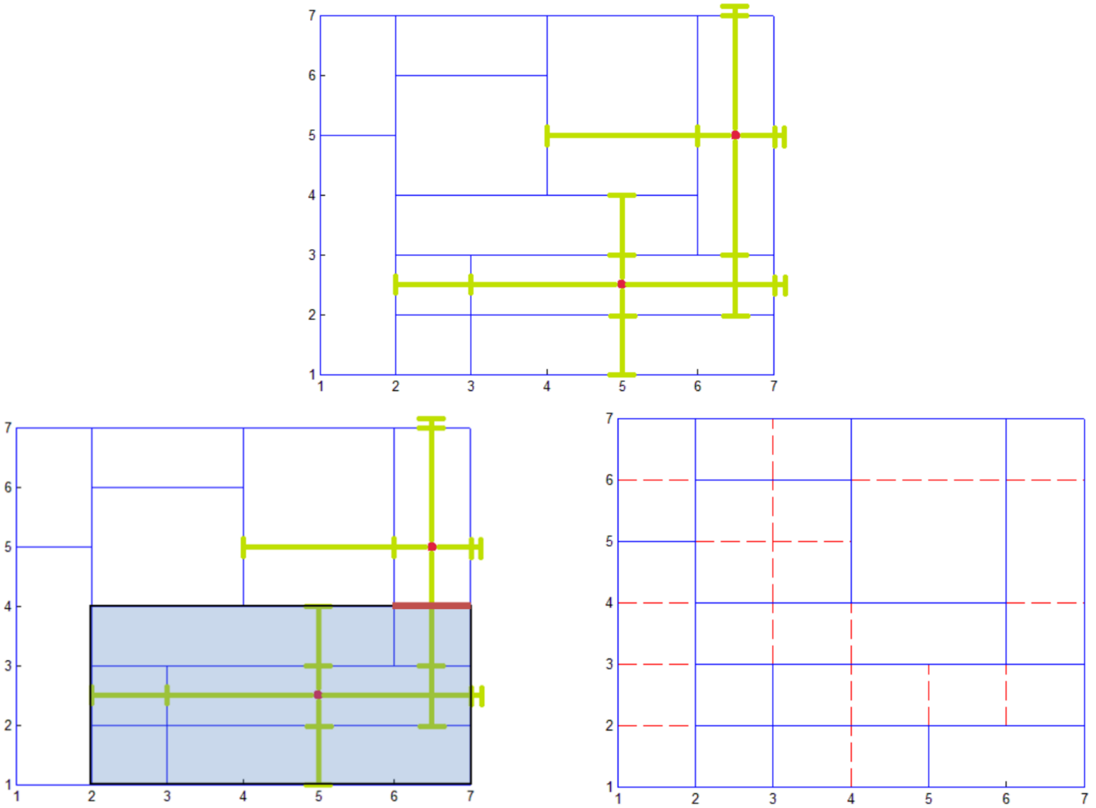
Για το λόγο αυτό, θα οριστούν εδώ οι βασικές ποσότητες ολοκλήρωσης που αντιστοιχούν με τα ισοπαραμετρικά πεπερασμένα στοιχεία της τεχνολογίας FEM. Στην περίπτωση των συναρτήσεων σχήματος $NURBS$, τα στοιχεία είναι παράγωγα του διανύσματος κόμβων ($knot\ vector$). Συγκεκριμένα στη μονοδιάστατη περίπτωση, τα στοιχεία ορίζονται ως τα μη μηδενικά διαστήματα μεταξύ διαδοχικών διακριτών τιμών του διανύσματος κόμβων. Ένα παράδειγμα διαχωρισμού ενός μονοδιάστατου άξονα σε στοιχεία απεικονίζεται στο Σχήμα 1. Στην περίπτωση δισδιάστατων ή τρισδιάστατων γεωμετριών τα στοιχεία προκύπτουν σε αυτήν την περίπτωση ως το τανυστικό γινόμενο μεταξύ των στοιχείων που έχουν δημιουργεί για κάθε ένα από τους παραμετρικούς άξονες ξ,η,ζ του παραμετρικού χώρου του φορέα όπως φαίνεται στο Σχήμα 2.



Σχήμα 1: Ισογεωμετρικά στοιχεία για μονοπαραμετρικό άξονα που ορίζεται από το διάνυσμα κόμβων $\Xi = \{0, 0, 0, 1, 2, 2, 3, 3, 3\}$.

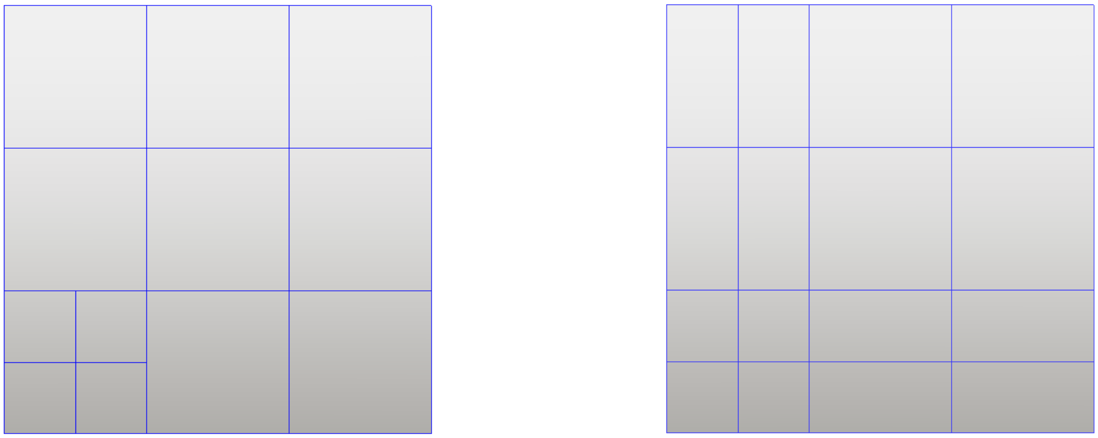


Σχήμα 2: Ισογεωμετρικά στοιχεία για δισδιάστατο χωρίο που ορίζεται από το διάνυσμα κόμβων $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 4\}$ $H = \{0, 0, 0, 1, 2, 3, 3, 3\}$.



Σχήμα 3: Διαδικασία υπολογισμού των στοιχείων ολοκλήρωσης για συναρτήσεις σχήματος $T - Splines$.

Σε αντίθεση με την τεχνολογία σχεδίασης $NURBS$, στην περίπτωση της τεχνολογίας $T - Splines$ η εύρεση των στοιχείων ολοκλήρωσης ειναι αρκετά πιο πολύπλοκη διαδικασία. Ξεκινά με την εύρεση των παραμετρικών θέσεων των σημείων ελέγχου που ονομάζονται *anchors*, στον χώρο δεικτών ή *index*. Ο χώρος αυτός είναι βοηθητικός χώρος της Ισογεωμετρικής Ανάλυσης,ο οποίος παρουσιάζει τις τιμές του διανύσματος κόμβων σε ίσες αποστάσεις, ανεξάρτητα αν διαδοχικές τιμές συμπίπτουν. Τα *anchors* τοποθετούνται είτε στο κέντρο είτε στις ακμές των πολυγώνων που σχηματίζονται στο χώρο δεικτών, ανάλογα με τον πολυωνυμικό βαθμό των συναρτήσεων σχήματος. Στη συνέχεια λόγω απουσίας του καθολικού διανύσματος κόμβων, υπολογίζονται τα τοπικά διανύσματα κόμβων κάθε σημείου ελέγχου. Παράδειγμα των τοπικών διανυσμάτων παρουσιάζεται στο Σχήμα 3α. Το τανυστικό γινόμενο αυτών των τοπικών διανυσμάτων κόμβων ορίζει την επιφάνεια επιρροής κάθε συνάρτησης, όπως η σκιαγραφημένη επιφάνεια που απεικονίζεται στο Σχήμα 3β . Όπως ειναι φανερό, η επιφάνεια επιρροής ορίζει γραμμές στον χώρο *index* που δεν υπήρχαν στο αρχικό δίκτυο. Αυτές οι γραμμές αποκαλούνται γραμμές μειωμένης συνέχειας, καθώς στο όριο τους ορίζεται η σύνδεση των τμημάτων πολυωνυμικών συναρτήσεων της τεχνολογίας $T - Splines$. Το σύνολο αυτών των γραμμών, ορίζει μια νέα διαμέριση σε ορθογωνικά χωρία, του χώρου *index* που είναι και τα ζητούμενα στοιχεία ολοκλήρωσης.



Σχήμα 4: Εξαγωγή δικτύου στοιχείων *Bezier* απο δίκτυο $T - Splines$.

Τόσο η τεχνολογία των $T - Splines$ όσο και αυτή των *NURBS*, αποτελούν μια γενίκευση των συναρτήσεων σχεδιασμού *Bezier*. Για το λόγο αυτό, πολλές φορές κατά την ανάλυση φορέων με την ισογεωμετρική μέθοδο, επιλέγεται η αναγωγή των στοιχείων ολοκλήρωσης πολύπλοκων συναρτήσεων σε συναρτήσεις *Bezier*. Οι συναρτήσεις *Bezier* απεικονίζουν τον τυχαίο παραμετρικό χώρο των συναρτήσεων $NURBS$,$T - Splines$ στον ίδιο παραμετρικό χωρίο $[-1, 1]$ των πεπερασμένων στοιχείων, ενώ οι συναρτήσεις σχήματος είναι ίδιες σε όλα τα στοιχεία ολοκλήρωσης και μπορούν να επαναχρησιμοποιηθούν για τον υπολογισμό του μητρώου στιβαρότητας των φορέων. Στο σχήμα 4 παρουσιάζεται ένα παράδειγμα εξαγωγής στοιχείων *Bezier* από ένα πλέγμα συναρτήσεων $T - Splines$.

Η ανάλυση φορέων με την ισογεωμετρική μέθοδο *Galerkin*, που χρησιμοποιεί σαν ποσότητες ολοκλήρωσης τα στοιχεία που περιγράφηκαν προηγουμένως, έχει αποδειχθεί πως παράγει αποτελέσματα, με σημαντικά αυξημένη ακρίβεια ανά βαθμό ελευθερίας, σε σχέση με τη μέθοδο των πεπερασμένων στοιχείων. Αυτό οφείλεται κυρίως στην αυξημένη συνέχεια των συναρτήσεων σχήματος, η οποία σε συνδυασμό με το μηδενισμό του γεωμετρικού σφάλματος οδηγεί σε αυξημένη ομαλότητα των χαρακτηριστικών δεύτερης τάξης, όπως οι τάσεις και παραμορφώσεις. Δυστυχώς, αυτό το πλεονέκτημα της ισογεωμετρικής ανάλυσης συνοδεύεται από ένα αυξημένο κόστος για τη μόρφωση των μητρώων, αφού τα απαιτούμενα σημεία ολοκλήρωσης είναι τάξης μεγέθους περισσότερα σε σχέση με τα πεπερασμένα στοιχεία. Έτσι η επιστημονική κοινότητα, έστρεψε τις προσπάθειες της στην ανάπτυξη εναλλακτικών τρόπων αριθμητικής ολοκλήρωσης που να ελαχιστοποιούν το κόστος μόρφωσης των μητρώων. Αυτό οδήγησε στη χρήση της μεθόδου *collocation* για την ανάλυση προβλημάτων συνοριακών τιμών στα πλαίσια της ισογεωμετρικής ανάλυσης. Η μέθοδος *collocation* μειώνει δραστικά τον αριθμό των απαιτούμενων σημείων ολοκλήρωσης, που είναι πλέον ίσος με τον αριθμό σημείων ελέγχου του φορέα. Η βασική διαφορά της μεθόδου *collocation* σε σχέση με τη *Galerkin*, είναι οτι απαιτείται η ικανοποίηση της ισχυρής μορφής της διαφορικής εξίσωσης σε περιορισμένο πλήθος σημείων. Έτσι οι εξισώσεις της γραμμικής ελαστικότητας που παρουσιάζονται στην εξίσωση1,

$$\nabla \cdot (C\nabla^S \mathbf{u}) + \mathbf{f} = 0 \qquad in\ \Omega \tag{1α'}$$

$$\mathbf{u} = \mathbf{g} \qquad on\ \Gamma_D \tag{1β'}$$

$$(C\nabla^S \mathbf{u}) \cdot \mathbf{n} = \mathbf{h} \qquad on\ \Gamma_N \tag{1γ'}$$

μπορούν να γραφούν σε μια εξίσωση ως εξής

$$\int_\Omega (C\nabla^S \mathbf{u}) : \nabla^S \mathbf{w} d\Omega = \int_\Omega \mathbf{f} \cdot \mathbf{w} d\Omega + \int_{\Gamma_N} h \cdot \mathbf{w} d\Gamma \tag{2}$$

ενώ και στις δυο μεθόδους *collocation* και *Galerkin*, ο όρος u αφορά την παρεμβολή του πεδίου των μετακινήσεων με χρήση των συναρτήσεων σχήματος, στην *collocation* ο όρος w αφορά τις συναρτήσεις δοκιμής (*test functions*), οι οποίες επιλέγονται να αντικατασταθούν από τη συνάρτηση *Dirac delta*.

$$\int_{\Omega}[\nabla\cdot(C\nabla^{S}\mathbf{u})+\mathbf{f}]\cdot\mathbf{w}d\Omega-\int_{\Gamma_N}[(C\nabla^{S}\mathbf{u})\cdot\mathbf{n}-\mathbf{h}]\cdot\mathbf{w}d\Gamma=0 \tag{3}$$

Μετά από ανάπτυξη της τελευταίας σχέσης, καταλήγουμε στον υπολογισμό του μητρώου στιβαρότητας της *collocation* μέσω της σχέσης

$$K_{ij}=\begin{cases} L(N_j(\bar{\xi}_{ij})), & for\ 1\leq i\leq k,\\ \mathbf{n}_i\cdot C\nabla N_j(\bar{\xi}_{ij}), & for\ k+1\leq i\leq n \end{cases} \tag{4}$$

και του αντίστοιχου διανύσματος εξωτερικών δράσεων αντίστοιχα:

$$f_i=\begin{cases} -L(\tilde{u}_D(\bar{\xi}_{ij}))+f(\bar{\xi}_{ij}), & for\ 1\leq i\leq k\\ -\mathbf{n}_i\cdot C\nabla u_D(\bar{\xi}_{ij})_i+h(\bar{\xi}_{ij}), & for\ k+1\leq i\leq n \end{cases} \tag{5}$$

όπου οι τελεστές L, B στη δισδιάσταση ελαστικότητα υπολογίζονται ως εξής:

$$L=\begin{pmatrix} (\lambda+2\mu)\frac{\partial^2}{\partial x^2}+\mu\frac{\partial^2}{\partial y^2} & (\lambda+\mu)\frac{\partial^2}{\partial x\partial y}\\ (\lambda+\mu)\frac{\partial^2}{\partial x\partial y} & (\lambda+2\mu)\frac{\partial^2}{\partial y^2}+\mu\frac{\partial^2}{\partial x^2} \end{pmatrix} \tag{6}$$

$$B=\begin{pmatrix} (\lambda+2\mu)n_x\frac{\partial}{\partial x}+\mu n_y\frac{\partial}{\partial y} & \lambda n_x\frac{\partial}{\partial y}+\mu n_y\frac{\partial}{\partial x}\\ \lambda n_y\frac{\partial}{\partial x}+\mu n_x\frac{\partial}{\partial y} & (\lambda+2\mu)n_y\frac{\partial}{\partial y}+\mu n_x\frac{\partial}{\partial x} \end{pmatrix} \tag{7}$$

Μετά την αντιμετώπιση του ζητήματος της αποδοτικής μόρφωσης των μητρώων στιβαρότητας, ένα δεύτερο ζήτημα που εγείρεται αφορά την αποδοτική επίλυση των γραμμικών συστημάτων που προκύπτουν. Σε αντίθεση με τη μέθοδο των πεπερασμένων στοιχείων, η οποία μετά από πολλές δεκαετίες έρευνας έχει αναπτύξει κατάλληλους επιλύτες για κάθε είδους πρόβλημα, η Ισογεωμετρική Ανάλυση είναι μια πρόσφατη τεχνολογία, στην οποία κάθε τεχνική και αλγόριθμος πρέπει να εφαρμοστεί και να ερευνηθεί από την αρχή. Έτσι τα κυριότερα είδη επιλυτών γραμμικών εξισώσεων, όπως οι άμεσοι, οι επαναληπτικοί καθώς και οι επιλύτες με βάση τη μέθοδο των υποφορέων πρέπει να ελεγχθούν για την επάρκειά τους. Συγκεκριμένα:

Άμεσοι Επιλύτες:

Είναι επιλύτες που βασίζονται κυρίως σε μεθόδους παραγοντοποίησης του μητρώου στιβαρότητας. Ο πιο γνωστός αλγόριθμος παραγοντοποίησης για συμμετρικούς πίνακες, όπως αυτοί που παράγονται από τη μέθοδο των πεπερασμένων στοιχείων ή την ισογεομετρική μέθοδο *Galerkin*, είναι η παραγοντοποίηση *Cholesky*, η οποία μετατρέπει το μητρώο στιβαρότητας σε γινόμενο άνω και κάτω τριγωνικού πίνακα. Αντίστοιχα, στην περίπτωση μη-συμμετρικών πινάκων, όπως αυτοί που προκύπτουν από την ισογεωμετρική μέθοδο *collocation*, μπορεί να εφαρμοστεί ο αλγόριθμος παραγοντοποίησης *LU*. Οι άμεσοι επιλύτες είναι ιδιαίτερα αποδοτικοί στην περίπτωση που χρειάζεται επίλυση συστήματος με πολλαπλά δεξιά μέλη. Μέχρι 100 χιλιάδες βαθμούς ελευθερίας θεωρούνται πιο αποδοτικοί, αλλά το μεγαλύτερο μειονέκτημα τους είναι η μεγάλη απαίτηση τους σε μνήμη, που τους καθιστά αναποτελεσματικούς για φορείς μεγάλης κλίμακας. Κατά την εφαρμογή τους στην Ισογεωμετρική Ανάλυση παρουσιάζουν αυξημένο υπολογιστικό κόστος, εξαιτίας της αύξησης του εύρους διαγωνίου, καθώς και της αύξησης του πλήθους των μη-μηδενικών στοιχείων.

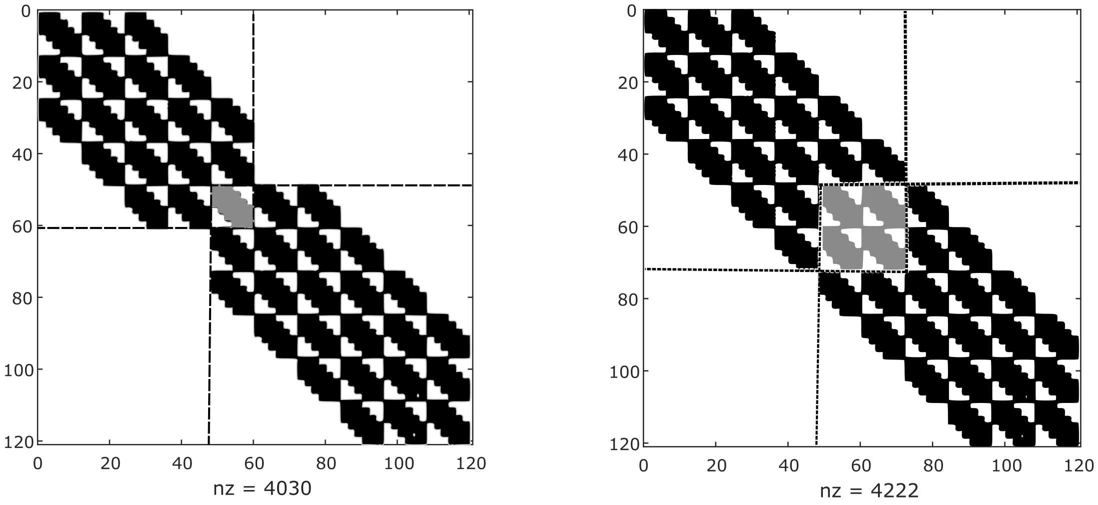
Επαναληπτικοί Επιλύτες:

Δεύτερη σημαντική κατηγορία επιλυτών είναι οι επαναληπτικοί επιλύτες. Στόχος τους είναι να προσεγγίσουν την τελική λύση μέσα από επαναλήψεις. Όταν ένα συγκεκριμένο κριτήριο ακρίβειας ικανοποιηθεί, τότε θεωρούμε ότι έχουμε σύγκλιση στην τελική λύση. Ένας από τους πιο γνωστούς αλγόριθμους επαναληπτικής επίλυσης συστημάτων με συμμετρικά, θετικά ορισμένα μητρώα, είναι ο αλγόριθμος των προσταθεροποιημένων συζυγών διανυσματικών κλίσεων (Preconditioned Conjegate Gradient).Αντίστοιχα, στην περίπτωση μη-συμμετρικών μητρώων είναι ευρέως διαδεδομένος ο αλγόριθμος γενικευμένης ελαχιστοποιήσης υπολοίπου (Generalized Minimal Residual). Η ταχύτητα σύγκλισης των επαναληπτικών επιλυτών εξαρτάται άμεσα από τον προσταθεροποιητή που θα χρησιμοποιηθεί, οποίος έχει στόχο να εκτιμήσει τη λύση του επόμενο επαναληπτικού βήματος με μικρό υπολογιστικό κόστος. Η εφαρμογή τους στην Ισογεωμετρική Ανάλυση αποδεικνύεται, όπως και στην περίπτωση των άμεσων επιλυτών, να επηρεάζεται σημαντικά από την αυξημένη συνέχεια δίνοντας εως και $2p$ μεγαλύτερους χρόνους σε σχέση με μια επίλυση με την ελάχιστη δυνατή συνέχεια, όπου p ο πολυωνυμικός βαθμός των συναρτήσεων σχήματος.

Μέθοδοι επίλυσης με υποφορείς:

Τελευταία κατηγορία αλγορίθμων επίλυσης γραμμικών συστημάτων που μελετήθηκαν είναι οι μέθοδοι επίλυσης με υποφορείς. Οι αλγόριθμοι αυτοί βασίζονται στον διαχωρισμό ενός φορέα σε μικρότερα τμήματα που ονομάζονται υποφορείς. Στην περίπτωση φορέων ισογεωμετρικής ανάλυσης, που έχουν σχεδιαστεί με την τεχνολογία σχεδίασης NURBS, ο φορέας είναι φυσικά διακριτοποιημένος σε επιμέρους τμήματα που ονομάζονται patches. Η δημιουργία των patches κατά τη διάρκεια της σχεδίασης, οφείλεται στην απαίτηση της τεχνολογίας σχεδίασης NURBS τα φυσικά χωρία που περιγράφει να απεικονίζονται στον παραμετρικό χώρο ως ευθύγραμμα τμήματα, ορθογώνια ή κύβοι. Εκμεταλλευόμενη το φυσικό αυτό χωρισμό, προτάθηκε η τεχνολογία $IETI$, η οποία αποτελεί μια ισογεωμετρική προσέγγιση της δυϊκής-πρωτογενούς μεθόδου επίλυσης φορέων ($FETI-DP$) για φορείς διακριτοποιμένους με την ισογεωμετρική μέθοδο *Galerkin*. Η μέθοδος αυτή μετατρέπει το συνολικό γραμμικό σύστημα, στο ενδοσυνοριακό πρόβλημα μεταξύ των υποφορέων, μειώνοντας σημαντικά το μέγεθος του τελικού συστήματος προς επίλυση. Παρόλα αυτά, η συγκεκριμένη μέθοδος παρουσιάζει ένα σημαντικό μειονέκτημα. Η δημιουργία των υποφορέων εξαρτάται από τις απαιτήσεις τη σχεδίασης, με αποτέλεσμα οι υποφορείς να δημιουργούνται με αυθαίρετο τρόπο και μέγεθος. Αυτό έχει σαν αποτέλεσμα, μια ανισοκατανομή του υπολογιστικού φόρτου μεταξύ των υποφορέων κατά την εφαρμογή της μεθόδου.

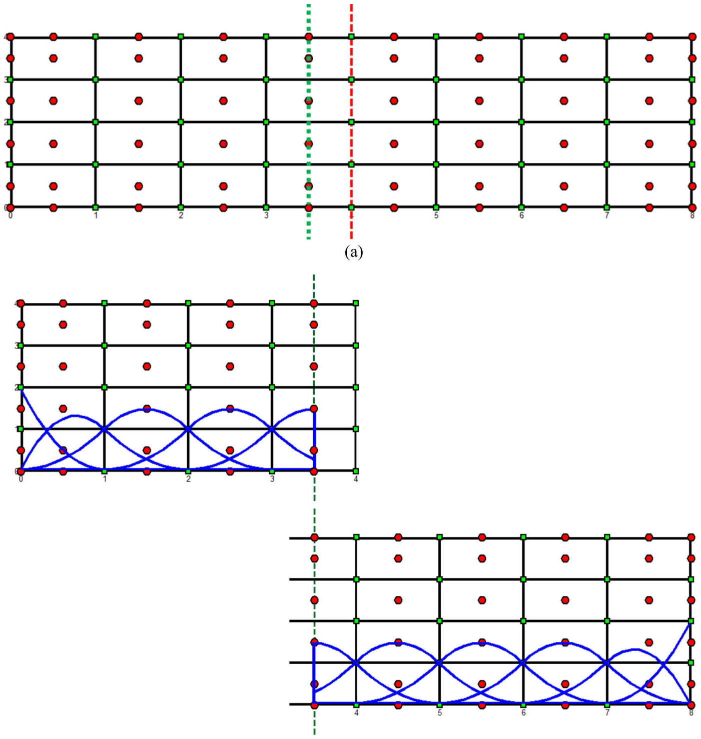
Στα πλαίσια αυτής της διατρίβης προτείνονται και υλοποιούνται εναλλακτικοί τρόποι διαχωρισμού του φορέα σε υποφορείς, που αμβλύνουν το μειονέκτημα της μεθόδου $IETI$. Η πρώτη εναλλακτική ονομάστηκε $IETI - P$ και αποτελεί μια επέκταση της υφιστάμενης μεθόδου $IETI$, η οποία χωρίζει τα ανισομεγέθη τμήματα σε περισσότερους υποφορείς. Αυτό επιτυγχάνεται μέσω της μείωσης της συνέχειας στους κόμβους ($knots$) των παραμετρικών αξόνων. Όταν η συνέχεια ενός κόμβου μειωθεί, μέσω της αύξησης της πολλαπλότητας του σε C^0, τότε ο αρχικός φορέας μπορεί να διασπαστεί στο σημείο του κόμβου σε δυο υποφορείς. Αυτός ο νέος διαχωρισμός, έχει θετική επίδραση στην επίλυση ενός φορέα με τη μέθοδο της IETI, καθώς απαλείφει την ανισοκατανομή του υπολογιστικού φορτίου, εισάγωντας όμως επιπλέον κόστος λόγω εισαγωγής νέων κόμβων. Η δεύτερη εναλλακτική προσέγγιση που προτάθηκε για το χωρισμό υποφορέων, είναι η μέθοδος $IETI - O$. Στην περίπτωση αυτή δεν απαιτείται μείωση της συνέχειας στους κόμβους, αλλά προκύπτει από τον αρχικό φορέα θεωρώντας μια διευρυμένη διεπιφάνεια μεταξύ των νέων υποφορέων. Όπως ακριβώς η $IETI - P$, έτσι και η μέθοδος $IETI - O$ καταφέρνει να αμβλύνει το μειονέκτημα της μεθόδου $IETI$. Το μειονέκτημα της $IETI - O$ είναι ότι αυξάνει σημαντικά το ενδοσυνοριακό πρόβλημα μεταξύ των υποφορέων, αυξάνοντας σημαντικά το κόστος επίλυσης του ενδοσυνοριακού προβλήματος. Η διαφορά των δύο μεθόδων $IETI - P$ και $IETI - O$ όσον αφορά το μέγεθος του ενδοσυνοριακού προβλήματος απεικονίζεται στο σχήμα 5, όπου φαίνεται η σημαντική διαφορά μεταξύ των δύο μεθοδολογιών.



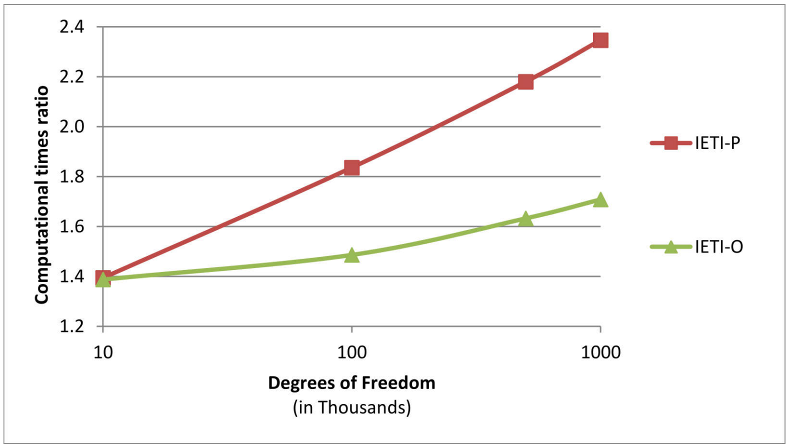
Σχήμα 5: Αλληλεπικαλυπτόμενες και μη μέθοδοι χωριμού υποφορέων

Παρά τις ευνοϊκές τους ιδιότητες, οι μέθοδοι $IETI - P$ και $IETI - O$, δημιοργούν νέα μειονεκτήματα, όπως η αύξηση του μεγέθους του ενδοσυνοριακού προβλήματος, που επιβαρύνουν σημαντικά την επίλυση του τελικού προβλήματος. Για το λόγο αυτό εισάγεται μια ακόμα μεθοδολογία επίλυσης με υποφορείς, η οποία συνδυάζει τα πλεονεκτήματα μιας επαναληπτικής μεθόδου όπως η PCG, με μεθόδους υποφορέων, η οποία ονομάστηκε $PCG - IETI - N$. Συγκεκριμένα, η μέθοδος αυτή εισάγει ένα προσομοίωμα που έχει την ίδια γεωμετρία και ίδιες ιδιότητες με το αρχικό, αλλά οι συναρτήσεις σχήματος έχουν αποκοπεί στη διεπιφάνεια μεταξύ υποφορέων. Αυτές οι ασυνέχειες έχουν σαν αποτέλεσμα τη μειωμένη ακρίβεια του νέου προσομοιώματος. Το νέο αυτό προσομοίωμα δεν μπορεί να χρησιμοποιηθεί για την επίλυση του συνολικού φορέα, μπορεί όμως να αποτελέσει έναν αποδοτικό προσταθεροποιητή για μια επαναληπτική μέθοδο επίλυσης όπως η PCG. Το σχήμα 6 απεικονίζει γραφικά τη μεθοδολογία αποκοπής των συναρτήσεων σχήματος, όπου η κόκκινη διακεκκομένη γραμμή αποτελεί ένα παράδειγμα σημείου χωρισμού των συναρτήσεων για άρτιο πολυωνυμικό βαθμό, ενώ η πράσινη αντίστοιχα για περιττό.

Παραμετρικές διερευνήσεις πραγματοποιήθηκαν, για να αποτιμήσουν την αποτελεσματικότητα της προτεινόμενης μεθόδου $PCG - IETI - N$, τόσο για διαφορετικούς πολυωνυμικούς βαθμούς όσο και για ποικίλες διακριτοποιήσεις φορέων. Τα αποτελέσματα έδειξαν μικρή επιρροή των παραπάνω χαρακτηριστικών στον αριθμό επαναλήψεων που απαιτούνται για τη σύγκλιση της μεθόδου. Ταυτόχρονα, έγινε σύγκριση της προτεινόμενης μεθόδου με τους δύο εναλλακτικούς τρόπους χωρισμού σε υποφορείς, $IETI-P$ και $IETI-O$. Οι φορείς που μελετήθηκαν κυμαίνονται από 100 χιλιάδες βαθμούς ελευθερίας έως 1 εκατομμύριο και αριθμό υποφορέων από 9 εως 4.800. Τα αποτελέσματα της σύγκρισης παρουσιάζονται γραφικά στο σχήμα 7 και δείχνουν την επιτάχυνση της μεθόδου $PCG - IETI - N$ σε σχέση με τις $IETI - P$ και $IETI - O$. Όπως φαίνεται και από το σχήμα, η προτεινόμενη μέθοδος παρουσιάζει μια επιτάχυνση $1,7x$ σε σχέση με την IETI-O και $2,3x$ σε σύγκριση με την $IETI - P$, με αυξητική τάση καθώς ο αριθμός των βαθμών ελευθερίας μεγαλώνει. Τα προηγούμενα αποτελέσματα καθιστούν την μέθοδο $PCG - IETI - N$, ιδανική για την επίλυση γραμμικών συστημάτων που προκύπτουν από την ισογεωμετρική μέθοδο με θεώρηση $Galerkin$, καθώς επιτρέπει τον τυχαίο χωρισμό του φορέα σε υποφορείς, επομένως καταφέρνει να εξασφαλίσει την ομοιόμορφη κατανομή του υπολογιστικού φορτίου μεταξύ των υποφορέων και συνεπώς σύμφωνα και με τα αριθμητικά αποτελέσματα προσφέρει μια σημαντική βελτίωση τόσο σε όρους επαναλήψεων όσο και υπολογιστικού χρόνου σε σχέση με υπάρχουσες μεθόδους επίλυσης.



Σχήμα 6: Μείωση συνέχειας συναρτήσεων σχήματος με σκοπό το χωρισμό φορέα σε μη αλληλεπικαλυπτόμενους υποφορείς.

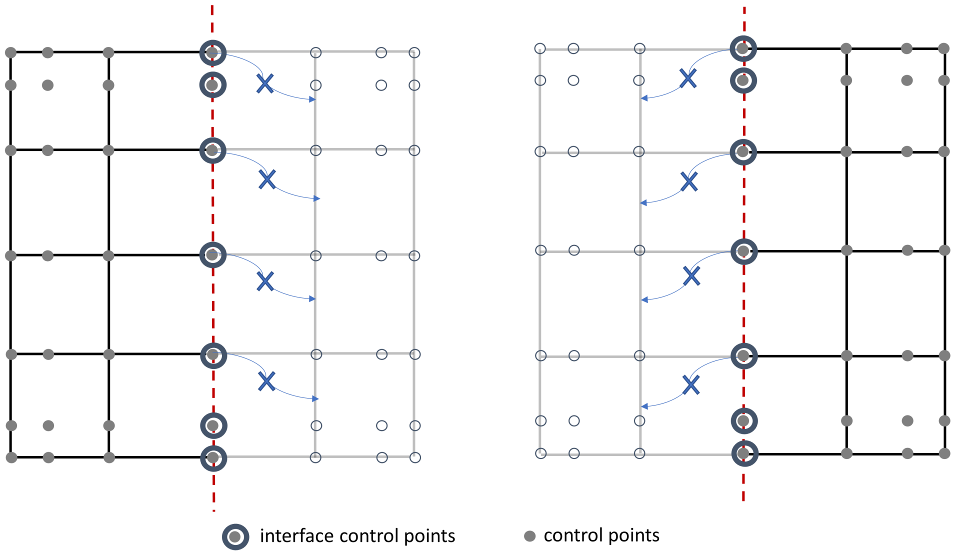


Σχήμα 7: Βελτίωση επιδώσης μεθόδος $PCG-IETI-N$ σε σχέση με προηγούμενες μεθόδους.

Η μελέτη για την επίλυση συστημάτων με υποφορείς επεκτείνεται στη συνέχεια της διατριβής και στην περίπτωση μη-συμμετρικών συστημάτων που πηγάζουν από την μέθοδο *collocation*. Πάρα το μειωμένο της κόστος κατά τη φάση μόρφωσης των μητρώων, η μέθοδος *collocation* μετατοπίζει σημαντικό κομμάτι του υπολογιστικού κόστους στην επίλυση των παραγόμενων γραμμικών συστημάτων καθώς τα μητρώα είναι μη συμμετρικά. Η υπάρχουσα βιβλιογραφία επικεντρώνεται στη δημιουργία αποτελεσματικών προσταθεροποιητών για την επιτάχυνση της μεθόδου *GMRES*. Στο πλαίσιο αυτό, η μοναδική συμβολή με τη χρήση υποφορέων, αποδίδεται στην δημιουργία ενός προσταθεροποιητή με τη μέθοδο *overlapping additive Schwarz* (*OAS*). Παρά τη ευνοϊκές δυνατότητές της, η μέθοδος αυτή παρουσιάζει ένα σημαντικό μειονέκτημα, που είναι το αυξημένο εύρος της ζώνης αλληλεπικάλυψης μεταξύ των υποφορέων. Το εύρος αυτό μπορεί να γίνει απαγορευτικά μεγάλο στην περίπτωση αυξημένου πολυωνυμικού βαθμού των συναρτήσεων σχήματος, επιβαρύνοντας έτσι σημαντικά το απαιτούμενο υπολογιστικό κόστος για την επίλυση.

Στα πλαίσια αυτής της διατριβής μελετήθηκαν τόσο πρωτογενείς όσο δυϊκές μέθοδοι επίλυσης με υποφορείς με στόχο να αντιμετωπίσουν τα μειονεκτήματα της *OAS* και να εισάγουν ένα αποτελεσματικότερο προσταθεροποιητή για τη μέθοδο *GMRES*. Οι πρωτογενείς μέθοδοι όπως για παράδειγμα η κύρια συνοριακή μέθοδος (*Primal Substructuring Method*), εισάγουν ένα ενδοσυνοριακό πρόβλημα μεταξύ των υποφορέων , του οποίου οι άγνωστες ποσότητες είναι οι μετακινήσεις των συνοριακών κόμβων. Αντίθετα, στις δυϊκές μεθόδους, οι άγνωστες ποσότητες του ενδοσυνοριακού προβλήματος είναι τα δυϊκά μεγέθη των μετακινήσεων, δηλαδή οι δυνάμεις αλληλεπίδρασης μεταξύ των ενδοσυνοριακών κόμβων. Για να γίνει εφικτή η χρήση των υφιστάμενων μεθόδων επίλυσης με υποφορείς που είναι γνωστές από τη μέθοδο των πεπερασμένων στοιχείων, πρέπει αρχικά να δημιουργειθεί ένα μη αλληλεπικαλυπτόμενο προσομοίωμα, μέσω της κατάργησης της αλληλεπικάλυψης των συναρτήσεων σχήματος. Ένα παράδειγμα μείωσης της συνέχειας μεταξύ των υποφορέων απεικονίζεται στο Σχήμα 8.

Με δεδομένη την ανωτέρω διαμέριση του φορέα σε μη αλληλεπικαλυπτόμενους υποφορε-



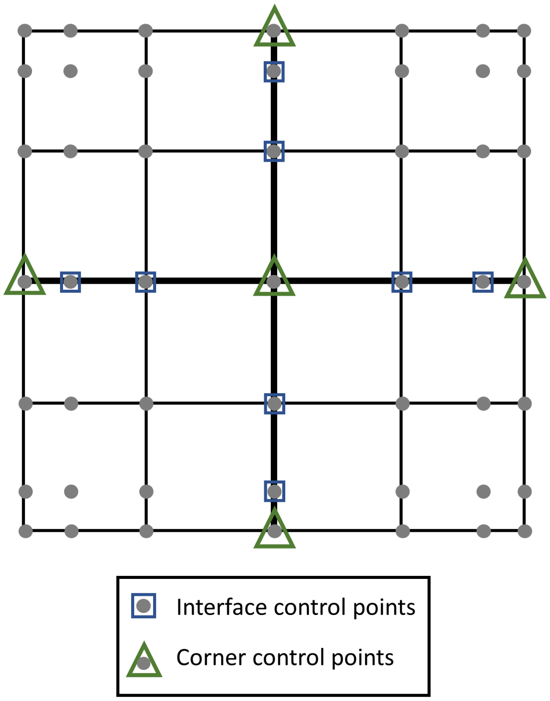
Σχήμα 8: Μείωση αλληλεπικάλυψης μεταξύ υποφορέων.

ίς, η αναζήτηση της καταλληλότερης μεθόδου προσταθεροποίησης ξεκίνησε με την μέθοδο $IETI$, που αποτελεί την ισογεωμετρική διατύπωση της μεθόδου $FETI-DP$. Τα αποτελέσματα της υλοποίησης αυτής ήταν εσφαλμένα, καθώς μερικοί φορείς συνέκλιναν σε λάθος λύση και σε άλλες περιπτώσεις δεν ήταν δυνατή η σύγκλιση του αλγορίθμου. Η συμπεριφορά αυτή μπορεί να ερμηνευτεί εξετάζοντας τη μηχανική αναλογία ενός όρου του μητρώου στιβαρότητας. Συγκεκριμένα, στην περίπτωση συμμετρικών μητρώων στιβαρότητας, όπως στην ισογεωμετρική μέθοδο $Galerkin$, ο όρος $\mathbf{K}_{ij}$ εκφράζει τη δύναμε που πρέπει να ασκηθεί στο βαθμό ελευθερίας i προκειμένου να αναπτυχθεί μοναδιαία μετατόπιση στον κόμβο j. Στην περίπτωση όμως μη-συμμετρικών μητρώων όπως στη μέθοδο *collocation*, αυτή η μηχανική αναλογία παύει να ισχύει καθώς οι γραμμές του μητρώου στιβαρότητας, αντιπροσωπεύουν σημεία ολοκλήρωσης της μεθόδου, ενώ οι στήλες τους βαθμούς ελευθερίας των κόμβων με αποτέλεσμα να πάψουν να υφίστανται και τα δυϊκά μεγέθη που αντιπροσωπεύονται από τις δυνάμεις αλληλεπίδρασης.

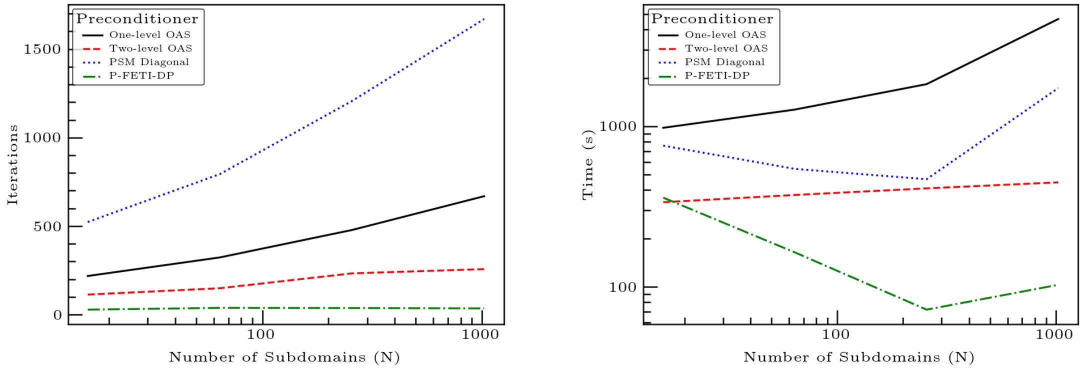
Για το λόγο αυτό, η μέθοδος $FETI - DP$ θεωρήθηκε αναποτελεσματική και η διερεύνηση συνεχίστηκε με τις πρωτογενείς μεθόδους. Η PSM εξετάστηκε πρώτη, με επιτυχή αποτελέσματα. Η εφαρμογή της σαν προσταθεροποιητής της $GMRES$ για την επίλυση του καθολικού προβλήματος, μείωσε δραστικά τις επαναλήψεις. Κάθως όμως το μέγεθος του ενδοσυνοριακού προβλήματος αυξάνει, αυξάνουν ταυτόχρονα και οι αντίστοιχες επαναλήψεις για την επίλυση του. Για το λόγο αυτό εξεταστηκε η μέθοδος $P - FETI - DP$ ώς προσταθεροποιητής της καθολικής $GMRES$. Πέρα από το καθολικό ενδοσυνοριακό πρόβλημα η μέθοδος αυτή εισάγει και το αδρό (*coarse*) πρόβλημα, το οποίο χρησιμοποιείται για την επιτάχυνση της λύσης του ενδοσυνοριακού προβλήματος. Συγκεκριμένα, οι κόμβοι του αδρού προβλήματος επιλέγονται ως τα σημεία ελέγχου που ανήκουν σε περισσότερους από δυο υποφορείς ή σε 2 υποφορείς αλλά είναι ταυτόχρονα και σημεία ελέγχου του συνόρου του φορέα. Η διαδικασία χωρισμού του φορέα σε υποφορείς και η επιλογή των ενδοσυνοριακών και γωνιακών κόμβων του αδρού προβλήματος απεικονίζεται στο Σχήμα 9.

Τα συγκριτικά απότελέσματα της χρήσης του προσταθεροποιητή $P - FETI - DP$ σε σχέση με τις υπάρχουσες εναλλακτικές παρουσιάζονται στο Σχήμα 10. Από την επισκόπηση των αποτελεσμάτων είναι φανερό ότι η προτεινόμενη μέθοδος προσταθεροποίησης του καθολικού προβλήματος με τη μέθοδο $P-FETI-DP$, προσφέρει σημαντική μείωση τόσο αναφορικά με τον αριθμό επαναλήψεων όσο και με τον συνολικό απαιτούμενο χρόνο για την επίλυση του συστήματος.

Έχοντας αντιμετωπίσει επιτυχώς την επίλυση προβλημάτων συνεχούς μέσου που διακριτοποιούνται είτε με τη μέθοδο *Galerkin* είτε με τη μέθοδο *collocation*, το τελευταίο τμήμα της διατριβής επικεντρώνεται στη μελέτη πολύπλοκων κατασκευών με δομικά ισογεωμετρικά στοιχεία κελύφους. Με δεδομένο ότι η ισογεωμετρική ανάλυση μπορεί να περιγράψει με μηδενικό γεωμετρικό σφάλμα οσοδήποτε πολύπλοκες γεωμετρίες, θεωρείται ιδανική στην περίπτωση ανάλυσης λεπτότοιχων κατασκευών όπου οι αρχικές γεωμετρικές ατέλειες μπορούν να επηρεάσουν σημαντικά την τελική μηχανική απόκριση του φορέα. Οι λεπτότοιχες κατασκευές στα πλαίσια των υπολογιστικών μεθόδων περιγράφονται συχνά από δυο θεω-



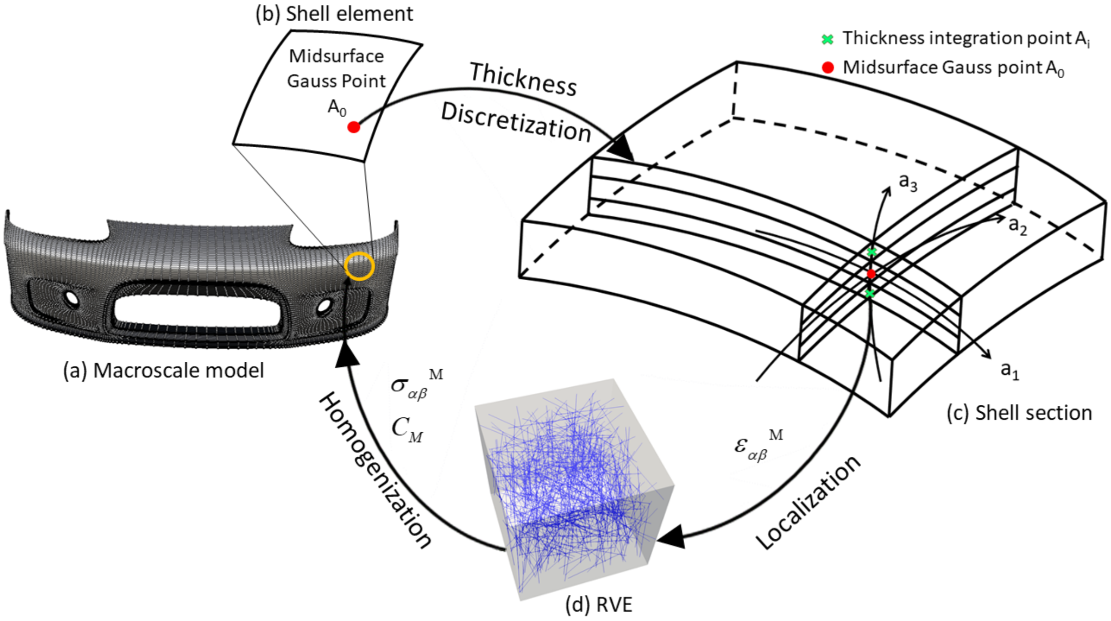
Σχήμα 9: Ενδοσυνοριακό και αδρό πρόβλημα της μεθόδου $P-FETI-DP$.



Σχήμα 10: Σύγκριση επιδόσεων μεθόδων επίλυσης φορέων διακριτοποιημένων με *collocation*.

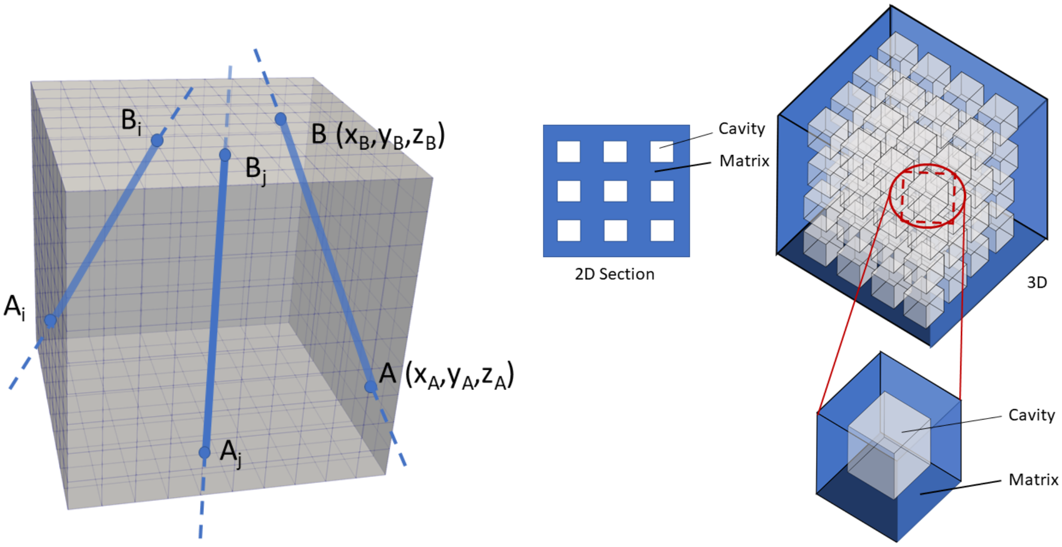
ρίες, συγκεκριμένα την Kirchhoff-Love για λεπτότοιχα κελύφη και την Reisner-Midlin για κελύφη με μεγάλο πάχος. Παρόλο που οι περισσότερες κατασκευές ανήκουν στην πρώτη κατηγορία, οι περιορισμοί συνέχειας της μεθόδου πεπερασμένων στοιχείων, καθιέρωσαν την θεωρία Reisner-Midlin στην πλειονότητα των λογισμικών. Οι αυξημένη συνέχεια από C^0 σε C^{p-1} που εισάγεται στην γεωμετρία των φορέων με την ισογεωμετρική ανάλυση, επιτρέπει πλέον μια άμεση υλοποίηση σε κώδικα της τεχνολογίας των λεπτότοιχων κελυφών. Για το λόγο αυτό, και σαν επέκταση των υφιστάμενων θεωριών λεπτών κελυφών για διαφορετικά υλικά, εισάγεται μια μεθοδολογία ανάλυσης τους, η οποία μπορεί να συνδεθεί με οποιοδήποτε υλικό λαμβάνοντας υπόψη τη μικροδομή του μέσω μιας ανάλυσης πολλαπλών κλιμάκων.

Συγκεκριμένα οι κατασκευές που λαμβάνονται υπόψη στην προτεινόμενη μεθοδολογία έχουν μια μακροδομή που αναπαριστάται με στοιχεία κελύφους, ενώ η μικροδομή του είναι ένα σύνθετο υλικό που μπορεί να περιέχει τυχαία εγκλείσματα ή κενά. Αυτές οι ετερογένιες της μικροδομής θεωρούνται μη αντιληπτές σε μακροσκοπικές διαστάσεις, όμως για να ληφθεί υπόψη η επιρροή τους στις καταστατικές εξισώσεις, προτείνεται ένα εμφολευμένο σχήμα Ισογεωμετρική Ανάλυσης - Πεπερασμένων Στοιχείων το οποίο περιγράφεται σχηματικά στην εικόνα 11.


(b) Shell element
Midsurface Gauss Point A_0
Thickness Discretization
Thickness integration point A_i
Midsurface Gauss point A_0
a_3
a_2
a_1
(a) Macroscale model
(c) Shell section
$\sigma_{\alpha\beta}{}^{M}$
C_M
Homogenization
$\varepsilon_{\alpha\beta}{}^{M}$
Localization
(d) RVE

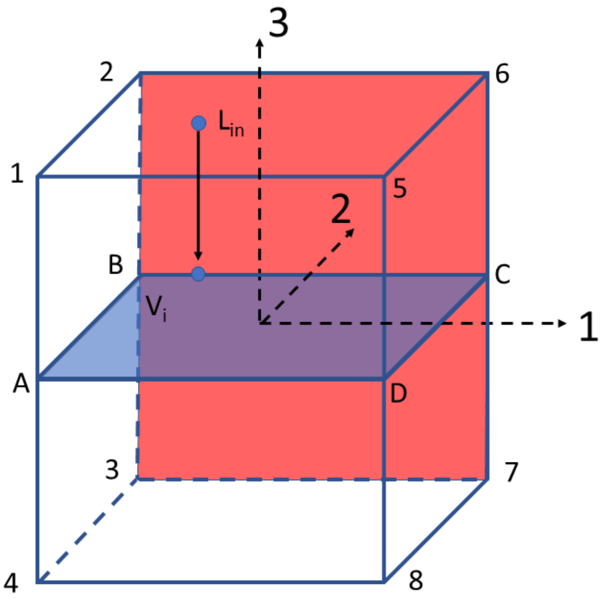
Σχήμα 11: Εμφωλευμένη διαδικασία ισογεωμετρικής ανάλυσης-πεπερασμένων στοιχείων για τον υπολογισμό καταστατικών σχέσεων με τη βοήθεια ανάλυσης πολλαπλών κλιμάκων.

Συγκεκριμένα στο πρώτο βήμα της διαδικασίας, γίνεται σχεδίαση και διακριτοποίηση του μακρομοντέλου με στοιχεία κελύφους με τη χρήση συναρτήσεων σχήματος $T - Splines$. Από το προσομοίωμα αυτό εξάγονται τα αντίστοιχα στοιχεία Bezier, τα οποία αποτελούν τη βάση για τον υπολογισμό του μητρώου στιβαρότητας. Για κάθε σημείο ολοκλήρωσης της μέσης επιφάνειας του κελύφους, ορίζεται ένα πλήθος σημείων ολοκλήρωσης κατά το πάχος της διατομής. Κάθε ένα από αυτά τα σημεία ορίζει μια ζώνη υλικού που παραμορφώνεται σε συνθήκες επίπεδης έντασης. Αυτή η στρωματοποιημένη αναπαράσταση επιτρέπει επίσης την μοντελοποίηση της μεταβλητότητας του υλικού στο πάχος της διατομής. Στο δεύτερο βήμα της διαδικασίας, οι καταστατικές σχέσεις του υλικού εξάγονται σε κάθε σημείο ολοκλήρωσης της μέσης επιφάνειας μέσω της υπολογιστικής ομογενοποίησης του. Σε κάθε σημείο ολοκλήρωσης της διατομής, ορίζεται ένας τρισδιάστατος αντιπροσωπευτικός όγκος, διακριτοποιήμένος με πεπερασμένα στοιχεία τρισδιάστατης ελαστικότητας, στον οποίο ενσωματώνονται τυχαία εγκλείσματα. Τα εγκλείσματα αυτά μπορούν να προσομοιωθούν με δομικά πεπερασμένα στοιχεία, όπως δικτυώματα, δοκοί και κελύφη. Σε περίπτωση τυχαίας μορφής ογκομετρικών εγκλεισμάτων, αυτά μπορούν να διακριτοποιηθούν και να προσομοιωθούν με τρισδιάστατα πεπερασμένα στοιχεία συνεχούς μέσου, όπως φαίνεται και στην εικόνα 12



Σχήμα 12: Παραδείγματα χαρακτηριστικών όγκων

Στον αντιπροσωπευτικό όγκο που απέχει απόσταση ζ από τη μέση επιφάνεια της διατομής, επιβάλλονται ισοδύναμες μετατοπίσεις που υπολογίζονται με βάση τις εντός επιπέδου παραμορφώσεις του μακροσκοπικού προσομοιώματος στο συγκεκριμένο σημείο. Με τον τρόπο αυτό ορίζεται το πρόβλημα συνοριακών τιμών του αντιπροσωπευτικού όγκου. Στα πλαίσια της προτεινόμενης διαδικασίας θεωρούμε γραμμική απόκριση του αντιπροσωπευτικού όγκου, αλλά η επέκταση σε μη-γραμμικές γεωμετρίες μπορεί να εφαρμοστεί άμεσα χωρίς αλλαγές στην διαδικασία.

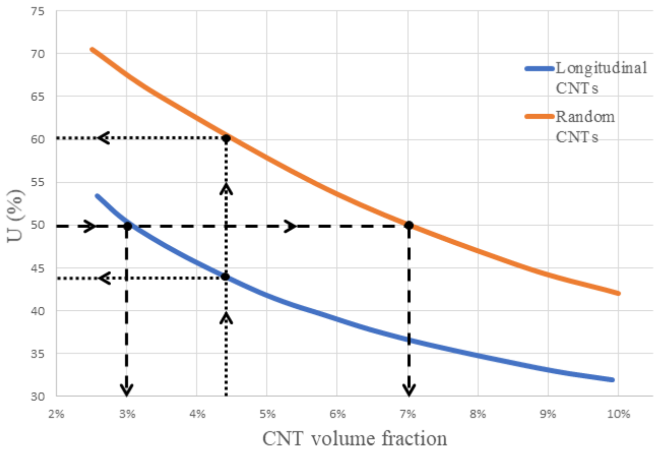


Σχήμα 13: Επιβολή συνοριακών συνθηκών χαρακτηριστικού όγκου μεσω εικονικού στοιχείου της μέσης επιφάνειας.

Στο σχήμα 13, απεικονίζεται γραφικά η επιβολή των μετατοπίσεων στον αντιπροσωπευτικό όγκο. Συγκεκριμένα, οι μακροσκοπικές παραμορφώσεις $\hat{\epsilon}_M$ μεταφράζονται σε επιβαλλόμενες μετατοπίσεις στο εικονικό στοιχείο $ABCD$ του σχήματος 13 μέσω της σχέσης 8.

$$\hat{u}_m = \hat{\epsilon}_M \hat{\mathbf{X}}_M \tag{8}$$

Με δεδομένες τις εντός επιπέδου μετακινήσεις του εικονικού στοιχείου, γίνεται επιβολή των μετακινήσεων των εντός επιπέδου βαθμών ελευθερίας, των πλευρικών επιφανειών του αντιπροσωπευτικού όγκου. Αντίθετα, οι εκτός επιπέδου βαθμοί ελευθερίας του φορέα είναι ελεύθεροι να παραμορφωθούν, με εξαίρεση του κατακόρυφους βαθμούς ελευθερίας των σημείων 3,4,7 με σκοπό την αποφυγή τυχόν κινήσεων στερεού σώματος. Στη συνέχεια, με χρήση του τύπου



Σχήμα 14: Διάγραμμα επιρροής προσανατολισμού CNT στην μακροσκοπική μηχανική συμπεριφορά.

$$C = \frac{1}{\|V\|} D\tilde{K}_{ee}D^T \tag{9}$$

προκύπτει η συνεισφορά του μητρώου επίπεδης έντασης από κάθε ζώνη της διατομής, όπου V ο όγκος της διατομής και $\tilde{K}_{ee}$ το μητρώο στιβαρότητας του αντιπροσωπευτικού όγκου στατικά συμπυκωμένο στους εντός επιπέδου περιφεριακούς βαθμούς ελευθερίας. Μετά από ολοκλήρωση κατά το πάχος της διατομής σε κάθε σημείο μέσης επιφάνειας, προκύπτουν τα καταστατικά μητρώα καθώς και οι αντίστοιχες τάσεις στη μέση επιφάνεια του φορέα.

Με τη βοήθεια στοχαστικών διαδικασιών, μελετήθηκαν κατασκευές πραγματικής κλίμακας, με τη χρήση της παραπάνω εμφωλευμένης διαδικασίας υπολογισμού των καταστατικών μητρώων, τόσο σε τυχαία κατανομή υλικού μέσα στο ίδιο φορέα, όσο και διακυμαινόμενο υλικό μεταξύ διαδοχικών αναλύσεων. Η εξέταση των αποτελεσμάτων, αναδεικνύει τη σημασία της λεπτομερούς προσομοίωσης της μικροδομής του υλικού, καθώς μικρές διακυμάνσεις στη περιεκτικότητα ή στο υλικό των εγκλεισμάτων, αποδεικνύεται πως έχει σημαντική επίπτωση στη δυσκαμψία του συνολικού φορέα, επηρεάζοντας τη συνολική αντοχή εως και 30% σε σχέση με την αρχική ονομαστική τιμή, όπως φαίνεται και στο Σχήμα 14.

Contents

List of Figures

List of Tables

IGA	Isogeometric Analysis
CAD	Computer Aided Design
CAE	Computer Aided Engineering
FEM	Finite Element Method
PDE	partial differential equation
NURBS	Non-Uniform Rational B-Splines
BVP	Boundary Value Problem
DDM	Domain Decomposition Methods
PCG	Preconditioned Conjugate Gradient
BDDC	Balancing Domain Decomposition by Constraints
RVE	Representative Volume Element
VF	Volume Fraction
CNT	Carbon Nanotube
PEEK	poly-ether-ether-ketone
EBE	Equivalent Beam Elements
MSM	Molecular Structural Mechanics
HPC	High performance computing
PSM	Primal Substructuring Method
DDM	Domain Decomposition Methods
dof	degrees of freedom
GMRES	Generalized Minimal Residual Technique
SSOR	Successive Symmetric Over-Relaxation

IETI	Isogeometric Tearing and Interconnecting
OAS	overlapping additive Schwarz
FETI-DP	Finite Element Tearing Interconnecting Dual-Primal

0
Introduction

0.1 Motivation

Until the early 20^{th} century, scientists and mathematicians tried to interpret the behavior of natural phenomena by introducing mathematical models that replaced their fundamental physics. In most cases, this required the transformation of an engineering system in terms of partial differential equations. At first, it was plausible for scientists to find an analytical solution that satisfied the boundary values problems at hand, yet the ever rising need to describe more complex natural systems, quickly eliminated the possibility of attaining an exact solution field that satisfies the problem. Instead, techniques to approximate the exact solution were devised. One of the most widespread methods for approximating the partial differential equation (PDE), the Finite Element Method (FEM), was born in the field of aeronautics. It wasn't until 1944, when John Argyris, faced the problem of reliably simulating inclined geometry wings, that the FEM method was born. After experimenting with all known methods with unsatisfactory results, he coined the use of the first triangular finite element and its implementation in the first electromechanical computers and published the ”Energy theorems” book where he first registers the method. In the years to come, many scientists such as Clough, Turner and Martin calculated the stiffness matrices of various elements, with Zienkiewicz being the first to publish a book on the field. In alignment with the evolution of computers, a large number of publications followed, that allowed engineers to become acquainted to the method and subsequently apply it to numerous aspects of their respective scientific fields. Even though the evolution of computational methods, with the spearhead being FEM was rapid, some significant deficiencies remained. For instance, exact representation of complicated geometries remains an open issue as even higher order finite elements can only reduce the geometrical error. This inherent approximation leads to erroneous results in geometry sensitive analyses such as shell buckling. In a similar manner, adaptive refinement techniques cannot be efficiently implemented, since in case of non-trivial geometries the immediate connection with the Computer Aided Design (CAD) representation is not existent. All the above deficiencies of the existing Computer Aided Engineering (CAE) methods, have raised the need for a common development pipeline with CAD.

Computer Aided Design is defined as the drafting procedure utilizing computers. Its main goal is to aid designers to produce accurate blueprints in a less laborious procedure. One of the earliest works in the field of design was proposed by Ferguson in 1964 [74]. This work introduced Ferguson curves, which are a specific category of third degree power basis curves calculated with known positions and gradients of the curve at its starting and ending points. The generalization of Ferguson curves for arbitrary polynomial degree yields Hermite curves [L'Hermite], where a $(2k+1)$ degree curve is evaluated by utilizing k values at its starting and ending points. Despite their straightforward mathematical representation, power basis curves were early deprecated by designers as they were considered impractical for design purposes. This is attributed to the algebraic nature of the existing algorithms which are prone to round-off errors, along with the accompanying coefficients that convey only little information regarding the geometry of the curve. To this end, Bézier curves were introduced by Pierre Bézier [111, 37, 42, 75, 81, 72, 88], as a design tool for the bodywork of cars. They are considered equivalent with polynomial curves, yet remedy some of their disadvantages. The shape of the Bézier curve now depends only on Control Points, which are Cartesian points that control the curves shape, while allowing for a more natural design process and manipulation of the curve. In addition, the augmented polynomial degree is tightly coupled to number of points used for the representation of the curve, fact that automatically leads to smoother curves. Unfortunately, this property of Bézier curves is at the same time their greatest merit, as well as their major drawback, as the increased polynomial degree, leads to instability of the algorithms. B-Splines were introduced [162, 54, 58, 149, 59] as a generalization of one span Bézier curves into multiple intervals. They disengage the number of shape function of the curve from the polynomial degree, thus enabling local shape function support. In addition to the partial control of the curves, abrupt changes of the curves' geometry is now allowed due to the introduction of the Knots which divide the curve into piece-wise polynomial segments. Yet, the constantly increasing need of designers to reproduce ever more complex shapes, rendered B-Splines insufficient as they cannot accurately represent all conic sections. To this end, Non-Uniform Rational B-Splines (NURBS) technology were introduced [180, 175, 142, 143], that incorporate a weight for each of the curve control points. This extensions introduces a projection that allows for the accurate description of more elaborate geometries. The limitation of NURBS arises due to the fact that surface or solid geometries cannot be efficiently represented as the physical geometry must be mapped into a linear, rectangular or cuboid domain in parameter space. As a result, the burden is sifted to designer to partition the geometrical domain into NURBS suitable parts. Each of these individual geometries defines a single parameter space called Patch, that can have different attributes with its adjacent pieces thus rendering it uncommon for their edges to coincide. T-Splines technology was recently introduced [170, 169, 168] to remedy the deficiencies of NURBS. Specifically, T-Splines abolish the tensor product parameter space structure of former technologies. This makes attainable the water-tight connection between patches, while being the first technology that enables true local refinement of the geometry.

Isogeometric Analysis (IGA) was introduced in [89] by T.J.R. Hughes in order to address the need of a common development pipeline between CAD and CAE industries in order to overcome geometry approximation issues that arise in analysis. The main idea behind the method is the utilization of the underlying geometry mesh as a basis for analysis. This concept renders the process of meshing the geometry obsolete, thus minimizing the time needed for an analysis suitable discretization. These favorable properties of IGA render it more efficient in various computational mechanics fields, such as optimization problems [181, 122, 148, 147, 171, 136]. Due to the high smoothness of CAD shape functions, the method has showcased significant advantages over conventional approaches in fluid mechanics applications [79, 133, 21, 22, 23, 135, 134] as well, while its utilization in structural dynamics applications [53, 65, 90, 30, 91] rendered favorable results over its FEM alternatives. These initial works, were mainly focused on continuum mechanics applications by utilizing the discretization of bivariate or trivariate domains with CAD shape functions, such as NURBS and T-Splines later on. Apart from planar and solid geometries, IGA is considered ideal for the analysis of shell structures, due to its ability to accurately describe complex geometries. Shell theories are derived from a dimensionality reduction of structures, for which one of the dimensions is significantly smaller than the rest. In case of shells, the redacted dimension regards the thickness of the structure. As a result, shell theories transform the three-dimensional elasticity equation, into two dimensional domains, represented only by the midsurface of the structure. Two major shell theory branches exist based on the thickness of the shell's section. In case of thick shells, transverse shear deformation is taken into account and thus Reissner-Midlin theory is used, while for thin shells Kirchhoff-Love theory is more suitable. Even though most industrial products belong to the thin shell case, its computational limitations when formulated in the FEM framework, made Reissner-Midlin theory dominant in most commercial FEM codes. This is attributed to the prerequisite of Kirchhoff-Love shells for a minimum C^1 continuity between adjacent elements, which classical FEM cannot efficiently provide, thus resulting to non-conforming meshes. Special care must be taken for thin shell theory to be applicable in the context of FEM such as non-local formulations, nodal enforcement of C^1 continuity or even penalty method applied to selected material points. The main benefit of IGA in this field is that the higher interelement continuity of the utilized shape functions, enables a straightforward implementation of Kirchhoff-Love shells. Specifically, the first isogeometric Kirchhoff-Love formulation was introduced in [99], which introduced a geometrically non-linear, rotation-free element. Having [99] as a basis, many material formulation where then coupled with it. For instance, [100] extended the latter formulation to large strains and compressible and incompressible hyperelastic non-linear material laws. Progressive damage models of composite laminates in the frameowrk of isogeometric thin shells were explored in [61], while biological membranes were investigated in [174]. In a similar fashion, the first plasticity models applied to Kirchhoff-Love shells appeared in [7] and composite laminate materials coupled with gradient enhanced damage models in [144]. Apart from the application of different material in the Kirchhoff-Love theory, significant contributions were performed in combining shells with different shape functions such as T-Splines in [39, 117, 40] or rational triangular Bezier splines in [186]. Finally, different aspects of Kirchhoff-Love shells were explored in a multitude of contributions, ranging from the coupling of non-conforming Kirchhoff-Love shell patches [85, 82, 163], to membrane locking effects addressed with mixed formulations.

All the aforementioned contributions, have their roots on the fact that a remarkable accuracy per degree of freedom is introduced along with IGA, which is mainly attributed to the increased interelement continuity of the shape functions. Unfortunately, this augmented accuracy comes at a compelling cost for the assembly of the resulting stiffness matrices in case of Galerkin discretizations [92, 15].To this end, the efforts of the scientific community have been shifted towards the development of computational effortless alternatives. This process lead to the development of isogeometric collocation methods [18, 11], which require the evaluation of a single integration point per shape function for the computation of the stiffness matrix, thus alleviating the computational cost for the assembly. Since its introduction a plethora of contributions studied the enforcement of Dirichlet [44] and Neumann [60] boundary conditions, while at same time delving into the method's convergence and integration properties [160, 113, 66, 131]. The majority of the published manuscripts explored the coupling of isogeometric collocation with structural elements. Specifically, scientific areas such as Beams [28, 124, 125, 126, 20, 98] and Rods [14, 17, 182, 183, 129] were extensively explored, as isogeometric collocation methods were proven extremely capable at alleviating locking effects [29, 14]. In a similar fashion new formulation for surface structural elements that utilize isogeometric collocation methods for their integration were proposed, such as plate [150, 97, 139, 128] and shell structural elements [31, 16, 101, 128]. In addition, due to their favorable properties, isogeometric collocation methods are utilized for the modelling of computationally demanding computational mechanics fields such as Phase-Field modelling [80, 159] and dynamic applications [62, 30, 65].

As mentioned in the latest paragraphs, the strongest asset of isogeometric methods is the increased interelement continuity of the utilized shape functions. In case of CAD, this results in smooth and accurate representation of curves, while for computational methods it provides smooth variation of the analysis characteristics. This property, despite providing enhanced accuracy, adds a significant computational burden to the solution of the resulting linear systems due to their reduced sparsity patterns. As a result, efficient solution schemes for large-scale isogeometric applications are a necessity for the establishment of IGA. In case of isogeometric Galerkin discretizations, various methods have been proposed in order to address the solution of the resulting linear systems. Initial publications were focused on assessing the relationship between solution cost and degrees of freedom (dof) in case of IGA. Continuities ranging from minimum C^0 of FEM to full C^{p-1} continuity of IGA were studied in [51] to provide a theoretical relationship between solution cost and polynomial order for the case of direct solvers, while

MUMPS [8] and PARDISO [158] solvers provided the equivalent solution times. This research was afterwards extended to iterative solvers, where the PCG solution algorithm was employed with various preconditioners, such as diagonal Jacobi [140], Successive Symmetric Over-Relaxation (SSOR) [45] and incomplete Cholesky factorizations [95] in order to assess their computational cost in case of highly continuous isogeometric discretizations. These investigations made the need for efficient solution schemes for IGA even more profound, hence the scientific community redirected its efforts to the development of efficient and scalable domain decomposition solvers. The first attempt in this direction was performed by [55] where overlapping additive Schwarz preconditioners were exploited for the solution of isogeometric elliptic problems. The latter work was extended in [179] by providing Schwarz preconditioners in both primal and mixed formulations for the case of the linear elasticity PDE. Balancing Domain Decomposition by Constraints (BDDC) preconditioner for isogeometric scalar elliptic problems was investigated in [27], which in [57] was paired with a novel scaling method. Isogeometric Tearing and Interconnecting (IETI) was also introduced [106] as an isogeometric variant of the well-known Finite Element Tearing Interconnecting Dual-Primal (FETI-DP) method. In a similar fashion, isogeometric mortar methods were investigated in [35, 36], in order to address the coupling of non-conforming subdomains.

Similar research initiatives addressed the solution of the linear systems emerging from isogeometric collocation method, which despite alleviating the computational cost for the formation of the resulting matrices, shift the burden to the solution of their non-symmetric linear systems whose condition number grows rapidly in cases of mesh refinement or augmented polynomial degree. To address this issue, several methods have been proposed. In [25] overlapping Schwarz preconditioners accelerated with GMRES [156] iterative method was studied. In this research, the non-symmetric matrix was decomposed into a set of overlapping submatrices, while a coarse problem was introduced based on an interpolation operator between the reference domain and the tensor product structure generated from the univariate common points among subdomains. The solution of nanolithography problems was investigated in [114], where a multi-frontal solver was implemented for Graphic Processing Units (GPU) for the case of one-dimensional isogeometric collocation. Finally, optimal multilevel preconditioners accelerated by GMRES were investigated in [47], for collocation discretizations deriving from second order elliptic problems.

The current research aims to address the efficient implementation of isogeometric methods in real-scale mechanical applicaions. Specifically, [173] proposes a family of algorithms for the purposes of IGA, that exploit the advantages of iterative solution schemes when combined with Domain Decomposition Methods (DDM), in the context of isogeometric Galerkin discretizations. By introducing an appropriate modification to the overlapping nature of NURBS shape functions, a robust and scalable preconditioner is developed that minimizes the computational cost for the solution of large-scale isogeometric problems. In this work, the PCG iterative solver is combined with the IETI domain decomposition method, that showcases a considerable improvement compared to existing solution techniques in case of positive definite symmetric linear systems, while at the same time ensuring properly load balanced subdomains partitioning. In a similar fashion, [178] introduces a non-overlapping decomposition of the non-symmetric matrices deriving from isogeometric collocation methods. This allows development of a preconditioner based on P-FETI-DP domain decomposition method. Thus, a two-level algorithm is generated based on GMRES iterative method. The preconditioner utilizes static condensation to minimize the global problem to the an interface problem among adjacent subdomains. The interface problem which is also a non-symmetric matrix, is then preconditioner by the first iteration of the FETI-DP method, which translates to a coarse problem defined by the common degrees among more than two subdomains. Numerical results indicate an enhanced performance of the proposed solution scheme, compared to the most competitive alternative, the overlapping additive Schwarz method. Finally, in terms of modelling of real-life structures, [177] extends existing isogeometric thin shell formulations, to incorporate constitutive laws generated by stochastic multiscale analyses. The integration of the constitutive law is performed through the thickness of the shell, where an arbitrary Representative Volume Element (RVE) defines the microstructural topology of the composite material under consideration. Numerical results demonstrate the applicability of the proposed formulation in both benchmark and real-scale numerical examples, thus rendering this formulation ideal for the simulation of shell structures in combination with composite materials.

0.2 Outline

This thesis is organized in 6 chapters as follows:

Chapter 1 presents the basic concepts of CAD shape functions, which lay foundations for the establishments of Isogeometric Analysis method. With starting point being the polynomial curves and their equivalent shape functions, this Chapter analyzes the historical evolution of geometrical representations, browsing through Bézier curves, B-Splines, NURBS and T-Splines and the reasons that lead to the establishment and equivalent deprecation of each individual methodology.

Chapter 2 introduces the Isogeometric Analysis method. In analogy to Finite Elements, the transformation of the mesh-less geometrical representation of CAD objects, into analysis related entities such as elements and integration points is presented. Elemental quantities in each shape function category are addressed explicitly, while their application on the linear elasticity PDE is examined for continuum mechanics, structural shell elements and the strong form of the PDE via the utilization of the isogeometric collocation method.

Chapter 3 provides the basic concepts of solution schemes utilized for the efficient solution of the linear systems deriving from isogeometric analysis. Two major categories of iterative and domain decomposition solution schemes are studied, thus introducing the concepts of robust and efficient solution methods for tackling both symmetric and non-symmetric linear systems. Specifically, for the case of iterative solvers, PCG and GMRES algorithms are studied, while the isogeometric equivalents of FETI-DP, PSM and P-FETI-DP method are elaborated.

Chapter 4 addresses the solution of symmetric positive definite linear systems deriving from the isogeometric Galerkin method for the case of B-Splines and NURBS discretizations. The increased interlement continuity that accompanies isogeometric analysis, along with the augmented shape function polynomial degrees, lead to a severe deterioration of the solution performance of the resulting linear systems. By proposing an appropriate non-overlapping decomposition of the domain into subdomains, a scalable and robust domain decomposition preconditioner emerges, that accelerates the PCG method. Numerical results showcase the efficacy of the proposed solution scheme that enables the subdivisioning of a model in an arbitrary fashion that ensures properly load balanced subdomains, while offering a substantial computational improvement compared to existing methods.

Chapter 5 addresses the solution of non-symmetric linear systems arising from the integration of the computational domain using the isogeometric collocation method. Additional to the increased population and bandwidth of the resulting matrices, introduces with the isogeometric Galerkin method, collocation significantly burdens the computational cost for the solution of the resulting linear systems due to their non-symmetric nature. To this end, a family of primal non-overlapping domain decomposition solution schemes are studied in this chapter in order to provide a scalable and robust preconditioner that accelerates the iterative GMRES method.

Chapter 6 examines the performance of isogeometric shells under the prism of semi-concurrent multiscale material modelling. Specifically, existing isogeometric thin shell formulations, employing Kirchhoff-Love shell theory, are extended to incorporate detailed modelling of advanced composites. A nested IGA-FEM multiscale analysis scheme is proposed, in which IGA is used for the discretization of the macroscopic level and FEM for the discretization of the corresponding RVE. As a result, the plane stress material required for the integration of the shell stiffness is derived via a computational homogenization procedure, thus taking into account the material microstructural topology. Numerical results, showcase the effect of detailed material modelling to the overall mechanical performance of the shell structure, as well as the applicability of the proposed formulation to real life applications.

Finally, Chapter 7 provides the conclusions drawn from this research and presents a summary of the contributions.

1

Computer Aided Design

1.1 Introduction

Until the early 20th century, designers were creating blueprints by hand utilizing flexible wooden or metal stripes to design any curves needed. The design process required extensive effort as any manipulation or correction to the design might lead to a re-iteration of the whole blueprint. Smooth polynomial lines (Splines) were introduced in 1960s with Bézier curves, which were simultaneously developed by Paul de Casteljau and Pierre Bézier and used for the design of automobiles. B-Splines and NURBS were introduced in the 1980s and were extensively developed and used in the industry. Recently, the design technology of T-Splines was proposed by Sederberg as a generalization of NURBS that can accommodate arbitrarily complex geometries. The mathematical breakthroughs of the design technology as well as the rapid evolution of the personal computer led to the evolution of Computer Aided Design (CAD) which replaces the manual drafting process and thus minimizes the required effort.

In this chapter, the evolution path of CAD from polynomial curves to advanced Splines is presented. The aim of this review is to introduce the basic principles of CAD shape functions, curves, surfaces and volumes that will be utilized throughout the rest of the manuscript, as well as the reasons that lead to their establishment.

1.2 Polynomial curves

The fundamental representations of curves and surfaces, in geometrical modeling, are performed with the aid of implicit and parametric equations. The implicit equation of a curve in the Cartesian plane XY is given by an equation of the form:

$$f(x, y) = 0 \tag{1.1}$$

Equation 1.1 describes the relationship between the x and y coordinates of every point that lies on the curve. An example of a ellipse described with an implicit equation forms is $f(x, y) = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$.

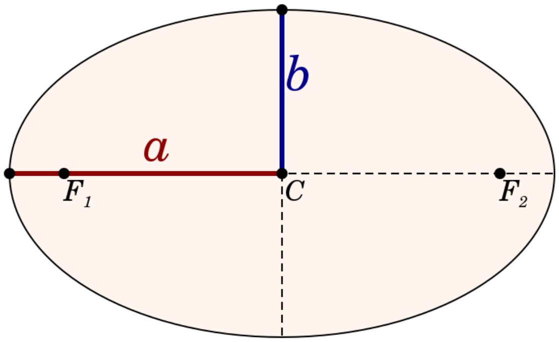


Figure 1.1: Ellipse

In contrast to the implicit case, the parametric form provides a separate equation for each of the coordinates of the curve points as functions of an independent parameter.

$$C(t) = (x(t), y(t)) \qquad t_0 \leq t < t_1 \tag{1.2}$$

With the aid of the parametric form, the equation of the ellipse can be written as follows:

$$\begin{aligned} x(t) &= a \cdot cos(t) \\ y(t) &= b \cdot sin(t) \qquad 0 \leq t < 2\pi \end{aligned} \tag{1.3}$$

Comparing these two equation forms, it is apparent that the implicit form cannot provide directly points that belong to the curve while it is highly depended on the coordinate system. As a result, the transformation to alternative coordinate systems is quite cumbersome. On the contrary, in the parametric form, any point of the curve can be imminently derived from the equations, even if the equations describe a closed or a multiple valued curve. The basic disadvantage of the parametric form is that it is not straightforward to determine whether a random point belongs to the curve. All free form curves presented in this chapter are based on the parametric representation of curves.

One of the most common types of parametric curves are power basis functions, where the curve equation is a sum over powers of a parameter u, multiplied by equivalent polynomial coefficients a_i. The general equation form that provides the power representation of curves is:

$$C(u) = \sum_{i=0}^{n} a_i \cdot u^i \tag{1.4}$$

1.2.1 Ferguson curves

Ferguson curves were proposed by Ferguson in 1964 [74] to aid the design of aircraft surfaces. Ferguson curves are a specific category of third degree power basis curves. A simple third degree power basis curve is defined as

$$C(u) = a_0 + a_1 \cdot u + a_2 \cdot u^2 + a_3 \cdot u^3 \qquad 0 \leq u \leq 1 \tag{1.5}$$

In the case of Ferguson curves, functions $1, u, u^2, u^3$ are replaced the Hermite polynomials as follows:

$$\begin{aligned} &1 \quad \textit{is replaced with } (1 - 3u^2 + 2u^3) \\ &u \quad \textit{is replaced with } (3u^2 - 2u^3) \\ &u^2 \quad \textit{is replaced with } (u - 2u^2 + u^3) \\ &u^3 \quad \textit{is replaced with } (-u^2 + u^3) \end{aligned} \tag{1.6}$$

For this choice of polynomial functions the polynomial coefficients known and are the positions and derivatives of the curve at its initial and final point. Specifically, $a_0 = C(0)$, $a_1 = C(1)$ and $a_2 = C'(0)$ $a_3 = C'(1)$ equivalently. As result eq. 1.5 with the aid of 1.6 becomes:

$$\begin{aligned} C(u) =&(1-3u^2+2u^3)\cdot C(0)+(3u^2-2u^3)\cdot C(1)+ \\ &(u-2u^2+u^3)\cdot C'(0)+(-u^2+u^3)\cdot C'(1) \end{aligned} \tag{1.7}$$

The basic deficiency of a Ferguson curve is that is represented only by two points and as a result a multitude of *n-1* Ferguson curves are needed to interpolate n points. In order to merge consecutive Ferguson curves, compatibility of the positions, derivatives as well as C^2 continuity are required. This translates the solution of a system with *n-2* equations in order to determine the internal derivative values for internal points. A generalization of the Ferguson curves for arbitrary degree is attained by Hermite curves, where a *(2k+1)* degree curve is defined by utilizing k derivatives at the start and end point of the curve.

The power representation of curves was abandoned early as it was considered impractical by designers due to their significant disadvantages. The most important one is that they are considered unnatural for interactive shape design, as the polynomial coefficients a_i convey little information about the geometry of curve. In addition, most algorithms for the evaluation of polynomial curves have an algebraic background rather than geometric one, while they are prone to round-off errors when the polynomial coefficients vary greatly in size.

1.3 Bézier curves

Bézier curves were introduced by Pierre Bézier in the 1960s [111] as a design tool for the bodywork of cars. They are considered mathematically equivalent with polynomial curve forms as they use polynomials for the coordinate functions. A thorough investigation of Bézier curves theory can be found in [37, 42, 75, 81, 72, 88].

Bézier curves remedy the disadvantages of power basis polynomials. The shape of curve depends only on the Control Points of the curve, thus allowing a more natural design process and manipulation of the curve. At the same time, the number of points used for the definition of a Bézier curve n is tight coupled to its polynomial degree p (p=n-1). This leads to smoother curves as higher continuity is achieved due to the increase of the polynomial degree. The general form of a Bézier curve is given by:

$$C(u) = \sum_{i=0}^{n} B_{i,n}(u) P_i \qquad 0 \leq u \leq 1 \tag{1.8}$$

where $B_{i,n}$ are the Bernstein polynomials given by:

$$B_{i,n} = D(n,i) u^i (1-u)^{n-i} \tag{1.9}$$

and $D(n,i)$ the binomial coefficients.

$$D(n,i) = \frac{n!}{i!(n-i)!} \tag{1.10}$$

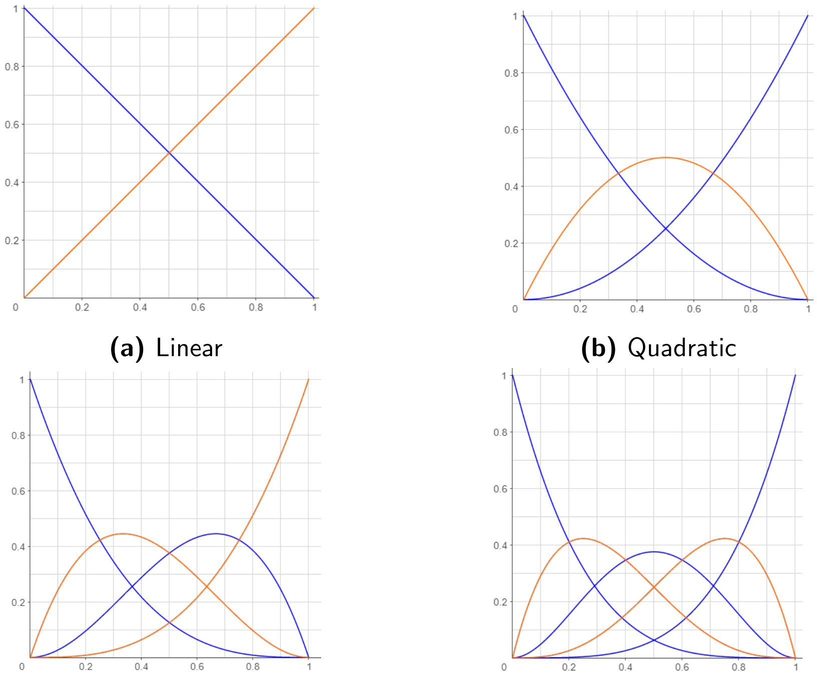
Figure 1.2 gives an example of the Bernstein polynomials used for various polynomial degrees. As noted, there is an immediate connection between the polynomial degree of the curve and the number of shape function or equivalent Control Points. For instance, for degree $p = 2$, a multitude of three Bernstein polynomials are required for the evaluation of the curve. In addition, by examining the form of the Bernstein polynomials the following properties are apparent. The shape functions are defined in the span $[0,1]$ and the shape functions are symmetric across x=0.5 axis, their values are non-negative and they abide to the partition of unity property.

Figure 1.3 illustrates an example of a cubic Bézier curve, where some principal properties are illustrated. Initially, the curve is interpolatory to the first and last Control Points and tangential to the Control Polygon, depicted with green line segments, at the edges of the curve. Since Bernstein polynomials are symmetric across the middle of the parametric domain, this property is also transferred to the curve and thus an inversion of the curve's Control Points does not affect the shape of the curve. A manipulation of a Control Point affects the entirety of the curve, as all Bernstein polynomials span throughout the parametric domain, while the most affect area of the curve is near the maximum value of the equivalent Bernstein shape function affected.

The major drawback of Bézier curves is the immediate link between polynomial degree and the number of Control Points. Specifically, since *p+1* Control Points are required for the definition of the Bézier curve, p being the polynomial degree, it is apparent that an increase of the polynomial degree raises the number of Control Point and affects the stability of the algorithms. In addition to this, as Figure 1.2 illustrates, shape functions

[*] Created with Geogebra

[†] Created with Desmos



(a) Linear

(b) Quadratic

(c) Cubic

(d) Quartic

Figure 1.2: Bernstein polynomials for various polynomial degrees.*

span throughout the parametric domain. This severely affects the geometric modeling process as even minor alterations of the Control Points affects entirety of the curve.

1.4 B-Splines

B-Splines were introduced in [162] as a generalization of one span Bézier shape functions into multiple consecutive intervals. They are considered an extension of Bézier curves that were developed to alleviate their deficiencies. Specifically they introduced the local support of shape functions, thus allowing partial control of the curves, while abrupt changes of the curve continuity are allowed with Knots.

Since B-Splines are considered a superset of Bézier curves, there is an immediate transformation between the two types. Specifically, each B-Spline curve can be split into a multitude of Bézier sections that are connected with C^2 continuity, while the new Control Points of each Bézier curve section are calculated with the aid of the existing B-Spline Control Points. Vice versa, a set of Bézier curves can be joined together to

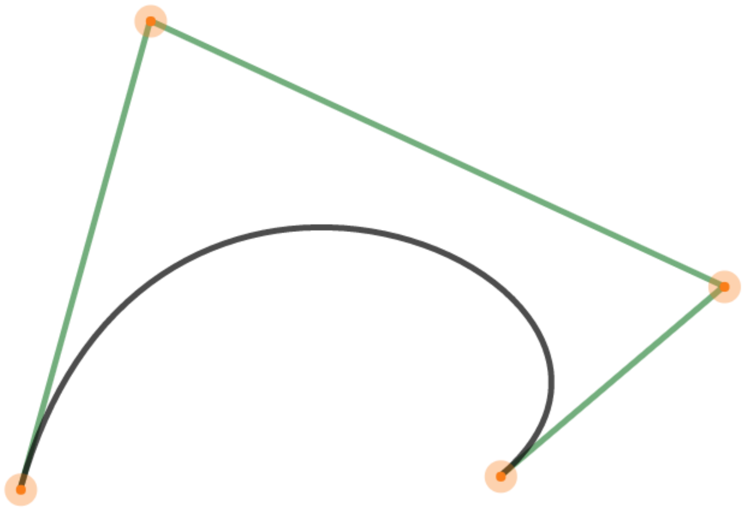


Figure 1.3: Bézier curve of polynomial degree p=3, created with 4 Control Points[†]

form a single B-Spline curve. The prerequisites for this merge are for the Bézier curves to have the same polynomial degree and C^2 continuity between them, while the B-Spline Control Points are calculated from the existing Control Points of the Bézier curves.

1.4.1 Knot Value Vector

A knot value vector is defined as a sequence of non-diminishing numbers. It is a generalization of the parent element [-1, 1] that defines the parameter space in both isoparametric Finite and Bézier elements. At this point, it is important to distinguish between the terms *Knot Value Vector* and *Knot Vector* that will be used for the definitions from now on.

Knot Value Vector: Non decreasing sequence of parametric coordinates that define the parametric space.

Knot Vector: The subset of unique values contained in the *Knot Value Vector*.

An example to clarify the difference between the two vectors is provided below for polynomial degree equal to 3.

Knot Value Vector: $\quad \Xi = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$

Knot Vector: $\Xi' = \{0, 1, 2, 3, 4\}$

In the example of the *Knot Value Vector*, the initial and the final values are repeated *p+1* times, where p is the polynomial degree of the shape functions. This repetition indicates the edges of the parametric domain. The Knot Value Vectors that posseses this property are called *Open Knot Value Vectors*. Another category of *Knot Value Vectors* are the *Uniform Knot Value Vectors*. The latter are defined as an ascending sequence of equidistant parametric values. Even though the definition of an *Open Uniform Knot Value Vector* seems to be a contradicting term based on the latter two definitions, in scientific literature this terms is used to describe the internal values of the sequence, excluding the initial and final values along with their multiplicity.

1.4.2 B-Spline spaces

The definition of B-Spline entities and their NURBS generalizations is based on three spaces.

- Cartesian space
- Parameter space
- Index space

The Cartesian space is the best perceivable space among the three, as curves, surfaces and solid entities are represented in it. On the other hand, Parameter and Index space are based on the parametric nature of B-Spline objects and thus will be explained in detail in the following subsections.

Parameter space

Parameter space is the space that defines the parametric domain of an isogeometric entity. Complex Cartesian geometries such as curves, surfaces or solids are transformed into line segments, rectangles and cuboids respectively. The parametric axes ξ, η, ζ are defined with the aid of the equivalent Knot Vectors per parametric direction. Specifically, given a *Knot Vector* Ξ' it's minimum and maximum values represent the edges of the domain while internal values define the Knots. In case of B-Splines entities of higher dimensions, univariate Knots are extended throughout the rest of the parametric domain for all remaining axes and form knot lines, according to the full tensor product property.

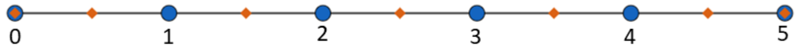


Figure 1.4: Parameter space defined by the Knot Value vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$. Blue circle indicate the Knots, while orange rhombi the parametric positions of the Control Points.

Index space

Index space is an auxiliary space of the B-Spline design technology. It is mainly utilized to represent the values of the *Knot Value Vector* in equidistant positions, regardless of their multiplicity. It mainly serves as a tool to determine the support of shape functions and the parametric coordinates of the Control Points as the middle of the shape function support.

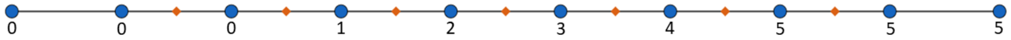


Figure 1.5: Index space defined by the Knot Value vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$. Blue circle indicate the Knots, while orange rhombi the parametric positions of the Control Points.

1.4.3 Control Points

In computer aided design, Control Points are defined as the Cartesian points that determine the shape of the geometry. They were initially introduced in Bézier curves, where a multitude of p+1 Cartesian points are required to define it, p being the polynomial degree. From this set of Cartesian points, the initial and the final one are interpolatory to the curve. As for the remaining ones it is not necessary for them to lie on the geometry.

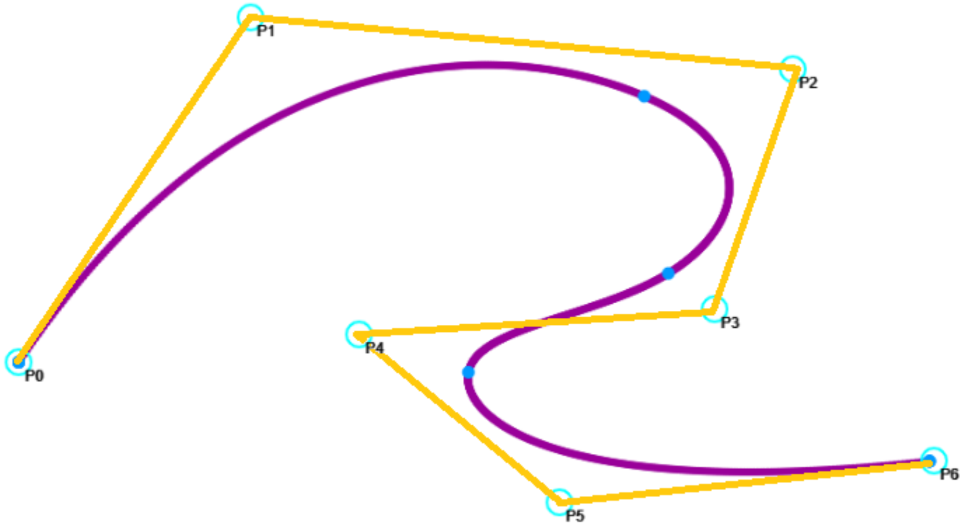


Figure 1.6: Quadratic B-Spline curve with Knot Value Vector $\Xi = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$. [‡]

Figure 1.6 provides an example of a quadratic B-Spline curve, generated based on the Knot Value Vector $\Xi = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$. A multitude of 7 cyan circles representing the Control Points are needed to define the curve. The Control Points are interconnected with a yellow polygonal line representing the Control polygon of the curve. As already stated at the initial point P_0 and final point P_6 the curve is interpolatory and tangential to the Control polygon. In addition, 5 blue dots that lie on the curve represent the Cartesian positions of the Knots $\Xi' = \{0, 0.25, 0.5, 0.75, 1\}$ that define the discretization of the curve into piecewise polynomial parts.

Similar to Bézier curves, B-Splines are also defined by a set of Control Points. Assuming a B-Spline curve of degree p, defined by an Open Uniform Knot Value Vector that contains k values, the necessary number of Control Points to create it are n=k-p-1. There is an one-to-one equivalence between Control Points and the shape functions of the curve in both B-Spline and Bézier curves. In addition, by utilizing the Greville abscissae, the parameteric coordinates of the Control Points can be computed

$$\xi_i^{CP} = \frac{\sum_{k=1}^{p} \xi_{i+1}}{p} \tag{1.11}$$

[‡] Created with UTCS-BSpline Demo

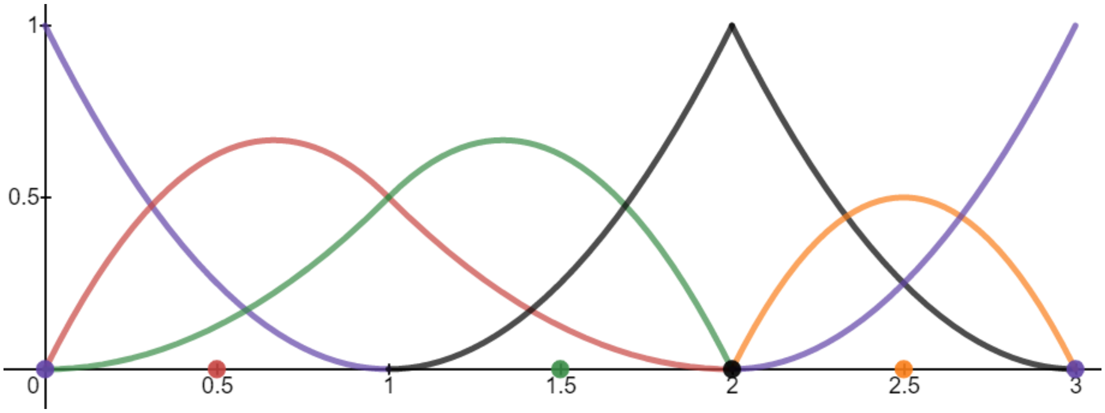


Figure 1.7: Quadratic B-Spline shape function with Knot Value Vector $\Xi = \{0, 0, 0, 1, 2, 2, 3, 3, 3\}$. Points that lie on the parametric axis ξ show the parametric coordinates of the Control Points. Color equivalence maps shape function with respective Control Point. [§]

Figure 1.7 illustrates the equivalence between shape functions and Control Points in the parametric space. The number shape function defined by the Knot Value vector $\Xi = \{0, 0, 0, 1, 2, 2, 3, 3, 3\}$ is $n = k - p - 1 = 9 - 2 - 1 = 6$, where k is the number of Knot Value and p the polynomial degree. Utilizing eq. 1.11 the parametric coordinates of the Control Points are derived which are $[0, 0.5, 1.5, 2, 2.5, 3]$. Both shape functions and Control Points are color coded to showcase their one-to-one equivalence, while as it can be observed the parametric position of each control point coincides with the maximum value of its equivalent shape function.

1.4.4 B-Spline shape functions

Given a Knot Value Vector $\Xi = \{\xi_1, \xi_2, ..., \xi_i, ..., \xi_{n+p+1}\}$ with *n+p+1* values, where p is the polynomial degree of the curve and n the number of Control Points that define its geometry, the shape function are computed recursively, with the aid of Cox-de Boor recursive formula [54, 58] starting with piecewise constant shape functions

$$N_{i,0}(\xi) = \begin{cases} 0, & if\ \xi_i \leq \xi < \xi_{i+1} \\ 1, & otherwise \end{cases} \tag{1.12}$$

[§] Created with Desmos

based on the formerly evaluated constant functions, piecewise functions of higher degree are evaluated with the assumption that $\frac{0}{0} \doteq 0$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \cdot N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p} - \xi_i} \cdot N_{i+1,p-1}(\xi) \tag{1.13}$$

1.4.5 Multi-variate B-Splines

B-Spline shape functions abide to a full tensor product nature. This means that for the evaluation of multivariate B-Splines a multitude of univariate shape functions can be combined. In addition, all properties of univariate B-Splines are retained in multivariate cases.

Two-dimensional B-Splines $R_{i,j}^{p,q}(\xi, \eta)$ are calculated as a tensor product of two univariate shape functions $N_i^p(\xi)$ and $M_j^q(\eta)$ of parametric axes ξ and η equivalently.

$$R_{i,j}^{p,q}(\xi, \eta) = N_i^p(\xi) \cdot M_j^q(\eta) \tag{1.14}$$

Similarly, three-dimensional B-Splines are calculated as a tensor of three univariate shape functions $N_i^p(\xi)$, $M_j^q(\eta)$ and $L_k^r(\zeta)$ of parametric axes ξ, η and ζ equivalently.

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = N_i^p(\xi) \cdot M_j^q(\eta) \cdot L_k^r(\zeta) \tag{1.15}$$

1.4.6 B-Spline shape function properties

According to [143], B-Spline shape functions possess the following properties.

1. Local Support

   Local support is a result of the Cox-de Boor recursive algorithm and means that shape functions are non-zero in certain knot spans of the parameter space. In mathematical notation this is expressed by

$$N_i^p(\xi) = 0 \qquad \forall \xi \notin [\xi_i, \xi_{i+p+1}] \tag{1.16}$$

   Specifically, observing Cox-de Boor recursive algorithm, it is apparent that for the definition of a univariate shape function of degree p, two consecutive functions of degree *p-1* are needed. In a similar fashion, the definition of two consecutive functions of *p-1* polynomial degree, requires three functions of degree *p-2*. Inductively, *p+1* constant functions are needed for the evaluation of a shape function of degree *p*. Since each constant function has support of one knot span, the support of shape function of degree p is the union of all constant B-Splines it depends on, hence *p+1* consecutive knot span. This dependency of higher degree shape functions from lower degree ones is depicted in Figure 1.8, where it can be seen that four constant shape functions, namely $N_5^0(\xi), N_6^0(\xi), N_7^0(\xi), N_8^0(\xi)$ are required for the definition of the support of the third degree shape function $N_5^3(\xi)$.

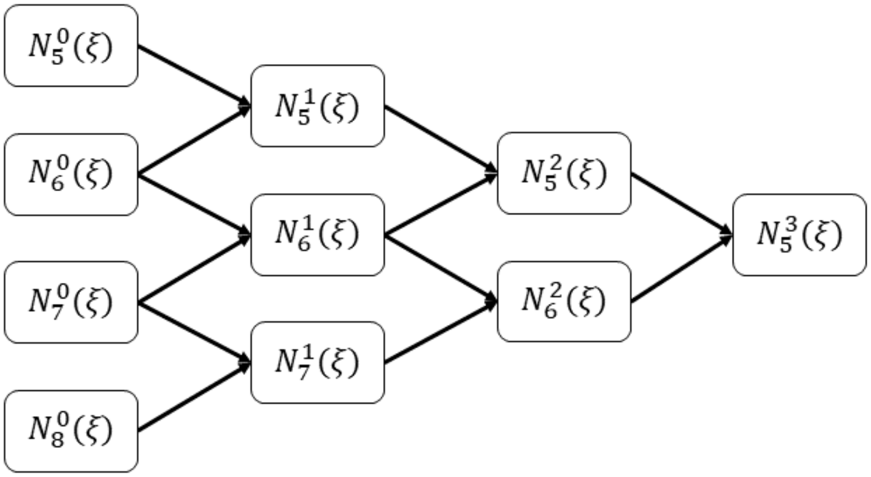


Figure 1.8: Shape functions of lower degree required for the creation of $N_5^3(\xi)$

2. Non-negativity

$$N_i^p(\xi) \geq 0 \tag{1.17}$$

3. Partition of unity

$$\sum_{i=1}^{n} N_i^p(\xi) = 1 \tag{1.18}$$

4. C^{p-m} continuity across Knots with multiplicity m.

   This property is a product of the Cox-de Boor recursive formula along with the multiplicity of the Knots. Specifically, given a Knot Value Vector that defines the parametric space of a B-Spline curve, the equivalent Knot Vector can be extracted. The multiplicity m of a knot, is the number of times this value appears in the Knot Value Vector. If the multiplicity of a knot is greater than one, then the shape function has C^{p-m} continuity, which means that at this point the shape functions has p-m continuous derivatives. The minimum continuity observed in case of an Open Uniform Knot Value vectors is m=p+1 for the initial and the final Knots, p being the polynomial degree. This translates to $C^{p-m} = C^{-1}$ continuity that indicates the edges of the parametric domain. In case of internal Knots, continuity less than C^0 is not acceptable. As a result, knot values can be repeated at most p times. Note that as continuity decreases, shape functions tend to become steeper.

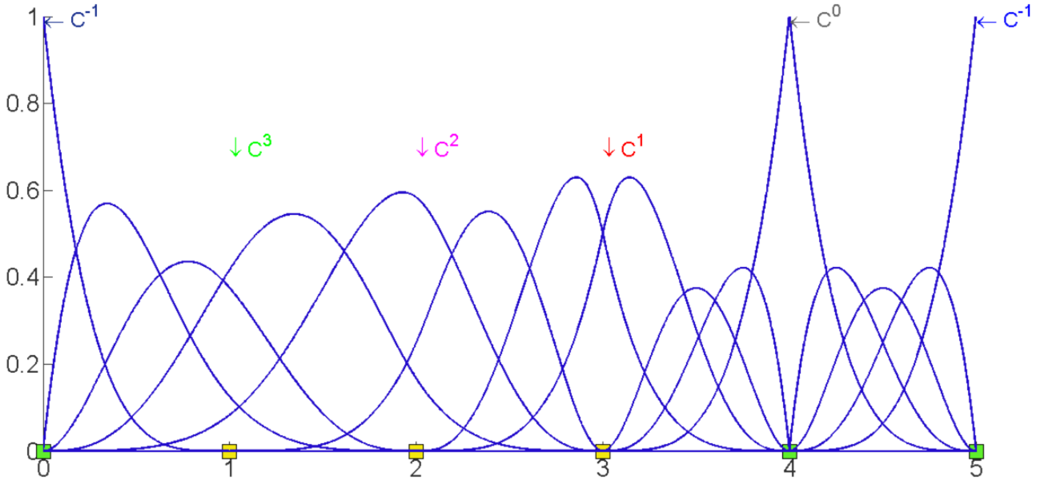


Figure 1.9: B-Spline shape functions continuity across a parametric domain defined by the Knot Value Vector $\Xi = \{0, 0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5\}$

Figure 1.9 provides an example of B-Spline shape functions with varying continuity at each knot. The initial and final knot are repeated p+1 times thus indicating the ends of the parametric domain. Internal Knots are displayed with continuity ranging from C^3 to C^0 depending on the multiplicity of the equivalent knot value.

The dependency of the shape function smoothness on the knot multiplicity is also apparent in this figure, as the higher the multiplicity the lower the smoothness of the function, with C^0 being the extreme case where kinks appear.

These properties of B-Spline shape function were extensively discussed in [162, 54, 58, 149, 59].

1.4.7 B-Spline derivatives

Apart from the shape function that are utilized for the generation of the curve geometry, their derivatives are also widely used in computational geometry and computational mechanics applications. As a result, this subsection will provide the basic of B-Spline derivatives that serves as a basis for advanced spline technologies that will be presented in the following sections. Similar to B-Spline basis functions, B-Spline first derivative is obtained from the following recursive formula:

$$\frac{d}{d\xi}N_i^p(\xi) = \frac{p}{\xi_{i+p} - \xi_i} \cdot N_i^{p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} \cdot N_{i+1}^{p+1}(\xi) \tag{1.19}$$

this equation can be generalized for the k^{th} derivative as follows:

$$\frac{d^k}{d\xi^k}N_i^p(\xi) = \frac{p!}{(p-k)!} \cdot \sum_{j=0}^{k} a_{k,j} \cdot N_{i+j}^{p-k}(\xi) \tag{1.20}$$

where the terms $a_{k,j}$ are given by:

$$\begin{aligned}
a_{0,0} &= 1 \\
a_{k,0} &= \frac{a_{k-1,0}}{\xi_{i+p-k+1} - \xi_i} \\
a_{k,j} &= \frac{a_{k-1,j} - a_{k-1,j-1}}{\xi_{i+p+j-k+1} - \xi_i}, \qquad forj = 1, ..., k-1 \\
a_{k,k} &= \frac{-a_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}}
\end{aligned} \tag{1.21}$$

In case of multivariate B-Splines, their partial derivatives are obtained by applying the quotient rule.

$$\begin{aligned}
\frac{\partial}{\partial \xi} R_{i,j}^{p,q}(\xi,\eta) &= \left(\frac{d}{d\xi} N_i^p(\xi)\right) \cdot M_j^q(\eta) \\
\frac{\partial}{\partial \eta} R_{i,j}^{p,q}(\xi,\eta) &= N_i^p(\xi) \cdot \left(\frac{d}{d\eta} M_j^q(\eta)\right)
\end{aligned} \tag{1.22}$$

Derivatives of trivariate shape functions can be obtained in a similar manner.

$$\begin{aligned}
\frac{\partial}{\partial \xi} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) &= \left(\frac{d}{d\xi} N_i^p(\xi)\right) \cdot M_j^q(\eta) \cdot L_k^r(\zeta) \\
\frac{\partial}{\partial \eta} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) &= N_i^p(\xi) \cdot \left(\frac{d}{d\eta} M_j^q(\eta)\right) \cdot L_k^r(\zeta) \\
\frac{\partial}{\partial \zeta} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) &= N_i^p(\xi) \cdot M_j^q(\eta) \cdot \left(\frac{d}{d\zeta} L_k^r(\zeta)\right)
\end{aligned} \tag{1.23}$$

1.4.8 B-Spline Geometries

As already described for Bézier curves, B-Splines geometries can be created by combining B-Spline shape functions with their corresponding Control Points. As a result, curves, surfaces and solid parametric geometries can be generated using the detailed processes described below.

B-Spline curves

A p^{th} degree B-Spline curve is defined by

$$C(\xi) = \underset{(1xn)}{\{N_i^p(\xi)\}^T} \underset{(nx3)}{\{P_i\}} = \sum_{i=1}^{n} N_i^p(\xi) \underset{(1x3)}{\{P_i\}}, \quad \xi_1 \leq \xi \leq \xi_{n+p+1} \tag{1.24}$$

where $\{P_i\}$ are the Control Points' Cartesian coordinates of the curve and $\{N_i^p(\xi)\}$ the p^{th} degree B-Spline basis functions defined by the open Knot Value vector $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$

B-Spline surfaces

A B-Spline surface is obtained by taking a bidirectional net of Control Points, two Knot Value vectors Ξ and H with respective polynomial degrees p, q and the tensor products of the univariate B-Spline functions:

$$S(\xi,\eta)=\sum_{i=1}^{n}\sum_{j=1}^{m}R_{i,j}^{p,q}(\xi,\eta)\underset{(1x3)}{\{P_{i,j}\}}=\sum_{i=1}^{n}\sum_{j=1}^{m}N_i^p(\xi)\cdot M_j^q(\eta)\underset{(1x3)}{\{P_{i,j}\}} \tag{1.25}$$

B-Spline solids

In the same fashion, a B-Spline volume is obtained by taking a three-dimensional net of Control Points, three Knot Value Vectors, Ξ, H, Z, with respective polynomial degrees p, q, r and the products of the univariate B-Spline functions:

$$\begin{aligned}(x,y,z)=V(\xi,\eta,\zeta)&=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l}R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)\underset{(1x3)}{\{P_{i,j,k}\}}\\&=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l}N_i^p(\xi)\cdot M_j^q(\eta)\cdot L_k^r(\zeta)\underset{(1x3)}{\{P_{i,j,k}\}}\end{aligned} \tag{1.26}$$

B-Spline geometry properties

B-Spline geometries abide to the following properties:

1. B-Spline curves are a generalization of Bézier curves.
2. B-Splines are piecewise polynomial curves.
3. Each basis function corresponds to a certain Control Point.
4. The first and the last Control Point as well as internal Control Points corresponding to C^0 continuous basis functions are interpolatory to the curve.
5. Moving a Control Point only affects part of the curve.
6. The Control Polygon is a piecewise linear approximation of the curve.

7. Any transformation applied to the curve can be applied directly at the Control Points.

1.5 NURBS

Existing CAD technologies such as Bézier and B-Spline curves allowed designers to create complex geometries while having the ability to locally control the geometry. Yet, the constantly increasing need of designers to accurately reproduce even more complex shapes rendered these technologies unable to cope with the demands. Even though linear or parabolic curves are precisely described by B-Splines, other conic sections such as circle, ellipse or hyperbole are only approximated. To remedy this drawback, Non Uniform Rational B-Splines were introduced [180, 175, 142]. The rational curves that NURBS are based on, are introduced with the incorporation of a weight for each of the Control Points that define the geometry. This transforms the non-homogeneous coordinate space (X,Y,Z) of Bézier and B-Splines curves to a homogeneous coordinate space (X,Y,Z,W) thus introducing a projection.

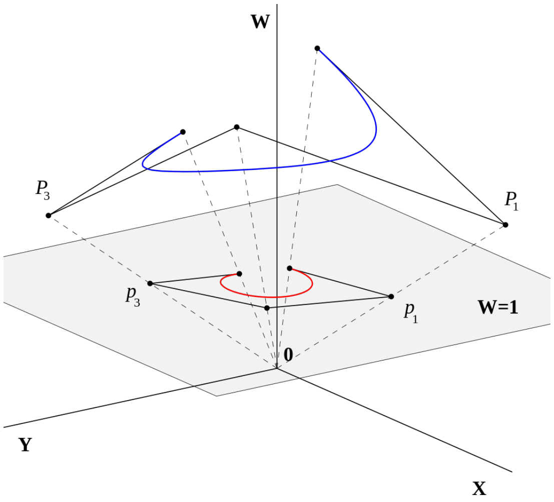


Figure 1.10: Projection of a non-rational B-Spline curve to the W=1 plane to create a Rational B-Spline. [¶]

[¶] Source: Wikimedia Commons

Figure 1.10 provides an example of the projection of a curve from $\mathbb{R}^3$ to $\mathbb{R}^2$. Specifically, a non-rational 3D curve $C^W(\xi)$ is defined with 3D Control Points $P^W = \{X^W, Y^W, Z^W\}$. The rational curve is calculated by projecting both the non-rational curve and the projective Control Points onto the plane W=1, thus producing the equivalent NURBS curve $C(\xi)$ and the 2D Control Points. The coordinate Z^W of the projective Control Points define the weights w_i of the resulting NURBS curve, while the 2D Control Points coordinates are defined by

$$P_i = \{X_i, Y_i\} = \left\{ \frac{X_i^W}{Z_i^W}, \frac{Y_i^W}{Z_i^W} \right\} \tag{1.27}$$

This concept can be generalized for any geometries of arbitrary dimensions. Thus, in order to create an d-dimensional Rational B-Spline, a *(d+1)* non-rational equivalent is required, hence defining the projection $\mathbb{R}^{d+1} \rightarrow \mathbb{R}^d$.

1.5.1 NURBS Shape functions

In a similar fashion to B-Splines, in order to compute the shape functions of a NURBS curve, a Knot Value Vector and the polynomial degree are needed along with the weights w_i of the Control Points.

$$R_i^p(\xi) = \frac{N_i^p(\xi) \cdot w_i}{\sum_{j=0}^{n} N_j^p(\xi) w_i} \tag{1.28}$$

where $N_i^p(\xi)$ are the equivalent p^{th} degree B-Spline shape functions. The denominator is called weighting function $W(\xi)$ and is defined as:

$$W(\xi) = \sum_{j=0}^{n} N_j^p(\xi) w_i \tag{1.29}$$

As a generalization of B-Splines, NURBS maintain all of their properties, most significant among them, the full tensor product nature. As a result, NURBS shape functions of higher dimensions are calculated as tensor products of univariate NURBS functions. Specifically, for the two-dimensional case

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_i^p(\xi) \cdot M_j^q(\eta) \cdot w_i j}{\sum_{i'=1}^{n} \sum_{j'=1}^{m} \left\{ N_{i'}^p(\xi) \cdot M_{j'}^q(\eta) \cdot w_{i'j'} \right\}} \tag{1.30}$$

and the equivalent weighting function:

$$W(\xi, \eta) = \sum_{i'=1}^{n} \sum_{j'=1}^{m} \left\{ N_{i'}^{p}(\xi) \cdot M_{j'}^{q}(\eta) \cdot w_{i'j'} \right\} \tag{1.31}$$

Similarly, the tensor product is expanded for the three-dimensional case.

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{i}^{p}(\xi) \cdot M_{j}^{q}(\eta) \cdot L_{k}^{r}(\zeta) \cdot w_{i}jk}{\sum_{i'=1}^{n} \sum_{j'=1}^{m} \sum_{k'=1}^{l} \left\{ N_{i'}^{p}(\xi) \cdot M_{j'}^{q}(\eta) \cdot L_{k'}^{r} \cdot w_{i'j'k'} \right\}} \tag{1.32}$$

the weighting function is now defined as:

$$W(\xi, \eta, \zeta) = \sum_{i'=1}^{n} \sum_{j'=1}^{m} \sum_{k'=1}^{l} \left\{ N_{i'}^{p}(\xi) \cdot M_{j'}^{q}(\eta) \cdot L_{k'}^{r} \cdot w_{i'j'k'} \right\} \tag{1.33}$$

1.5.2 NURBS Shape function derivatives

By application of the quotient rule to eqs.1.28-1.33 the derivatives of the NURBS shape functions are calculated. For the 1D case, the first derivatives are obtained by

$$\frac{d}{d\xi} R_{i}^{p}(\xi) = \frac{\left(\frac{d}{d\xi} N_{i}^{p}(\xi) \right) \cdot W(\xi) - \left(\frac{d}{d\xi} W(\xi) \right) \cdot N_{i}^{p}(\xi)}{W(\xi)^{2}} \tag{1.34}$$

where

$$\frac{d}{d\xi} W(\xi) = \sum_{i=1}^{n} \left(\frac{d}{d\xi} N_{i}^{p}(\xi) \right) \cdot w_{i} \tag{1.35}$$

For two-dimensional shape functions:

$$\frac{\partial}{\partial\xi} R_{i,j}^{p,q}(\xi, \eta) = \frac{\left(\frac{d}{d\xi} N_{i}^{p}(\xi) \right) \cdot M_{j}^{q}(\eta) \cdot W(\xi, \eta) - \left(\frac{\partial}{\partial\xi} \xi W(\xi, \eta) \right) \cdot N_{i}^{p}(\xi) \cdot M_{j}^{q}(\eta)}{W(\xi, \eta)^{2}} \cdot w_{ij} \tag{1.36}$$

$$\frac{\partial}{\partial \eta} R_{i,j}^{p,q}(\xi,\eta) = \frac{N_i^p(\xi) \cdot \left(\frac{d}{d\eta} M_j^q(\eta)\right) \cdot W(\xi,\eta) - \left(\frac{\partial}{\partial \eta} W(\xi,\eta)\right) \cdot N_i^p(\xi) \cdot M_j^q(\eta)}{W(\xi,\eta)^2} \cdot w_{ij} \tag{1.37}$$

$$\frac{\partial}{\partial \xi} W(\xi,\eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\frac{d}{d\xi} N_i^p(\xi)\right) \cdot M_j^q(\eta) \cdot w_{ij} \tag{1.38}$$

$$\frac{\partial}{\partial \eta} W(\xi,\eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_i^p(\xi) \cdot \left(\frac{d}{d\eta} M_j^q(\eta)\right) \cdot w_{ij} \tag{1.39}$$

Equivalently, first derivatives of three-dimensional shape functions are evaluated as follows:

$$\begin{aligned} \frac{\partial}{\partial \xi} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) = & \frac{\left(\frac{d}{d\xi} N_i^p(\xi)\right) \cdot M_j^q(\eta) \cdot L_k^r(\zeta) \cdot W(\xi,\eta,\zeta)}{W^2(\xi,\eta,\zeta)} - \\ & \frac{\left(\frac{\partial}{\partial \xi} W(\xi,\eta,\zeta)\right) \cdot N_i^p(\xi) \cdot M_j^q(\eta) \cdot L_k^r(\zeta)}{W^2(\xi,\eta,\zeta)} \cdot w_{ijk} \end{aligned} \tag{1.40}$$

$$\begin{aligned} \frac{\partial}{\partial \eta} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) = & \frac{N_i^p(\xi) \cdot \left(\frac{d}{d\eta} M_j^q(\eta)\right) \cdot L_k^r(\zeta) \cdot W(\xi,\eta,\zeta)}{W^2(\xi,\eta,\zeta)} - \\ & \frac{\left(\frac{\partial}{\partial \eta} W(\xi,\eta,\zeta)\right) \cdot N_i^p(\xi) \cdot M_j^q(\eta) \cdot L_k^r(\zeta)}{W^2(\xi,\eta,\zeta)} \cdot w_{ijk} \end{aligned} \tag{1.41}$$

$$\frac{\partial}{\partial \zeta} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) = \frac{N_i^p(\xi) \cdot M_j^q(\eta) \cdot \left(\frac{d}{d\zeta} L_k^r(\zeta)\right) \cdot W(\xi,\eta,\zeta)}{W^2(\xi,\eta,\zeta)} - \frac{\left(\frac{\partial}{\partial \zeta} W(\xi,\eta,\zeta)\right) \cdot N_i^p(\xi) \cdot M_j^q(\eta) \cdot L_k^r(\zeta)}{W^2(\xi,\eta,\zeta)} \cdot w_{ijk} \tag{1.42}$$

For the sake of plenitude of the thesis the second derivatives of two-dimensional NURBS shape functions are also provided, since they are utilized in Isogeometric Analysis formulations in the following chapter.

$$\frac{\partial^2}{\partial \xi^2} R_{i,j}^{p,q}(\xi,\eta) = \left(\frac{\frac{d^2}{d\xi^2} N_i^p(\xi)}{W(\xi,\eta)} - 2\frac{\frac{d}{d\xi} N_i^p(\xi) \cdot \frac{\partial}{\partial \xi} W(\xi,\eta)}{W^2(\xi,\eta)} - \frac{N_i^p(\xi) \cdot \frac{\partial^2}{\partial \xi^2} W(\xi,\eta)}{W^2(\xi,\eta)} + 2\frac{N_i^p(\xi) \cdot \left(\frac{\partial}{\partial \xi} W(\xi,\eta)\right)^2}{W^3(\xi,\eta)} \right) \cdot M_j^q(\eta) \cdot w_{ij} \tag{1.43}$$

$$\frac{\partial^2}{\partial \eta^2} R_{i,j}^{p,q}(\xi,\eta) = N_i^p(\xi) \cdot \left(\frac{\frac{d^2}{d\eta^2} M_j^q(\eta)}{W(\xi,\eta)} - 2\frac{\frac{d}{d\eta} M_j^q(\eta) \cdot \frac{\partial}{\partial \eta} W(\xi,\eta)}{W^2(\xi,\eta)} - \frac{M_j^q(\eta) \cdot \frac{\partial^2}{\partial \eta^2} W(\xi,\eta)}{W^2(\xi,\eta)} + 2\frac{M_j^q(\eta) \cdot \left(\frac{\partial}{\partial \eta} W(\xi,\eta)\right)^2}{W^3(\xi,\eta)} \right) \cdot w_{ij} \tag{1.44}$$

$$\frac{\partial^2}{\partial\xi\partial\eta}R_{i,j}^{p,q}(\xi,\eta) = \Bigg(\frac{\frac{d}{d\xi}N_i^p(\xi)\frac{d}{d\eta}M_j^q(\eta)}{W(\xi,\eta)} - \frac{\frac{d}{d\xi}N_i^p(\xi)M_j^q(\eta)\frac{\partial}{\partial\eta}W(\xi,\eta)}{W^2(\xi,\eta)} - \frac{N_i^p(\xi)\frac{d}{d\eta}M_j^q(\eta)\frac{\partial}{\partial\xi}W(\xi,\eta)}{W^2(\xi,\eta)} - \frac{N_i^p(\xi)M_j^q(\eta)\frac{\partial^2}{\partial\xi\partial\eta}W(\xi,\eta)}{W^2(\xi,\eta)} + 2\frac{N_i^p(\xi)M_j^q(\eta)\frac{\partial}{\partial\xi}W(\xi,\eta)\frac{\partial}{\partial\eta}W(\xi\eta)}{W^2(\xi,\eta)} \Bigg) \cdot w_{ij} \tag{1.45}$$

where

$$\frac{\partial^2}{\partial\xi^2}W(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{d^2}{d\xi^2}N_i^p(\xi)\right)\cdot M_j^q(\eta)\cdot w_{ij} \tag{1.46}$$

$$\frac{\partial^2}{\partial\eta^2}W(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m}N_i^p(\xi)\cdot\left(\frac{d^2}{d\eta^2}M_j^q(\eta)\right)\cdot w_{ij} \tag{1.47}$$

$$\frac{\partial^2}{\partial\xi\partial\eta}W(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{d}{d\xi}N_i^p(\xi)\right)\cdot\left(\frac{d}{d\eta}M_j^q(\eta)\right)\cdot w_{ij} \tag{1.48}$$

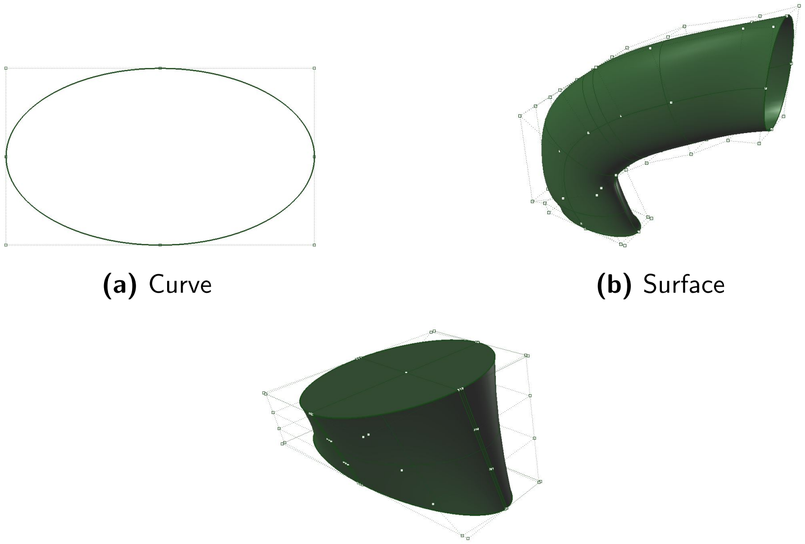
1.5.3 NURBS Geometries

NURBS geometries are created by utilizing NURBS shape function R in combination with weighted Control Points P, in similar way to B-Spline entities. The following equations provide the formulas for the creation of NURBS curves, surfaces and volumes.

$$C(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\cdot P_i \tag{1.49}$$

$$S(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\cdot P_{ij} \tag{1.50}$$

$$V(\xi,\eta,\zeta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)\cdot P_{ijk} \tag{1.51}$$



(a) Curve

(b) Surface

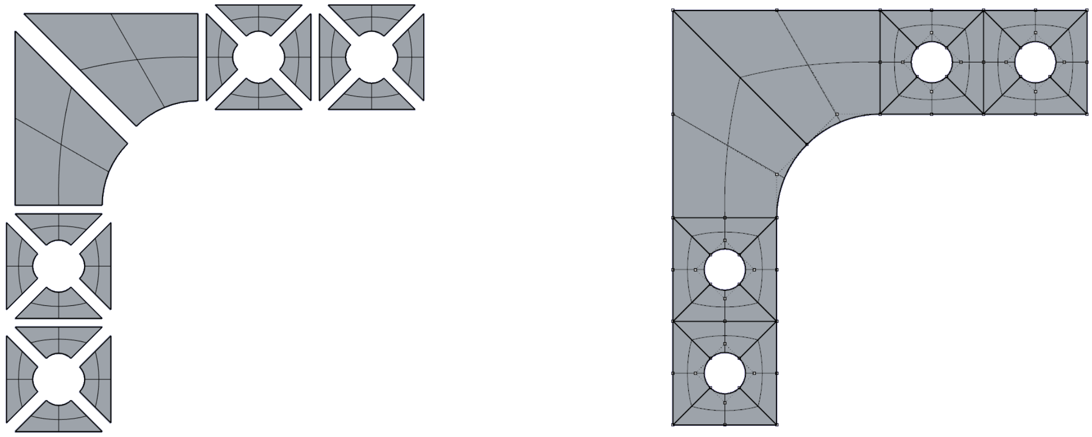
(c) Solid

Figure 1.11: Nurbs entities

1.5.4 NURBS Patches

During the design of a complex structure, several issues arise with the ability of B-Spline and NURBS design technologies to describe them. The most common issue is that the whole geometry cannot be described with the use of a single line, rectangle or cuboid in parameter space. To circumvent this, a multitude of such parameter spaces have to be combined in order to design efficiently and accurately the desired geometry. Each independent geometry defined only by a single parameter space is called a Patch. Since each of the individual NURBS patches have different attributes, it is uncommon for their common edges to coincide. As a result, their properties have to be revised in order for the patches to be conforming.

Figure 1.12a provides an example of a complicated geometry designed with 18 separate patches. Figure 1.12b illustrates the same patches merged into a single entity. In this case, special care was taken for the coinciding edges to have the same polynomial degree and Knot Value Vector in order to achieve a seamless interconnection. Unfortunately, in most cases, even this treatment is not sufficient to ensure a watertight connection, thus a new design technology had to be introduced.



(a) Separate patch geometries

(b) Single merged entity

Figure 1.12: Nurbs conforming patches before and after merging into a single CAD entity.

1.6 T-Splines

NURBS is an established CAD technology that has been extensively researched by the computational geometry industry and is widely used by designers. Yet, this design technology is accompanied by several drawbacks, the most profound of which is that they achieve only C^0 across patch boundaries. In case two NURBS patches do not share a common boundary curve, even the aforementioned C^0 is not attainable thus making the coupling laborious as several knot insertions to a Knot Value Vector of the common edge are required to render them compatible. The knot insertion operation in a two- or higher-dimensional patch is a procedure that propagates throughout the domain. This results to an addition of a series of Control Points adding a significant burden to the geometrical modeling procedure. In order to circumvent these deficiencies, T-Splines technology was introduced [170, 169, 168] to enable local refinement and water-tight connection of patches.

1.6.1 T-Spline spaces

In contrast to B-Splines and NURBS shape functions that are defined based on global Knot Value Vectors, T-Splines introduce a local Knot Value Vectors for each shape function, which are inferred from a global structure, thus rendering them independent of each other. In contrast to B-Splines and NURBS where parameter space was utilized

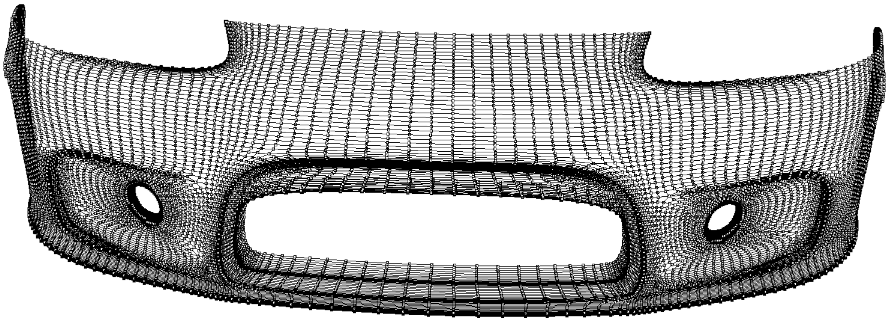


Figure 1.13: Example of a T-Spline object.

for the calculation of shape functions and index space was auxiliary, this order is now reversed.

Index space - T-mesh

For two-dimensional entities, the Index space is represented as a rectangular tiling of $\mathbb{R}^2$, that permits vertices with multitude converging edges not equal to four. These vertices are named T-junction due to their shape. In contrast to NURBS, the Index space plays a major role in T-Splines as all basic variables, such as Knot Value Vectors and parametric Control Points, needed for the computation of the shape functions are calculated here. Note that lines in the T-mesh correspond to knot indices. Given a T-mesh, a polynomial degree for each parametric direction is then selected.

Figure 1.14 provides an example of such a T-mesh. it is apparent that the tensor product nature of B-Splines and NURBS, is no longer present in T-Splines and as a result the calculation of the shape functions is no longer based on a global Knot Value Vector for each axis. On the contrary, local Knot Value Vectors are derived from the T-mesh layout. Similarly, the parametric positions of the Control Points called Anchors are found indirectly as the center of the equivalent basis function support.

Parameter space

Similar to NURBS, the parameter space of T-Splines is created by collapsing spans of each parametric direction with zero distance. As a result,the T-mesh structure of

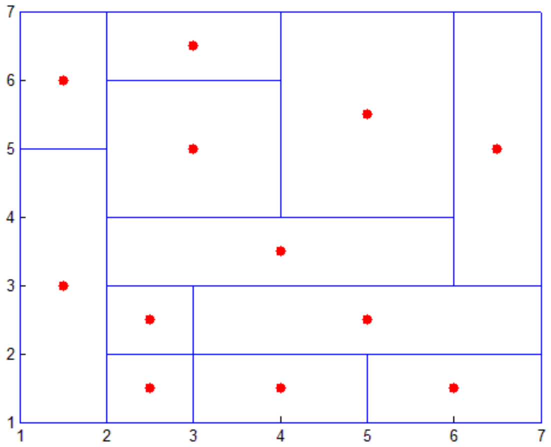


Figure 1.14: T-mesh of a T-Spline. Positions of the Control Points are depicted in red circles.

the index space is retained, yet some knot lines are merged. The final layout of the T-junctions appearing in the parameter space, provides the final Knots of the T-Spline.

Figure 1.15 provides an example of the T-Spline parameter space configuration. In addition to the Anchors positions found in the Index space, the Parameter space also displays the positions of the Knots at each remaining T-junction vertex of the parametric domain.

1.6.2 Anchor-Control Point

In order to define the support of each shape function and the local knot vector accordingly, the parametric positions of the Control Points called Anchors must be defined. As illustrated in Figure 1.16, anchors lie on different positions of the T-mesh, depending on the polynomial degree chosen for each parametric axis. In case of odd polynomials degree for axes, Anchors lie on the vertices defined by the T-mesh. When both odd and even degrees are present, Anchors lie on the center of horizontal or vertical line segment. Finally, when even degree are chosen for all axes, Anchors lie on the center of rectangles.

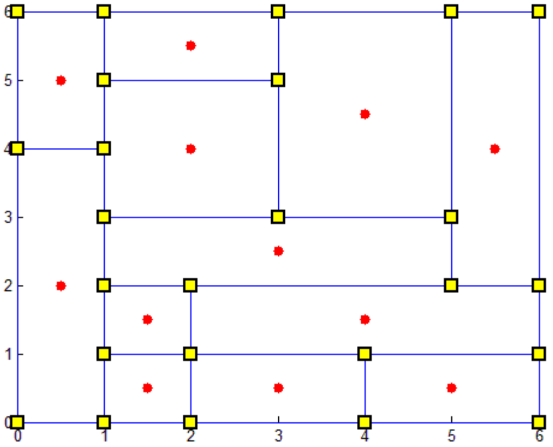
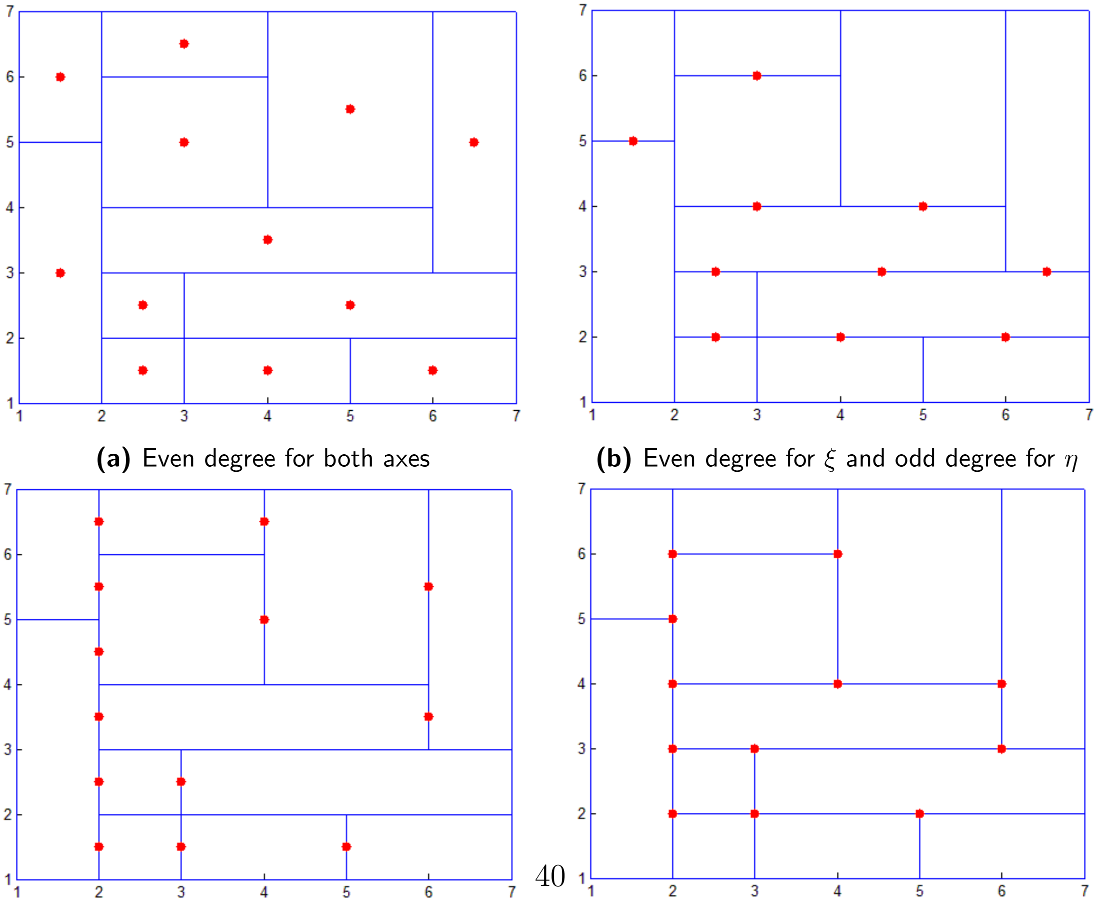


Figure 1.15: Parameter space of a T-Spline. Anchor are depicted in red circles and T-junctions as yellow rectangles.



(a) Even degree for both axes

(b) Even degree for ξ and odd degree for η

(c) Odd degree for ξ and even degree for η

(d) Odd degree for both axes

Figure 1.16: Examples of Anchor positions for odd/even degrees.

1.6.3 Local Knot Vectors

Generally, every Anchor contains information about the local Knot Value Vectors of each shape function as they are directly related. In order to acquire the local Knot Value Vector *p+2* value have to be computed. Similar to the definition of Anchor positions, special cases for odd/even polynomial degrees must be taken into account.

In case of odd polynomial degree for both axes and given an Anchor $s_a = (\xi_a, \eta_a)$ that lies on a vertex of the T-mesh, the local Knot Value Vector for axis ξ, Ξ_a is computed from the following procedure. Initially, the length of the Knot Value Vector is defined as *p+2*, which is an odd number as the polynomial degree p is also odd. The middle position of the vector is occupied by the Anchor's parametric coordinate ξ_a. In order to fill the first half of the Knot Value Vector, a horizontal line to the left of the Anchor is drawn. The first *(p+1)/2* vertical lines or T-mesh vertices encountered fill in the empty positions of the first half of the vector. Similarly, a horizontal line to the right provides the final *(p+1)/2* values of the vector. Note that some anchors may lie closer to the border of the T-mesh than others. As a result, during the Knot Value Vector calculation procedure, a border may be reached before the required *(p+1)/2* values are completed. In this case, the standard procedure is to repeat the border Knot Value as many times needed in order to complete the remaining positions of the vector. The same procedure is conducted for the computation of the local Knot Value Vector H_a of the parametric direction η. The middle position of the vector is occupied by the Anchor ordinate η_a, while the rest of the positions are completed by moving upwards or downwards until *(p+1)/2* vertices or horizontal lines are encountered.

The procedure is similar in case of even degrees.Initially, the number of values of the local Knot Value Vector Ξ_a are computed as *(p+2)* which is an even number. Since the total number of indices is even, the Anchor coordinates are not taken into account for the computation of the local Knot Value vector. On the contrary, the leftwards and rightwards movements for *(p/2+1)* values for axis ξ and equivalently upwards and downwards for axis η are sufficient for the completion of the local Knot Value vectors.

In Figure 1.17 examples of the computed Knot Value Vectors are illustrated

(a) Even degree in both axes:

Top right anchor: $\Xi_a = \{4, 6, 7, 7\}$ and $H_a = \{2, 3, 7, 7\}$

(b) Odd degree for axis ξ and even for axis η:

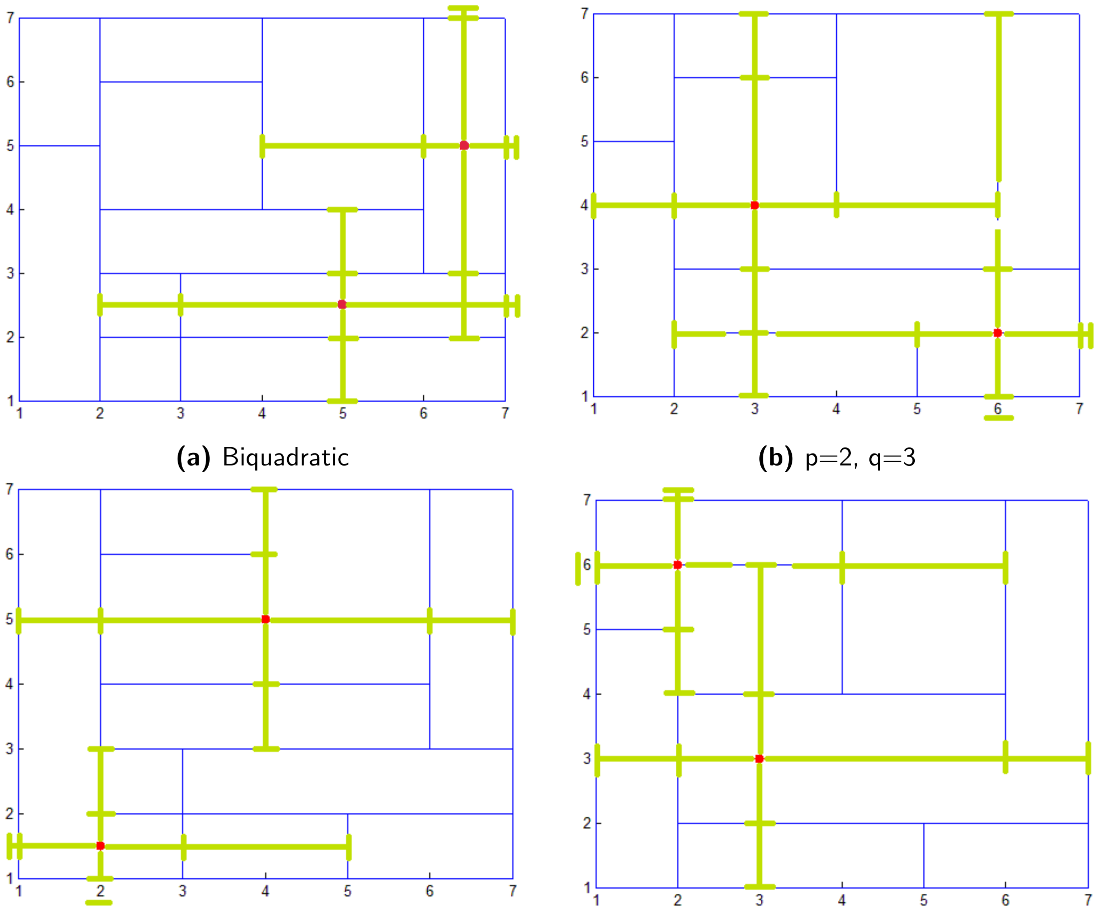
Top right anchor: $\Xi_a = \{1, 2, 6, 7\}$ and $H_a = \{3, 4, 5, 6, 7\}$

(c) Even degree for axis ξ and odd for axis η:

Bottom right anchor: $\Xi_a = \{2, 5, 6, 7, 7\}$ and $H_a = \{1, 1, 3, 7\}$

(d) Odd degree in both axes:

Top left anchor: $\Xi_a = \{1, 1, 2, 4, 6\}$ and $H_a = \{4, 5, 6, 7, 7\}$



(a) Biquadratic

(b) p=2, q=3

(c) p=3, q=2

(d) Bicubic

Figure 1.17: Examples of local Knot Value Vectors for odd/even degrees. Green horizontal and vertical lines illustrate the Knot value vectors of each Anchor.

1.6.4 Linear Independence-Analysis suitable T-mesh

It is mathematically proven that T-meshes in general do not produce linear independent T-Spline basis functions. Yet under specific constrains a non-suitable T-mesh can be converted to define an independent base of basis functions.

As [170] defines, analysis suitable is a T-mesh whose extended mesh is analysis suitable. The extended T-mesh is defined by extending T-junctions in both directions. For instance, if a T-junction is created by removing the horizontal left line of a cross junction, then the main extension appears on the vanished side until the first line segment or junction is encountered. The other extension has direction opposite to the previous one.

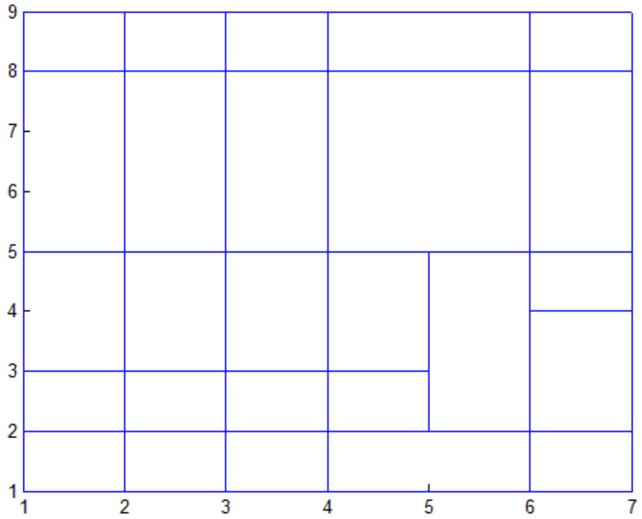
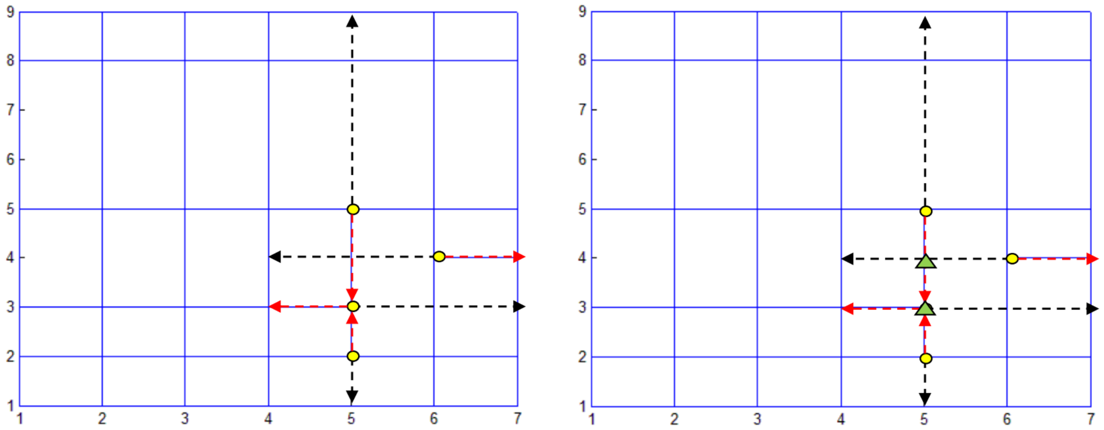


Figure 1.18: Analysis non-suitable T-mesh.

Figure 1.18 provides an example of a typical. Nevertheless, this T-mesh cannot be considered analysis suitable. Figure 1.19a illustrates the extended T-mesh of the previous analysis non-suitable topology. Yellow circles define the T-junctions that need to extended in order to verify the suitability of the mesh. Black dashed lines define the primary extension of the T-junction, while red dashed lines the secondary direction. It is apparent that numerous extension of the T-junctions intersect with each other. These intersections are depicted in Figure 1.19 as green triangles. If such intersections exist, then the T-mesh is considered non analysis suitable. Fortunately, with a simple refinement of the T-mesh it is plausible to convert the T-mesh to an analysis suitable one which creates linearly independent basis functions.



(a) Extended T-mesh.

(b) Extended T-mesh with intersections.

Figure 1.19: Extended T-mesh configuration utilized to define the analysis suitability of a T-mesh. Yellow circles define the T-junctions that need to be extended, Black and red lines the extension and green triangles the intersection of the extended T-mesh.

An analysis non-suitable T-mesh can be transformed into an analysis suitable one with the aid of the Anchors' local Knot Value vectors. Specifically, iterating through each Anchor, the local Knot Value vectors define a support per direction. The tensor product of the unidirectional supports creates a multi-dimensional domain of influence for every Anchor. The envelope of this domain of influence generates Knot lines that are not present in the real mesh. These non-existing lines of the T-mesh are called continuity reduction lines and serve as a partition of the domain into elements that are C^{∞} continuous. On the verge of these lines, the continuity is limited due to the connection different polynomials. The resulting mesh which is produced as a combination of the initial T-mesh with the addition of the continuity reduction lines produces an analysis suitable configuration.

1.6.5 T-Spline basis functions

Given a T-mesh and degrees p, q for both parametric axes ξ and η, $\Omega \subseteq \mathbb{R}^2$ is the index domain that encloses every index a, such that s_a is an Anchor. With the aid of the Anchor's local Knot Value Vectors $\Xi_a = \{\xi_1, \xi_2, ..., \xi_{p+1}, \xi_{p+2}\}, H_a = \{\eta_1, \eta_2, ..., \eta_{p+1}, \eta_{p+2}\}$, the univariate functions for each axis is defined recursively as follows:

$$N_{i,0}(\xi) = \begin{cases} 0, & if\ \xi_i \leq \xi < \xi_{i+1} \\ 1, & otherwise \end{cases} \tag{1.52}$$

based on the formerly evaluated constant functions, piecewise functions of greater degree are evaluated with the assumption that $\frac{0}{0} \doteq 0$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \cdot N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p} - \xi_i} \cdot N_{i+1,p-1}(\xi) \tag{1.53}$$

1.6.6 T-Spline basis function properties

According to [170], T-Spline basis functions posses the following properties:

1. Local Support

$$N_i^p(\xi) = 0 \qquad \forall \xi \notin [\xi_i, \xi_{i+p+1}] \tag{1.54}$$

2. Non-negativity

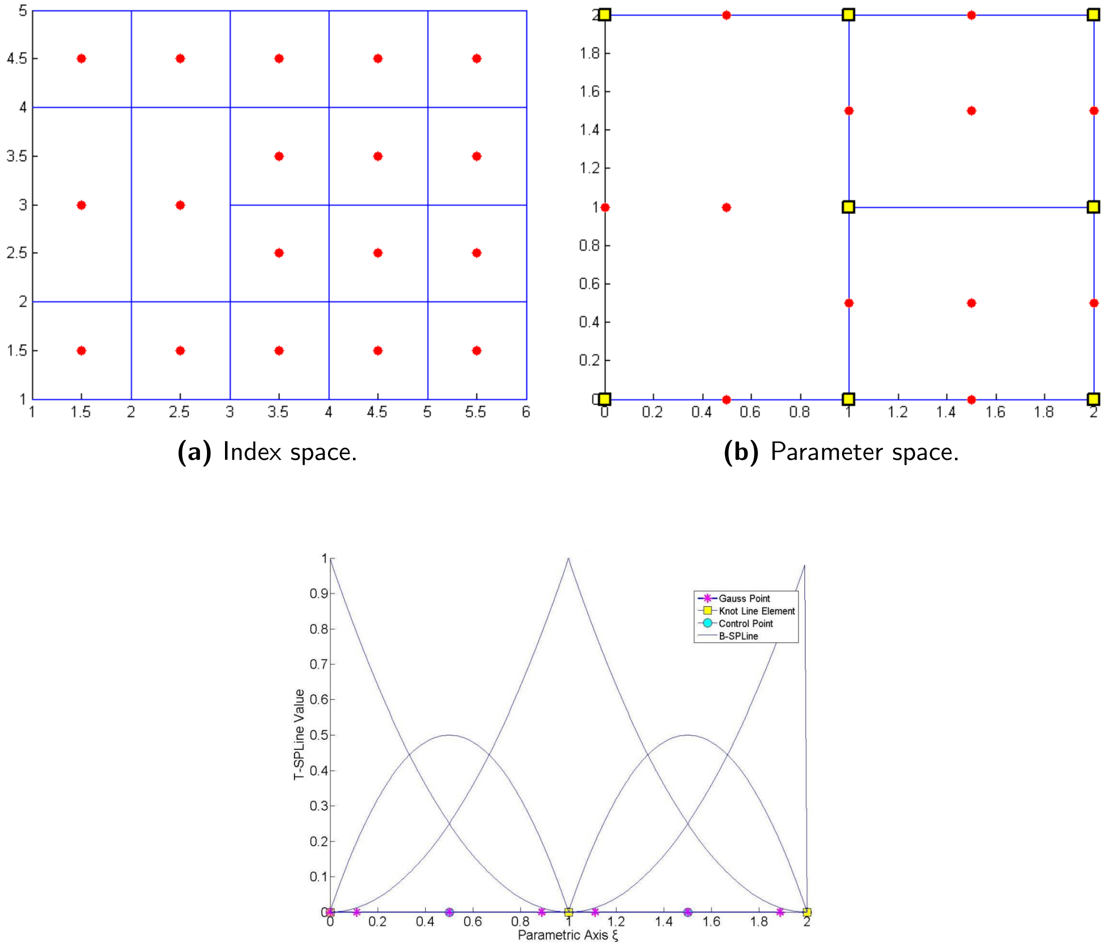
$$N_i^p(\xi) \geq 0 \qquad \forall \xi, i, p \tag{1.55}$$

3. Partition of unity

$$\sum_{i=1}^{n} N_i^p(\xi) = 1 \qquad \forall \xi, p \tag{1.56}$$

4. C^{p-m} continuity across Knots with multiplicity m.

   The property is retained from the extension of NURBS to T-Splines. Given a local Knot Value vector of an Anchor, the equivalent T-Spline basis function has C^{p-m} continuity. This means that this specific basis function can produce p-m continuous derivatives. Continuity less than C^0 is not acceptable for internal Knots, meaning that a certain Knot Value can be repeated in the context of a local Knot Value vector at most p times. Note that as continuity decreases, basis function become steeper. In addition, since the tensor product property of NURBS is no longer present abrupt changes of continuity can occur that depend on the layout of the T-mesh in the index space.



(a) Index space.

(b) Parameter space.

(c) Variable continuity shape functions.

Figure 1.20: C^{p-m} continuity knot lines with multiplicity m.

1.6.7 T-Spline shape functions

Given the local Knot Value Vectors of an anchor for all axes, ξ, η, ζ and the equivalent polynomial degrees p,q,r, the univariate T-Spline basis functions $N_i^p(\xi), M_j^q(\eta), L_k^r(\zeta)$ are generated. Using these functions, their rational T-Spline shape function counterparts can be obtained by

$$T_i^p(\xi) = \frac{N_i^p(\xi) \cdot w_i}{W(\xi)} = \frac{N_i^p(\xi) \cdot w_i}{\sum_{i'=1}^n N_i'^p(\xi) \cdot w_i} \tag{1.57}$$

Inductively for 2D and 3D cases:

$$T_{i,j}^{p,q}(\xi,\eta) = \frac{N_i^p(\xi) \cdot M_j^q(\eta) \cdot w_{ij}}{W(\xi,\eta)} = \frac{N_i^p(\xi) \cdot M_j^q(\eta) \cdot w_{ij}}{\sum_{i'=1}^n \sum_{j'=1}^m N_i'^p(\xi) \cdot M_j'^q(\eta) \cdot w_{ij}} \tag{1.58}$$

$$\begin{aligned} T_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) &= \frac{N_i^p(\xi) \cdot M_j^q(\eta) \cdot L_k^r(\zeta) \cdot w_{ijk}}{W(\xi,\eta,\zeta)} \\ &= \frac{N_i^p(\xi) \cdot M_j^q(\eta) \cdot L_k^r(\zeta) \cdot w_{ijk}}{\sum_{i'=1}^n \sum_{j'=1}^m \sum_{k'=1}^l N_i'^p(\xi) \cdot M_j'^q(\eta) \cdot L_k'^r(\zeta) \cdot w_{ijk}} \end{aligned} \tag{1.59}$$

1.6.8 T-Spline geometries

T-Spline surfaces

$$S(\xi,\eta) = \sum_{i=1}^n \sum_{j=1}^m T_{i,j}^{p,q}(\xi,\eta) \underset{(1x3)}{\{P_{i,j}\}} = \sum_{i=1}^n \sum_{j=1}^m N_i^p(\xi) \cdot M_j^q(\eta) \underset{(1x3)}{\{P_{i,j}\}} \tag{1.60}$$

T-Spline solids

$$\begin{aligned} V(\xi,\eta,\zeta) &= \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l T_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) \underset{(1x3)}{\{P_{i,j,k}\}} \\ &= \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_i^p(\xi) \cdot M_j^q(\eta) \cdot L_k^r(\zeta) \underset{(1x3)}{\{P_{i,j,k}\}} \end{aligned} \tag{1.61}$$

1.6.9 Bézier extraction

As already mentioned in Section 1.4, there is an immediate connection between B-Splines and their predecessor CAD technology, Bézier curves. Specifically, a B-Spline curve can be split into multiple Bézier sections, interconnected with C^2 continuity. In a similar fashion, this subsection discusses the decomposition of NURBS and T-Splines into simple Bézier entities that will be utilized in the following chapters.

NURBS Bézier extraction

The Bézier extraction process described analytically in [34] is summarized here. The main purpose of the Bézier extraction is to enable the representation of a highly continuous advanced Splines into multiple piecewise C^0 Bézier pieces. This decomposition in case of NURBS shape functions is achieved with h-refinement and specifically by repeating all internal Knot Values until their multiplicity is raised to p, p being the polynomial degree.

The Knot insertion procedure of Isogeometric h-refinement work as follows. Given a Knot Value vector $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$ and ξ_k a Knot Value to be inserted in one of the internal spans of the Knot Value Vector, *n+1* new Control Points have to be computed.

$$\bar{P}_A = \begin{cases} P_1, & A = 1, \\ a_A P_A + (1 - a_A) P_{A-1}, & 1 < A < m, \\ P_n, & A = m \end{cases} \tag{1.62}$$

where

$$a_A = \begin{cases} 1, & 1 \leq A \leq k - p, \\ \frac{\xi_k - \xi_A}{\xi_{A+p} - \xi_A}, & k - p + 1 \leq A \leq k, \\ 0, & A \geq k + 1 \end{cases} \tag{1.63}$$

Calculation of the new Control Points with the aid of eqs. 1.62, 1.63 preserves both the continuity of the curve as well as its shape, while the continuity of the shape functions is reduced by one for every single Knot insertion. For example given a Knot Value vector $\Xi = \{0, 0, 0, 1, 2, 3, 3, 3\}$, by raising the multiplicity of the internal Knot Values to p this will be transformed to $\Xi' = \{0, 0, 0, 1, 1, 2, 2, 3, 3, 3\}$. The process described in the later equations for a single Knot insertion can be generalized for multiple simultaneous insertions and eq. 1.62 is transformed to

$$\bar{P}^{j+1} = (C^j)^T \bar{P}^j \tag{1.64}$$

where

$$C^j = \begin{bmatrix} a_1 & 1-a_2 & 0 & \dots & & & 0 \\ 0 & a_2 & 1-a_3 & 0 & \dots & & 0 \\ 0 & 0 & a_3 & 1-a_4 & 0 & \dots & 0 \\ \vdots & & & & & & \\ 0 & \dots & & & 0 & a_{(n+j-1)} & 1-a_{(n+j)} \end{bmatrix} \tag{1.65}$$

[34] proves that eq. 1.65 is the Bézier extraction operator that maps B-Splines and NURBS shape functions to the equivalent Bézier polynomials.

T-Splines Bézier extraction

The process described in the previous subsection can generalized for T-Spline shape functions. The main difference of the procedure according to [167] is that a global Bézier extraction procedure is no longer present. This occurs due to abolishment of the tensor product structure of NURBS. As a result, the extraction operator must be computed in a function-by-function basis. The extraction operator of a single Bézier element is generated row-by-row from the shape functions that affect it. In addition, since global Knot Value Vectors no longer exist, the extraction operator is based on each of the univariate local Knot Value vectors calculated for each Anchor.

A significant difference with NURBS is the introduction of the extended Knot Value vector. The extended Knot Value vector transforms an arbitrary local Knot Value vector into an open one. This is achieved by repeating the initial and final values until the multiplicity is *p+1*. Then the extraction operation is performed at the extended local Knot Value vector. Finally, in case of multivariate Bézier extractions, the operator C of a T-Spline element is produced in a tensor product fashion based on the univariate operators produced for each one of the parametric directions.

In both cases of NURBS and T-Spline shape functions, after the computation of the equivalent extraction operator the transformation between advanced shape functions and Bézier is performed by

$$N(\xi) = C \cdot B(\xi) \tag{1.66}$$

For the bivariate case, the extraction operator C_A is computed by

$$C_A = C_\xi \otimes C_\eta \tag{1.67}$$

and for the trivariate case

$$C_A = C_\xi \otimes C_\eta \otimes C_\zeta \tag{1.68}$$

where $\otimes$ is the Kronecker product of the univariate matrices.

2
Isogeometric Analysis

2.1 Introduction

Parallel to the development of CAD technologies, the scientific area of CAE has rapidly evolved. In the early stage of their development, manual blueprints were shared between designers and computational scientists, who had imminent interactions towards the optimization of the product's design and mechanical performance. Since the dawn of personal computers, ever more complex geometries had to be reproduced, while engineers were assigned with the burden of translating these geometries into analysis-suitable input for the intricate computational mechanics software. This lead to a deviation of the development paths of CAD and CAE, due to the different requirements for either efficient representation of arbitrary geometries or highly accurate analyses of engineering structures. This separate development generated a mismatch between the actual and the analysis geometry, as the tools developed by popular computational mechanics processes, such as the FEM, can only approximate the structure's geometry.

This inherent approximation often leads to erroneous results in geometry sensitive analyses. For example, shell buckling or contact mechanics applications, cannot be accurately reproduced without precise geometry descriptions. In addition, adaptive refinement techniques cannot be efficiently implemented, since they require a direct connection of the approximate design with the exact geometrical representation, which is nonexistent. Similarly, the shape optimization of structures faces the bottleneck of to and from CAD mapping and re-meshing procedures. All these deficiencies of the mappings between CAD and CAE technologies, have raised the need for a single pipeline that treats geometrical and analysis models as a single entity. IGA was introduced by Hughes as design-through-analysis procedure that unifies the until then diverging CAD and CAE industries.

IGA shares many similarities with its predecessor, the FEM method. Specifically, according to [5], both are isoparametric implementations of the Galerkin method and share similar code flow, while many properties such as the partition of unity and bandwidth of the resulting matrices is preserved. On the contrary, several aspect that the average FEA practitioner is accustomed to are now invalidated. Specifically, the exact geometry is utilized in IGA, fact that significantly aids the analysis procedure. Unfortunately, Control Points that describe the geometry do not interpolate it, which in contrast to FEM nodes, does not provide a direct interpretation of the results, but only when combined with the shape functions. The purpose of this chapter is to clarify these differences, between IGA and FEM methods and introduce the basic quantities and processes needed for the analysis of boundary value problems with the aid of IGA.

2.2 Continuum mechanics

All applications of this thesis are based on the PDE of linear elasticity, thus this is the basic subject that will be analyzed in this chapter. All model quantities such as the elements and the integration rules presented can be readily applied in any other PDEs taking into account the proper modification for the assembly of the resulting matrices. The derivation of the linear systems of linear elasticity PDE are summarized here. The detailed procedure can be found in [4].

Starting with the strong form of the Linear Elasticity PDE, where Ω is the boundary value problem domain, the Dirichlet and Neumann boundary conditions imposed on the Γ_{D_i} and Γ_{N_i} parts of the domain equivalently, are presented in eq.2.1 b,c.

$$\sigma_{ij,j} + f_i = 0 \quad in\ \Omega \tag{2.1a}$$

$$u_i = g_i \quad on\ \Gamma_{D_i} \tag{2.1b}$$

$$\sigma_{ij} n_j = h_i \quad on\ \Gamma_{N_i} \tag{2.1c}$$

To retrieve the weak form of the PDE, eq.2.1a is multiplied by weighting function and integrated by parts.

$$\int_\Omega w(i,j)\sigma_{ij} d\Omega = \int_\Omega w_i f_i + \sum_{i=1}^{d}\left(\int_{\Gamma_{N_i}} w_i h_i d\Gamma\right) \tag{2.2}$$

Note that in both eqs. 2.1 and 2.2 the term σ_{ij} represents the stress tensor given by the Hooke's law

$$\sigma_{ij} = c_{ijkl}\epsilon_{kl} \tag{2.3}$$

where ϵ_{ij} is the strain vector, calculated as a gradient of the displacement u and c_{ijkl} is the equivalent constitutive tensor. Eq. 2.2 can be alternatively written as

$$\alpha(\mathbf{w},\mathbf{u}) = L(\mathbf{w}) \tag{2.4a}$$

$$\alpha(\mathbf{w},\mathbf{u}) = \int_\Omega w(i,j)\sigma_{ij} d\Omega \tag{2.4b}$$

$$L(\mathbf{w}) = \int_\Omega w_i f_i + \sum_{i=1}^{d}\left(\int_{\Gamma_{N_i}} w_i h_i d\Gamma\right) \tag{2.4c}$$

The first term of 2.4a provides the matrix of the linear system equation and the second term represents the external force vector.

$$K = \alpha(\mathbf{w},\mathbf{u}) \tag{2.5a}$$

$$F = L(\mathbf{w}) \tag{2.5b}$$

In matrix notation, eq.2.4a can be written as

$$K \cdot u = F \tag{2.6}$$

where the term u represents the solution vector of the resulting linear system and ensures the compatibility between w, u terms of eq. 2.4.

In the discretized form, terms u, w are approximated by

$$u = \sum N_A d_{iA} \cdot \mathbf{e_i} \tag{2.7a}$$

$$w = \sum N_B c_{iA} \cdot \mathbf{e_i} \tag{2.7b}$$

The term $\mathbf{e_i}$ represents the unit basis vectors of $\mathbb{R}^3$, that serve to break down the displacement vector to it's Cartesian axes components. The assembly of the stiffness matrix in the linear elasticity case can be performed in a similar manner to the FEM method, by iterating through the elements of the discretized domain. The following subsections will examine the exact procedure utilized for the derivation of the elements in IGA, considering both NURBS and T-Spline discretizations.

2.2.1 NURBS based isogeometric analysis

For the purposes of this thesis, two types of shape functions are examined, the first of which is NURBS. In order to integrate the stiffness matrix of the linear elasticity case, described in eq.2.5, the integration entities have to be defined. As defined in Chapter 1, the basic entity of a NURBS based geometry is the Patch. Each Patch has similar attributes to the parent element of the FEM method. Sepcifically, the whole geometric entity of a single Patch is mapped to only one parametric domain, utilizing the same isoparametric concept of FEM. On the contrary, even simpler entities, called isogeometric elements, can be considered as integration domains, based on the tensor product structure of NURBS. Both Patch and element structures can be used for integration, yet for the purposes of this manuscript the latter case is utilized, as it resembles the decomposition of a structure into independent entities of FEM.

Elements

Taking into account the definition of a *Knot Vector* given in Chapter 1, the elements in the univariate case are defined as the non-zero spans, formed by its consecutive values.

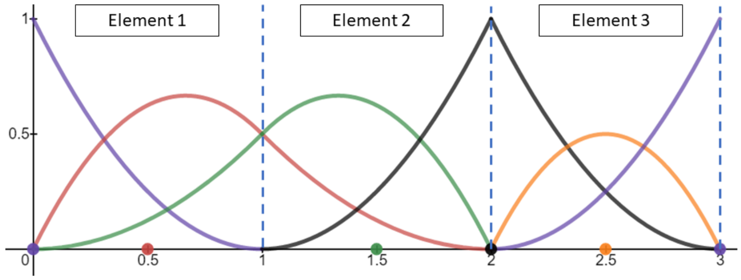


Figure 2.1: Elemental decomposition of an one-dimensional NURBS parameter space with Knot Value Vector $\Xi = \{0, 0, 0, 1, 2, 2, 3, 3, 3\}$.

Figure 2.1, provides an example of the decomposition of an one-dimensional parametric NURBS domain into elements. Given the parametric domain defined by the Knot Value Vector $\Xi = \{0, 0, 0, 1, 2, 2, 3, 3, 3\}$, the equivalent Knot Vector can be extracted which corresponds to $\Xi' = \{0, 1, 2, 3\}$. As apparent, the four consecutive values of the Knot Vector, generate three non-trivial spans. These spans are the definition of one-dimensional elements in Isogeometric Analysis. In case of higher dimensional entities, the tensor product structure of NURBS is utilized.

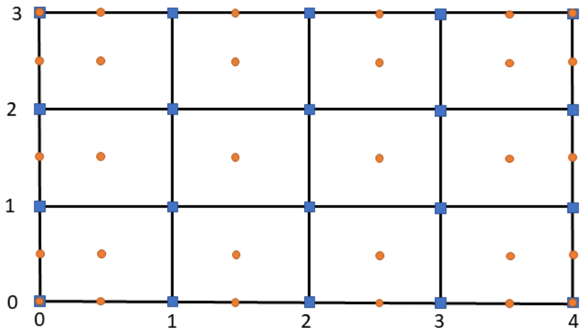


Figure 2.2: Elemental decomposition of an one-dimensional NURBS parameter space with Knot Value Vectors $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 4\}$ and $H = \{0, 0, 0, 1, 2, 3, 3, 3\}$.

An example of the two-dimensional elemental structure of a NURBS plane is given in Figure 2.2. Given the Knot Value Vectors $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 4\}$ and $H = \{0, 0, 0, 1, 2, 3, 3, 3\}$, the equivalent Knot Vectors $\Xi' = \{0, 1, 2, 3, 4\}$ and $H' = \{0, 1, 2, 3\}$ are computed. Each of the rectangular tiles appearing in the illustration of Figure 2.2

are created by a tensor product combination of one-dimensional knot spans. As expected from 4 spans on axis ξ and 3 one axis η, a total of 12 isogeometric NURBS elements are created. Blue rectangles represent the two dimensional Knots of the parametric domain, while orange circles depict the parametric positions of the Control Points.

In a similar fashion, three-dimensional elements can be derived for the analysis of trivariate Boundary Value Problem (BVP)s.

Mesh Refinement

A vital process of computational methods, that is significantly simplified with the development of IGA, is mesh refinement. Mesh refinement's scope is to enhance the minimum viable mesh required for the design of a structure, in order to provide a fine enough discretization, that can efficiently and accurately analyze the model under consideration. IGA extends the already existing processes of h-refinement and p-refinement of FEM and renames them as *Knot Value Insertion* and *Degree Elevation* equivalently, while simultaneously introducing k-refinement which is a combination of the aforementioned methods.

Knot Value Insertion refinement procedure, creates a finer mesh by adding new Knot Values in the Knot Value Vector. Boundary Knot Values cannot be inserted at the edges of the domain, as continuity is already at a minimum C^{-1} and thus must remain unaltered. By adding internal Knot Values, either more integration elements are generated, or the multitude of an existing Knot Value is increased. The total multiplicity of internal Knot Values cannot exceed the polynomial degree p, thus resulting to a minimum C^0 continuity. Due to the augmentation of the initial Knot Value Vector Ξ with new values and the requirement that the curve remains unaltered the following formula holds

$$C(\Xi) = \sum_{i=1}^{n} \{N_i^p(\xi) \cdot X_i\} = \sum_{j=1}^{m} \{\overline{N}_j^p(\xi) \cdot \overline{X}_j\} \tag{2.8}$$

where overline in eq.2.8 denotes the new configuration. Note that the number of Control Points changes between the initial and the final configuration. The multitude of new Control Points equals the number of Knot Values inserted. A transformation between the two Control Points networks can be obtained via

$$\underset{(mx1)}{\{\overline{X}\}} = \underset{(mxn)}{[T^p]} \cdot \underset{(nx1)}{\{X\}} \tag{2.9}$$

where the transformation matrix $[T^p]$ is formed recursively as

$$T^0_{i,j} = \begin{cases} 1, & \overline{\xi}_i \in [\xi_j, \xi_{j+1}] \\ 0, & otherwise \end{cases} \tag{2.10a}$$

$$T^q_{i,j} = \frac{\overline{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi} \cdot T^{q-1}_{i,j} + \frac{\xi_{j+q+1} - \overline{\xi}_{i+q}}{\xi - j + q + 1 - \xi_{j+1}} \cdot T^{q-1}_{i,j+1}, \quad q = 1, 2, ..., p \tag{2.10b}$$

The second mesh refinement technique of isogeometric analysis, Degree Elevation, raises the polynomial degree of the shape functions to $\overline{p}$ from initial p. Since the curve with all of its properties must remain unaltered, the multiplicity of all Knot Values is increased by $\overline{p} - p$ in order to retain initial continuity of the geometry at the Knot positions. The new Control Points of the curve will be evaluated with the aid of the transformation matrix defined by eq. 2.10. The resulting Control Points will be utilized for the evaluation of the new curve functions $\underset{(mxm)}{[\overline{N}]}$ thus

$$\underset{(mx1)}{\{\overline{X}\}} = \left(\underset{(mxm)}{[\overline{N}]}^{T}\right)^{-1} \cdot \underset{(mxn)}{[N]^T} \cdot \{X\} \tag{2.11}$$

where the transformation matrix equals

$$\underset{(mxn)}{[T]} = \left(\underset{(mxm)}{[\overline{N}]}^{T}\right)^{-1} \cdot \underset{(mxn)}{[N]^T} \tag{2.12}$$

In case the term $\left(\underset{(mxm)}{[\overline{N}]}^{T}\right)^{-1}$ is not reversible, a new set of Control Points must be evaluated.

The final refinement method introduced with IGA is k-refinement. The purpose of k-refinement is to increase the continuity at the introduced Knot Values, as the latter h- and p-refinement methods, either reduce or retain the existing continuity. To achieve this, both p- and h-refinements are performed consecutively. Initially, p-refinement increases the polynomials degree of the shape functions, while retaining the existing continuity at the Knots. In a second step, h-refinement introduces new Knots with continuity raised to $C^{\overline{p}-1}$ where $\overline{p}$ is the new augmented polynomial degree.

Note that all former refinement techniques are applicable at B-Spline entities. In case of NURBS, the same procedures cannot be directly applied to the homogeneous coordinates $\{X_i, Y_i, Z_i, W_i\}$, rather to the projective Control Point coordinates that incorporate the weight information to the rest of the Cartesian position data.

Stiffness Matrix - Integration

As mentioned in Section 2.2, the entities over which integration is performed are either Patches that represent the entirety of the parametric domain, or elements whose extraction was previously described. For the purposes of this thesis, the latter option is employed, to maintain the analogy of FEM and IGA computational methods.

The matrices produced by eq. 2.5a are integrated over the isogeometric elements, taking into account the Gauss quadrature. Recall that elements based on the definition given in the latter section, are domains where shape functions can be represented by piecewise polynomials. It has been proven that for multivariate elements, *p+1* Gauss Points per axis are required, where p is the polynomial degree of the shape functions.

The integration points coordinates for each axes are obtained on the parent element $[-1, 1]$, as the roots of the Legendre polynomials in this domain. The next step is to transform their coordinates ξ^R and weights w_ξ^R from the reference knot span $[-1, 1]$ to the desired knot span $[\xi_i, \xi_{i+1}]$.

$$\xi = \frac{(\xi_{i+1} - \xi_i) \cdot \xi^R + (\xi_{i+1+\xi_i})}{2} \tag{2.13a}$$

$$w_{GP}^R = \frac{(\xi_{i+1} - \xi_i)}{2} \cdot w_\xi^R \tag{2.13b}$$

The same formula holds for the evaluation of Gauss Points and their equivalent weights for all parametric directions. The multivariate Gauss Points are derived by applying the tensor product rule to univariate Gauss Points for each of the multiple parametric direction.

n	ξ_i	w_i
1	0	2
2	-0.57735	1
	0.57735	1
3	-0.77459	0.55555
	0	0.88888
	0.77459	0.55555
4	-0.86113	0.34785
	-0.33998	0.65214
	0.33998	0.65214
	0.86113	0.34785
5	-0.90617	0.23692
	-0.33998	0.47862
	0	0.56888
	0.33998	0.47862
	0.90617	0.23692

Table 2.1: Univariate Gauss point coordinates and weights in the parent domain $[-1, 1]$.

One-dimensional stiffness matrix

In the single dimension case, the strain of each point is calculated as a gradient of the deformation by

$$\underset{(1x1)}{\{\epsilon\}} = \epsilon_x = \frac{\partial u}{\partial x} \tag{2.14}$$

Note that gradient is calculated with respect to position of a point in the physical space. Yet, the Gauss Points are located at the parameter space and thus a transformation between the two spaces must be defined. This transformation is the Jacobian one and remains the same with the FEM equivalent as both are based on the same isoparametric concept. Specifically the following relationship holds,

$$\left[\frac{\partial \phi}{\partial \xi}\right] = [J] \cdot \left[\frac{\partial \phi}{\partial x}\right] \tag{2.15}$$

where $[J]$ is the Jacobian matrix. In the discretized domain, the Jacobian matrix is evaluated with the aid of the shape functions as follows:

$$[J(\xi)] = \underset{(1xn)}{[R_{1,\xi}(\xi)\ R_{2,\xi}(\xi)\ ...R_{n,\xi}(\xi)]} \cdot \underset{(1xn)}{[X_1\ X_2\ ...X_n]}^T \tag{2.16}$$

where $R_{i,\xi}$ denotes the first derivative of the shape functions per axis ξ. In both FEM and IGA the inverse transformation of eq. 2.15 is performed and thus the matrix $[J]^{-1}$ is needed. As a result, special care must be taken for the Jacobian matrix to be reversible by eliminating points of singularity and ensuring that the directions of parameter and physical space coincide.

The next step is to calculate the deformation matrix $[B]$, which is utilized to compute the strains at any arbitrary point of the domain based on the nodal quantities of the discretization. This matrix is split into two parts, $[B_1]$ and $[B_2]$. Matrix $[B_1]$ transfers the strains of the element from parameter to physical space and matrix $[B_2]$ transfers the nodal displacements of the elements to the strains at the parameter space. In case of one-dimensional problems they are calculated by

$$\underset{(1x1)}{[B_1(\xi)]} = \left[\frac{1}{J_{11}}\right] \tag{2.17a}$$

$$\underset{(1xn)}{[B_2(\xi)]} = [R_{1,\xi}(\xi)\ R_{2,\xi}(\xi)\ ...R_{n,\xi}(\xi)] \tag{2.17b}$$

Matrix $[B]$ is calculated as a product of the submatrices $[B_1]$, $[B_2]$ as

$$\underset{(1xn)}{[B(\xi)]} = \underset{(1x1)}{[B_1(\xi)]} \cdot \underset{(1xn)}{[B_2(\xi)]} \tag{2.18}$$

As a result, the stiffness matrix for one-dimensional isogeometric elements is evaluated as

$$\underset{(nxn)}{[K]} = \int_{\xi_0}^{\xi_{n+p+1}} \underset{(nx1)}{[B(\xi)]^T} \cdot E \cdot \underset{(1xn)}{[B(\xi)]^T} \cdot A \cdot det[J] d\xi \tag{2.19}$$

where terms ξ_0 and ξ_{n+p+1} denote the boundaries of the parametric domain and A the section's area of the one-dimensional model and E the Young's modulus of elasticity. The integral presented in eq. 2.19 is never calculated analytically. Instead a numerical integration is performed, taking into account the Gauss Points evaluated at each separate parametric integral. This transforms eq. 2.19 to:

$$\underset{(nxn)}{[K]} = \sum_{i=1}^{nGP} \left\{ \underset{(nx1)}{[B(\xi_i)]^T} \cdot E \cdot \underset{(1xn)}{[B(\xi_i)]^T} \cdot A \cdot det[J] \cdot w_i^{GP} d\xi \right\} \tag{2.20}$$

where w_i^{GP} is the equivalent weight of each independent Gauss Point.

Two-dimensional stiffness matrix

The aforementioned procedure for the univariate stiffness matrix can be extended to multidimensional cases. Specifically, in the two-dimensional case, the strain vector at each physical is defined based on the directional displacements u, v as

$$\underset{(3x1)}{\{\epsilon\}} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} \tag{2.21}$$

In a similar fashion to the one-dimensional case, the Jacobian transformation is required to transfer information between the parameter and the physical space.

$$\begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \xi} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \end{bmatrix} \Longrightarrow \begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \end{bmatrix} = \underset{(2x2)}{[J]} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \end{bmatrix} \tag{2.22}$$

or equivalently, the inverse transformation to transition from parameter to physical space

$$\begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \end{bmatrix} = \underset{(2x2)}{[J]}^{-1} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \end{bmatrix} \tag{2.23}$$

Taking advantage of the discretized notation, the Jacobian transformation at each parametric point is calculated as

$$\underset{(2x2)}{[J]} = \begin{bmatrix} R_{1,\xi}(\xi,\eta) & R_{2,\xi}(\xi,\eta) & ... & R_{N,\xi}(\xi,\eta) \\ R_{1,\eta}(\xi,\eta) & R_{2,\eta}(\xi,\eta) & ... & R_{N,\eta}(\xi,\eta) \end{bmatrix} \cdot \begin{bmatrix} X_1 & Y_1 \\ X_2 & Y_2 \\ ... & ... \\ X_n & Y_N \end{bmatrix} \tag{2.24}$$

The deformation matrix $[B]$ in the two-dimensional case is also computed as a product of the submatrices $[B_1]$ and $[B_2]$, where $[B_1]$

$$\underset{(3x4)}{[B_1(\xi,\eta)]} = \frac{1}{det[J]} \cdot \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix} \tag{2.25}$$

and $[B_2]$:

$$\underset{(4x2N)}{[B_2(\xi,\eta)]} = \frac{1}{det[J]} \cdot \begin{bmatrix} R_{1,\xi} & 0 & R_{2,\xi} & 0 & ... & R_{N,\xi} & 0 \\ R_{1,\eta} & 0 & R_{2,\eta} & 0 & ... & R_{N,\eta} & 0 \\ 0 & R_{1,\xi} & 0 & R_{2,\xi} & 0 & ... & R_{N,\xi} \\ 0 & R_{1,\eta} & 0 & R_{2,\eta} & 0 & ... & R_{N,\eta} \end{bmatrix} \tag{2.26}$$

In order to evaluate the bivariate stiffness matrix, integration is required

$$\underset{(2Nx2N)}{[K]} = \int_{\xi_0}^{\xi_{n+p+1}} \int_{\eta_0}^{\eta_{m+q+1}} \underset{(2Nx3)}{[B(\xi,\eta)]^T} \cdot \underset{(3x3)}{[E]} \cdot \underset{(3x2N)}{[B(\xi,\eta)]} \cdot t \cdot det[J] d\eta d\xi \tag{2.27}$$

By applying numerical quadrature, eq. 2.27 is transformed to:

$$\underset{(2Nx2N)}{[K]} = \sum_{i=1}^{GP_\xi} \sum_{j=1}^{GP_\eta} \underset{(2Nx3)}{[B(\xi_i,\eta_j)]^T} \cdot \underset{(3x3)}{[E]} \cdot \underset{(3x2N)}{[B(\xi_i,\eta_j)]} \cdot t \cdot det[J] \cdot w_i^{GP_\xi} \cdot w_i^{GP_\xi} \tag{2.28}$$

where t is the thickness of the bivariate domain at each integration point.

Three-dimensional stiffness matrix

A similar process is followed in the trivariate continuum mechanics case, where the strain field is calculated as the directional gradient of the displacements u, v, w

$$\underset{(6x1)}{\{\epsilon\}} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{bmatrix} \tag{2.29}$$

Once more, for the definition of the Jacobian matrix we have

$$\begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \\ \frac{\partial \phi}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{bmatrix} \Longrightarrow \begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \\ \frac{\partial \phi}{\partial \zeta} \end{bmatrix} = \underset{(3x3)}{[J]} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{bmatrix} \tag{2.30}$$

and the inverse Jacobian as well:

$$\begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{bmatrix} = \underset{(3x3)}{[J]}^{\,-1} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \\ \frac{\partial \phi}{\partial \zeta} \end{bmatrix} \tag{2.31}$$

For auxiliary purposes, the terms of the computed inverse Jacobian matrix are named as follows

$$\underset{(3x3)}{[J]}^{\,-1} \begin{bmatrix} J^*_{11} & J^*_{12} & J^*_{13} \\ J^*_{21} & J^*_{22} & J^*_{23} \\ J^*_{31} & J^*_{32} & J^*_{33} \end{bmatrix} \tag{2.32}$$

in order to faciliate an immediate formula representation of the deformation matrix part $[B_1]$.

$$\underset{(6x9)}{[B_1(\xi,\eta,\zeta)]} \begin{bmatrix} J^*_{11} & J^*_{12} & J^*_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & J^*_{21} & J^*_{22} & J^*_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & J^*_{31} & J^*_{32} & J^*_{33} \\ J^*_{21} & J^*_{22} & J^*_{23} & J^*_{11} & J^*_{12} & J^*_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & J^*_{31} & J^*_{32} & J^*_{33} & J^*_{21} & J^*_{22} & J^*_{23} \\ J^*_{31} & J^*_{32} & J^*_{33} & 0 & 0 & 0 & J^*_{11} & J^*_{12} & J^*_{13} \end{bmatrix} \tag{2.33}$$

Equivalently based on the shape function derivatives, the second subpart of the deformation matrix $[B_2]$ is calculated as follows:

$$\underset{(9x3N)}{[B_2(\xi,\eta,\zeta)]}\begin{bmatrix} R_{1,\xi} & 0 & 0 & ... & ... & ... & R_{N,\xi} & 0 & 0 \\ R_{1,\eta} & 0 & 0 & ... & ... & ... & R_{N,\eta} & 0 & 0 \\ R_{1,\zeta} & 0 & 0 & ... & ... & ... & R_{N,\zeta} & 0 & 0 \\ 0 & R_{1,\xi} & 0 & ... & ... & ... & 0 & R_{N,\xi} & 0 \\ 0 & R_{1,\eta} & 0 & ... & ... & ... & 0 & R_{N,\eta} & 0 \\ 0 & R_{1,\zeta} & 0 & ... & ... & ... & 0 & R_{N,\zeta} & 0 \\ 0 & 0 & R_{1,\xi} & ... & ... & ... & 0 & 0 & R_{N,\xi} \\ 0 & 0 & R_{1,\eta} & ... & ... & ... & 0 & 0 & R_{N,\eta} \\ 0 & 0 & R_{1,\zeta} & ... & ... & ... & 0 & 0 & R_{N,\zeta} \end{bmatrix} \tag{2.34}$$

and the equivalent deformation matrix for 3D elasticity is derived as a product of the former two matrices:

$$\underset{(6x3N)}{[B(\xi,\eta,\zeta)]} = \underset{(6x9)}{[B_1(\xi,\eta,\zeta)]} \cdot \underset{(9x3N)}{[B(\xi,\eta,\zeta)]} \tag{2.35}$$

The corresponding stiffness matrix is integrated as follows:

$$\underset{(3Nx3N)}{[K]} = \int_{\xi_0}^{\xi_{n+p+1}} \int_{\eta_0}^{\eta_{m+q+1}} \int_{\zeta_0}^{\zeta_{l+r+1}} \underset{(3Nx6)}{[B(\xi,\eta,\zeta)]^T} \cdot \underset{(6x6)}{[E]} \cdot \underset{(6x3N)}{[B(\xi,\eta,\zeta)]} \cdot det[J] d\xi d\eta d\zeta \tag{2.36}$$

Finally, by applying numerical integration we obtain:

$$\underset{(3Nx3N)}{[K]} = \sum_{i=1}^{GP_{xi}} \sum_{j=1}^{GP_\eta} \sum_{k=1}^{GP_\zeta} \underset{(3Nx6)}{[B(\xi_i,\eta_j,\zeta_k)]^T} \cdot \underset{(6x6)}{[E]} \cdot \underset{(6x3N)}{[B(\xi_i,\eta_j,\zeta_k)]} \cdot det[J] \cdot w_i^{GP_\xi} \cdot w_j^{GP_\eta} \cdot w_k^{GP_\zeta} \tag{2.37}$$

2.2.2 T-Spline based isogeometric analysis

As already mentioned in the previous chapter, due the linear independence limitations imposed by the shape functions, the T-mesh differs from the integration mesh in case of T-Spline geometries. In this section the generation of the final integration mesh for T-Spline entities will be discussed.

Elements

The integration mesh is a product of the initial T-mesh configuration and the supports of the Anchors. An example of a T-mesh structure is given in Figure 2.3. Two Anchors, positioned at $(\xi_a, \eta_a) = (5, 2.5)$ and $(\xi_b, \eta_b) = (6.5, 5)$ are depicted with red circles, while their equivalent Knot Value Vectors are shown as green lines.

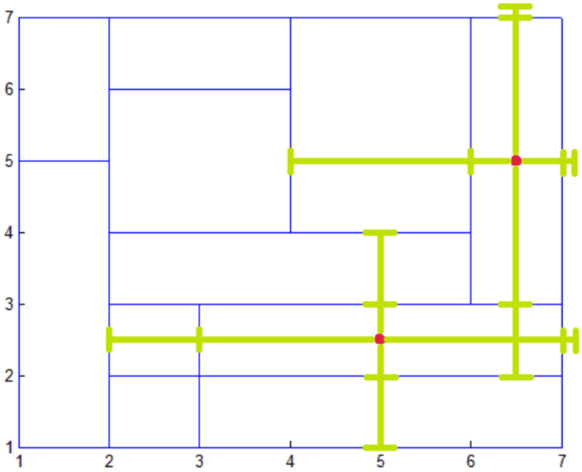


Figure 2.3: T-mesh configuration. Anchors depicted in red circles, with their equivalent Knot Value Vectors depicted with green lines.

The support of each Anchor is defined as the tensor product domain generated based on it's local Knot Value Vectors. An example of this influence domain is shown as a shaded area in Figure 2.4. It is apparent that the defined area creates lines that do not exist in the initial T-mesh layout. An example of such a line is shown in Figure 2.4 as dark red line. By iterating through all Anchors, this procedure returns as a product these additional lines of the whole T-mesh configuration. They are named continuity reduction lines, as they divide the Index space into elements, where shape functions are C^∞ continuous. On the verge of these lines, continuity is limited due to the connection of different piecewise polynomial functions.

These continuity reduction lines form the final integration mesh, on which integration will be performed after the placement of Gauss Points. An example of the final T-mesh configuration containing all continuity reduction lines is illustrated in 2.5.

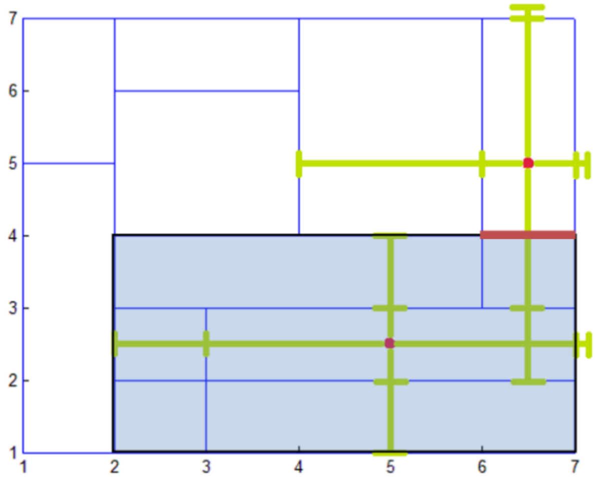


Figure 2.4: T-mesh configuration. Anchor with parametric coordinates (5, 2.5), defines the shaded influence domain. Continuity reduction line is illustrated with bold red line.

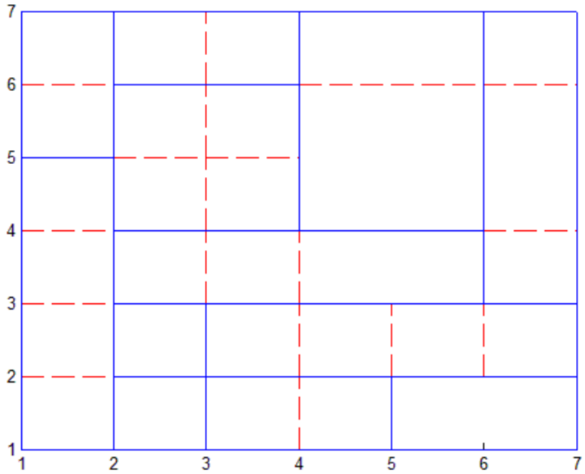


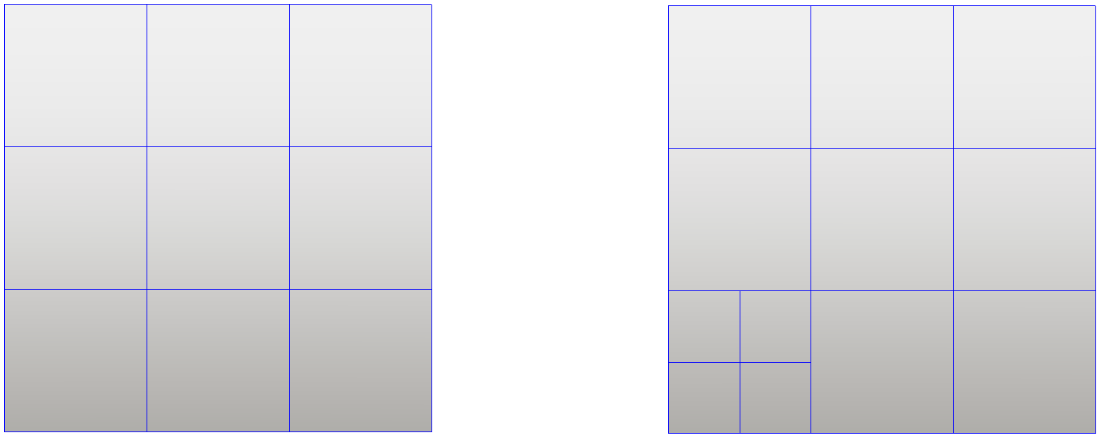
Figure 2.5: T-mesh configuration. Continuity reduction lines generated for the entire mesh are shown as red dashed lines.

Mesh Refinement

T-Splines are the first CAD technology that enables true local refinement of the topology. The insertion of T-junctions enables the refinement to take place on an element level.

Faces subdivision is the most common refinement strategy. The main process is to divide one of the elements into four new ones. In addition, T-Splines allow the insertion of single Control Points in places where geometry needs special local handling.

An example of a face subdivision procedure for the refinement of T-Splines spaces is provided in Figure 2.6. Note that Figure 2.6a resembles a common NURBS mesh. The representation of NURBS tensor product spaces, is allowed within T-Splines technology as it's theory renders it a superset of NURBS. Figure 2.6b, illustrates the subdivision of the bottom left face of the T-Spline index space, into four subfaces. As apparent, T-junctions enable the subdivision of a single geometrical entity, thus enabling true local refinement. The Knot lines inserted in Figure 2.6b, do not propagate throughout the parametric domain as in the case of NURBS, thus minimizing the number of introduced Control Points needed, to attain the same level of local control of the geometry. Detailed processes and algorithms for the refinement of isogeometric T-Spline spaces can be found in [166].



(a) Initial T-mesh configuration.

(b) Refined T-mesh configuration.

Figure 2.6: The initial T-mesh configuration is refined by subdividing a face into four independent ones.*

T-Splines Bézier extraction

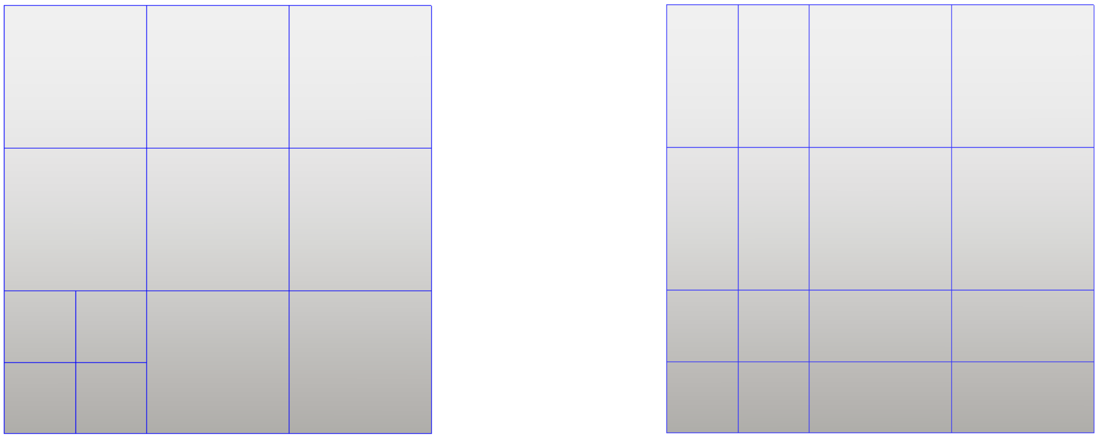
Throughout this thesis, whenever T-Spline geometries were utilized, the process of Bézier extraction is employed, as it offers an immediate integration with the existing in-house

*Created with Rhinoceros

FEM code. A T-Spline plugin for Rhino was utilized [†], that extracted geometric data according to process described in [165].

Figure 2.7b provides an example of the extracted Bézier elements layout, based on the initial Index Space configuration of Figure 2.7a. As described in Chapter 1, the Bézier extraction operator for T-Spline element is calculated in a function-by-function basis, due to the intricate pattern of shape function that affect it. The mapping between elemental T-Spline shape functions and Bézier element shape functions is the following

$$\mathbf{N}^e(\xi) = \mathbf{C}^e\mathbf{B}(\xi) \tag{2.38}$$


(a) T-mesh configuration.

(b) Bézier elements layout.

Figure 2.7: T-mesh configuration and extracted Bézier elements.[‡]

Rational T-Spline shape functions of an element are calculated using the following formula

$$\mathbf{R}^e(\xi) = \frac{\mathbf{W}^e\mathbf{N}^e(\xi)}{(\mathbf{w}^e)^T\mathbf{N}^e(\xi)} \tag{2.39}$$

where $\mathbf{R}^e(\xi)$ is the vector of T-Spline shape functions, $\mathbf{w}^e$ a vector containing the weights of the elemental Control Points and $\mathbf{W}^e$ the diagonal matrix created from $\mathbf{w}^e$. With the aid of eq.2.38, eq.2.39 is transformed to

[†] Autodesk T-Spline plugin for Rhino.

[‡] Created with Rhinoceros

$$\mathbf{R}^e(\xi) = \frac{\mathbf{W}^e\mathbf{C}^e\mathbf{B}(\xi)}{(\mathbf{w}^e)^T\mathbf{N}^e\mathbf{B}(\xi)} \tag{2.40}$$

In order to integrate the Linear elasticity PDE with T-Splines shape functions, the same process of NURBS is repeated. Starting from the strong form of the PDE, multiplying by a set of trial functions and integrating by parts, the weak form of the PDE is obtained.

$$\alpha(\mathbf{w}, \mathbf{u}) = L(\mathbf{w}) \tag{2.41a}$$

$$\alpha(\mathbf{w}, \mathbf{u}) = \int_\Omega w(i,j)\sigma_{ij} d\Omega \tag{2.41b}$$

$$L(\mathbf{w}) = \int_\Omega w_i f_i + \sum_{i=1}^{d} \left(\int_{\Gamma_{N_i}} w_i h_i d\Gamma \right) \tag{2.41c}$$

As analytically described for NURBS, the computation of the stiffness matrix requires a mapping from physical to parametric coordinates and vice versa. This mapping is based on the derivatives of the shape functions. Since only shape functions are computed in a different manner compared to NURBS, only the Bézier extracted T-Spline derivatives will be provided here. For this purpose, eq.2.40 is rewritten as

$$\mathbf{R}^e(\xi) = \mathbf{W}^e\mathbf{C}^e\frac{\mathbf{B}(\xi)}{W^e(\xi)} \tag{2.42}$$

where

$$W^e(\xi) = (\mathbf{w}^e)^T\mathbf{N}^e\mathbf{B}(\xi) \tag{2.43}$$

Thus, the derivatives of T-Spline shape functions $\mathbf{R}^e$ with respect to the local parametric axes ξ, η are calculated as follows:

$$\frac{\partial \mathbf{R}^e(\xi,\eta)}{\partial \xi} = \mathbf{W}^e\mathbf{C}^e\frac{\partial}{\partial \xi}\left(\frac{\mathbf{B}(\xi,\eta)}{W^e(\xi)}\right) = \mathbf{W}^e\mathbf{C}^e\left(\frac{1}{W^e(\xi,\eta)}\frac{\partial \mathbf{B}(\xi,\eta)}{\partial \xi} - \frac{\partial W^e(\xi,\eta)}{\partial \xi}\frac{\mathbf{B}(\xi,\eta)}{W^e(\xi,\eta)^2}\right) \tag{2.44a}$$

$$\frac{\partial \mathbf{R}^e(\xi,\eta)}{\partial \eta} = \mathbf{W}^e\mathbf{C}^e\frac{\partial}{\partial \eta}\left(\frac{\mathbf{B}(\xi,\eta)}{W^e(\xi)}\right) = \mathbf{W}^e\mathbf{C}^e\left(\frac{1}{W^e(\xi,\eta)}\frac{\partial \mathbf{B}(\xi,\eta)}{\partial \eta} - \frac{\partial W^e(\xi,\eta)}{\partial \eta}\frac{\mathbf{B}(\xi,\eta)}{W^e(\xi,\eta)^2}\right) \tag{2.44b}$$

2.3 Structural elements

Despite most FEM and IGA element formulations are produced from the discretization of PDEs, there is a wider category of elements that are generated based on simplifications of the general PDE used for continua. These simplifications are usually derived from the dimensionality of the structure, its special boundary conditions, or its allowed deformation state. For instance, shells structures are considered a special continuum case, since one of the three dimensions of the structure is significantly smaller than the rest, specifically the thickness. As a result, the three-dimensional differential equation that governs this structure can be expressed in terms of only the mid-thickness or alternatively, middle surface of the structure.

Various theories have been developed in order to accurately model the response of shell in the field of computational mechanics, most profound among which are Reissner-Midlin and Kirchhoff-Love shell theories. Reissner-Midlin is mainly utilized for the modelling of thick shells in order to take into account transverse shear deformation, while Kirchhoff-Love theory disregards them. Even though most engineering structures belong to the thin shell part of the theory, continuity limitations imposed by Kirchhoff-Love shells made Reissner-Midlin theory dominant in most commercial and academic FEM codes. The underlying reason is that a minimum C^2 continuity required in case of Kirchhoff-Love shells cannot be obtained in a straightforward manner by FEM, as finite elements are connected with C^0 continuity and thus lead to a requirement for non-local formulations or penalty-like constraints at the nodes to enforce it. IGA with its enhanced continuity shape functions, allows for a straightforward implementation of Kirchhoff-Love shells. This subsection will discuss the basic of linear small deformation Kirchhoff-Love shell that will be utilized later on in this thesis.

2.3.1 Differential geometry of surfaces

A brief introduction to the differential geometry of surfaces will be provided, that serves as a basis for the Kirchhoff-Love shell formulation. Based on the midsurface representation of the shell, each point that lies on the surface can be identified by its position vector $\mathbf{x}_{mid}$.

$$\mathbf{x}_{mid} = x_{mid}\mathbf{e}_1 + y_{mid}\mathbf{e}_2 + z_{mid}\mathbf{e}_3 \tag{2.45}$$

As eq.2.45 shows, the position vector of every midsurface point is generated by linear combination of the Cartesian unit vectors $\mathbf{e}_i$ and position ordinates $\{x, y, z\}$, one for each Cartesian axis. Any point $\mathbf{x}_t$ that lies ζ thickness distance from the midsurface is given by

$$\mathbf{x}_t = \mathbf{x}_{mid} + \zeta\mathbf{a}_3, \qquad -\frac{t}{2} \leq \zeta \leq \frac{t}{2} \tag{2.46}$$

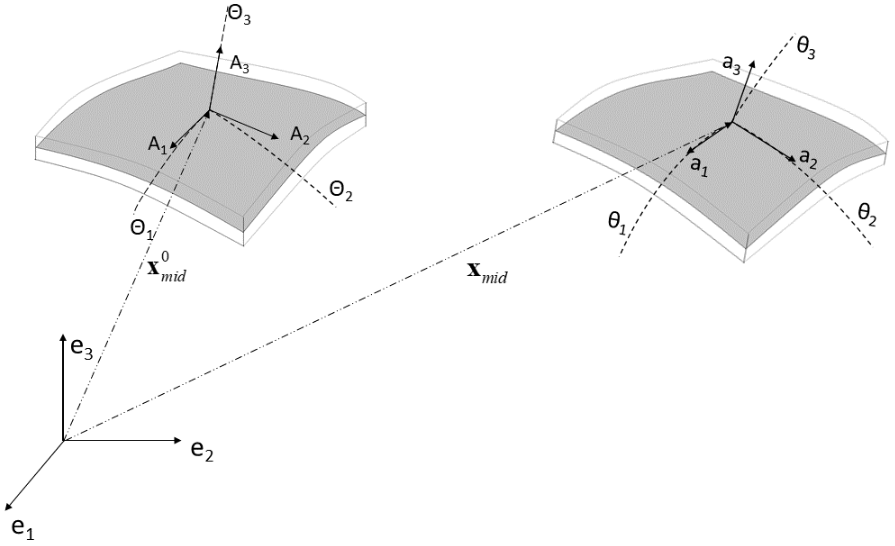


Figure 2.8: Initial and deformed configuration of a shell.

where a_3 is the normal covariant vector. Given the position of a midsurface point $\mathbf{x}$, the tangential covariant base vectors are defined

$$\mathbf{a}_i = \frac{\partial \mathbf{x}}{\partial \theta^i} = \mathbf{x}_{,i} \tag{2.47}$$

In eq.2.47, θ_i denotes the contravariant axes, while the term $(\;), i$ expresses the gradient of the vector per contravariant axis θ_i. Covariant and contravariant vectors at every midsurface point are related with the following expression

$$\mathbf{g}_i \cdot \mathbf{g}^j = \delta_i^j \tag{2.48}$$

where $\delta_i^j$is the Kronecker delta function. Based on the former, the third normal covariant base vector is defined as

$$\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\|\mathbf{a}_1 \times \mathbf{a}_2\|} \tag{2.49}$$

The covariant metric coefficients $g_{\alpha\beta}$ are computed as dot products of the covariant base vectors

$$a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta \tag{2.50}$$

and finally, the curvature tensor coefficients $b_{\alpha\beta}$, that provide the curvature properties of the surface are defined as

$$b_{\alpha\beta} = \mathbf{a}_{\alpha,\beta} \cdot \mathbf{a}_3 \tag{2.51}$$

2.3.2 Kirchhoff-Love shell

The Kirchhoff-Love shell formulation is considered an extension of Euler-Bernoulli beam theory, where the shell kinematics are described by the position of the shell's midsurface. The main assumptions taken into account for the derivation of Kirchhoff-Love theory are that straight lines, normal to the midsurface remain straight and normal to the midsurface in the deformed configuration , while thickness remains unchanged.

The equation that regulates the deformation of any shell midsurface point is obtained as the difference between that position of the point in the current and the reference configuration. Based on the illustration provided in Figure 2.8, this translates to:

$$\mathbf{u} = \mathbf{x}_{mid} - \mathbf{x}_{mid}^0 \tag{2.52}$$

The kinematic variables of the shell are described in terms of the quantities introduced for the differential geometry of the surfaces. Specifically, by utilizing eq. 2.50, 2.51 and neglecting the quadratic terms of the displacement gradients, the strain tensor components are computed as follows:

$$\epsilon_{\alpha\beta} = \frac{1}{2}(a^0_\alpha \cdot u,_\beta + a^0_\beta \cdot u,_\alpha) \tag{2.53}$$

Equivalently, the bending strains are calculated by

$$\kappa_{\alpha\beta} = \frac{1}{2}(a^0_\alpha \cdot \Delta a_3,_\beta + a^0_\beta \cdot \Delta a_3,_\alpha + u,_\alpha \cdot a^0_3,_\beta + u,_\beta \cdot a^0_3,_\alpha) \tag{2.54}$$

where a^0_α denotes the shell tangent vector at the reference configuration, while the normal vector derivatives in the reference configuration are calculated using the reference Jacobian matrix j^0 as

$$a^0_3,_\alpha = (j^0)^{-1}(a^0_1,_\alpha \times a^0_2 + a^0_1 \times a^0_2,_\alpha) \tag{2.55a}$$

$$\Delta a_3,_\alpha = (j^0)^{-1}(u,_{1\alpha} \times a^0_2 + u^0,_1 \times a^0_2,_\alpha + a^0_1,_\alpha \times u,_2 + a^0_1 \times u_2,_\alpha) \tag{2.55b}$$

By combining eqs. 2.55a and 2.55b the total in-plane strain at a point that lies distance ζ from the midsurface can be calculated.

$$\epsilon^\zeta_{\alpha\beta} = \epsilon^{mid}_{\alpha\beta} + \zeta\kappa_{\alpha\beta} \tag{2.56}$$

The stiffness matrix of the shell is derived by applying the principle of virtual work, which can be written as

$$W = \int_V \sigma : \delta\epsilon dV = \int_V f \cdot \delta u dV \tag{2.57}$$

where δu is the virtual displacement. The equation of virtual work can be further elaborated as:

$$\int_V \sigma : \delta\epsilon dV = \int_A (\mathbf{n} : \delta\epsilon + \mathbf{m} : \delta\kappa) dA \tag{2.58}$$

where the stress resultant components of the forces **n** and moments **m** are calculated via through thickness integration as

$$n_{\alpha\beta} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma^{\alpha\beta} d\zeta \tag{2.59a}$$

$$m_{\alpha\beta} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma^{\alpha\beta} \zeta d\zeta \tag{2.59b}$$

In a similar fashion, the constitutive matrices at each midsurface integration point are obtained by the thickness integration of constitutive matrices corresponding to equivalent thickness integration points.

$$C_A = \int_{-\frac{h}{2}}^{\frac{h}{2}} C d\zeta \tag{2.60a}$$

$$C_B = \int_{-\frac{h}{2}}^{\frac{h}{2}} C\zeta d\zeta \tag{2.60b}$$

$$C_D = \int_{-\frac{h}{2}}^{\frac{h}{2}} C\zeta^2 d\zeta \tag{2.60c}$$

and the gradients of the membrane forces and bending moments are computed by

$$dn^{\alpha\beta} = C_A^{\alpha\beta\gamma\delta} d\epsilon_{\gamma\delta} + C_B^{\alpha\beta\gamma\delta} d\kappa_{\gamma\delta} \tag{2.61a}$$

$$dm^{\alpha\beta} = C_B^{\alpha\beta\gamma\delta} d\epsilon_{\gamma\delta} + C_D^{\alpha\beta\gamma\delta} d\kappa_{\gamma\delta} \tag{2.61b}$$

Since equilibrium must be met for any variation of the displacement matrices δu_r, the following equation holds for the variation of the principle of virtual work:

$$\delta W = \frac{\partial W}{\partial u_r} \delta u_r = 0 \tag{2.62}$$

Based on eq. 2.62 the internal and external nodal forces can be extracted:

$$F_r^{internal} = \int_A (\mathbf{n} : \frac{\partial \epsilon}{\partial u_r} + \mathbf{m} : \frac{\partial \kappa}{\partial u_r}) dA \tag{2.63a}$$

$$F_r^{external} = \int_A f \cdot \frac{\partial u}{\partial u_r} dA \tag{2.63b}$$

Subsequently, the stiffness matrix is obtained by

$$K_{rs}^{internal} = \int_A (\frac{\partial \mathbf{n}}{\partial u_s} : \frac{\partial \epsilon}{\partial u_r} + \frac{\partial \mathbf{m}}{\partial u_s} : \frac{\partial \kappa}{\partial u_r}) \tag{2.64}$$

By substituting eq. 2.61, eq.2.64 can be further elaborated as:

$$K_{rs}^{internal} = \int_A \left((C_A : \frac{\partial \epsilon}{\partial u_s} + C_B : \frac{\partial \kappa}{\partial u_s}) : \frac{\partial \epsilon}{\partial u_r} + (C_B : \frac{\partial \epsilon}{\partial u_s} + C_D : \frac{\partial \kappa}{\partial u_s}) : \frac{\partial \kappa}{\partial u_r} \right) dA \tag{2.65}$$

2.4 Isogeometric collocation

The analysis of linear elasticity PDEs with the isogeometric Galerkin method has proven to provide results with significantly increased accuracy per degree of freedom compared to its FEM counterparts. This is mainly attributed to the increased continuity of the shape functions, that inevitably lead to minimized error and increased smoothness of second order quantities such as stresses and strain. Unfortunately, this increased accuracy comes at a great cost for the assembly of the resulting matrices, since the equivalent integration points of IGA can be orders magnitude more compared to FEM. As a result, great efforts are made by the scientific community to employ alternative integration schemes, in order to minimize this computational burden.

To this end, the well known collocation methods are examined, as a strong form alternative to the weak formulation and integration of PDEs introduced by Galerkin, that minimizes the multitude of integration points required. Thus, this section will provide a brief summary of isogeometric NURBS-based collocation methods for the analysis of linear elasticity PDEs. According to [151], given a structure represented by a domain $\Omega \subset \mathbb{R}^d$, where d the dimensionality of the structure, which is subjected to body forces **f** and the boundary Γ of the domain is divided into a portion Γ_D with prescribed displacements **g** and a portion Γ_N with prescribed tractions **h**, the strong form of the linear elasticity equation is defined as

$$\nabla \cdot (C\nabla^S \mathbf{u}) + \mathbf{f} = 0 \qquad in\ \Omega \tag{2.66a}$$

$$\mathbf{u} = \mathbf{g} \qquad on\ \Gamma_D \tag{2.66b}$$

$$(C\nabla^S \mathbf{u}) \cdot \mathbf{n} = \mathbf{h} \qquad on\ \Gamma_N \tag{2.66c}$$

In eq.2.66, the terms ∇, ∇^S are the standard and symmetric nable operators equivalently, C the elasticity tensor and **n** the outward normal vector at the boundary of the domain. In its variational form eq. 2.66 can be written as:

$$\int_\Omega (C\nabla^S \mathbf{u}) : \nabla^S \mathbf{w} d\Omega = \int_\Omega \mathbf{f} \cdot \mathbf{w} d\Omega + \int_{\Gamma_N} h \cdot \mathbf{w} d\Gamma \tag{2.67}$$

where **w** are the test functions. Integrating eq.2.67 by parts produces

$$\int_\Omega [\nabla \cdot (C\nabla^S \mathbf{u}) + \mathbf{f}] \cdot \mathbf{w} d\Omega - \int_{\Gamma_N} [(C\nabla^S \mathbf{u}) \cdot \mathbf{n} - \mathbf{h}] \cdot \mathbf{w} d\Gamma = 0 \tag{2.68}$$

It is already known from the weak Galerkin formulation that the displacement field **u** is approximated with the aid of the shape functions by

$$\mathbf{u} = \sum_{i=1}^{n} R_i u_i \tag{2.69}$$

where u_i are the unknown displacement variables. In case of collocation, the test function **w** is selected to be the Dirac delta function at each independent. The total number of Collocation Points is selected to be equal to the number of Control Points that define the geometry. The positions of the collocation points in the parametric domain can be defined by various formulas, the most common among which is by the Greville abscissae

$$\bar{\xi}_i = \frac{\sum_{k=1}^{p} \xi_{i+1}}{p} \tag{2.70}$$

Eq. 2.70 provides the univariate Collocation Points, for each one of the parametric directions of the NURBS domain. An example of bivariate Collocation points is depicted in Figure 2.9.

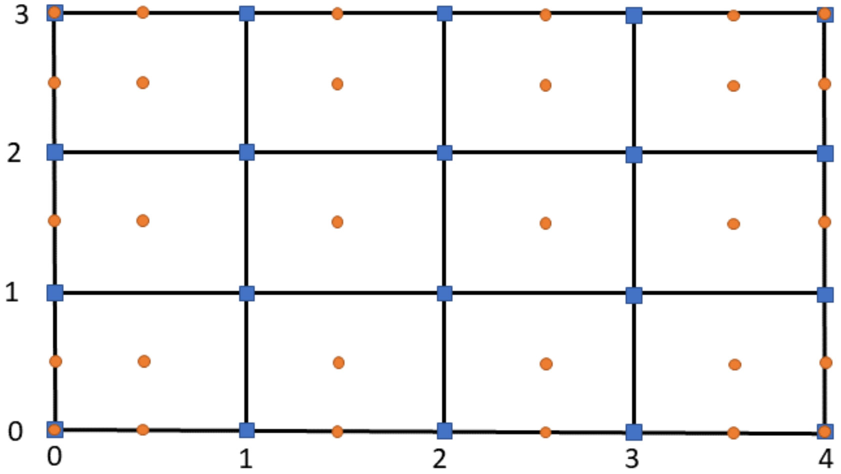


Figure 2.9: Parametric positions of Collocation Points in a two-dimensional domain. Blue rectangles represent the Knots while orange circles the Collocation points positions at the Greville absissae.

The method used to enforce boundary conditions in the collocation method is two-fold. In case of Dirichlet boundaries, no special treatment is required as the test functions utilized in the variational form of the linear elasticity PDE have a prerequisite that at the boundary the strong form of the Dirichlet conditions is satisfied.

On the other hand, in order to enforce Neumann boundary conditions a set of Dirac delta functions at the boundary collocation points have to be chosen. Here, it is important to make a distinction between the resulting equations at the interior Collocation points of the domain and the boundary ones. For the interior points $\overline{\xi}_{ij}$ the equation that governs them is defined by

$$[\nabla \cdot (C\nabla^S \mathbf{u}) + \mathbf{f}](\overline{\xi}_{ij}) = 0 \qquad \overline{\xi}_{ij} \subset \Omega \tag{2.71}$$

Equivalently, for the collocation points on the Neumann boundary, a distinction must be made among the edge collocation points and the corner ones due to the coexistence of different outwards normal vector in the second case. Specifically, for the edge case we have

$$[(C\nabla^S\mathbf{u})\cdot\mathbf{n}-\mathbf{h}](\bar{\xi}_{ij})=0 \qquad \bar{\xi}_{ij}\subset edge\subset\Gamma_N \tag{2.72}$$

and for the corner case equivalently,

$$[(C\nabla^S\mathbf{u})\cdot\mathbf{n}'-\mathbf{h}'](\bar{\xi}_{ij})+[(C\nabla^S\mathbf{u})\cdot\mathbf{n}''-\mathbf{h}''](\bar{\xi}_{ij})=0 \qquad \bar{\xi}_{ij}:corner\subset\Gamma_N \tag{2.73}$$

where $\mathbf{n}',\mathbf{n}''$. According to [161], from the former variational equations for the case of linear elasticity, the stiffness matrix K at each collocation point $\bar{\xi}_{ij}$ is defined by:

$$K_{ij}=\begin{cases} L(N_j(\bar{\xi}_{ij})), & for\ 1\leq i\leq k,\\ \mathbf{n}_i\cdot C\nabla N_j(\bar{\xi}_{ij}), & for\ k+1\leq i\leq n\end{cases} \tag{2.74}$$

and the load vector F equivalently:

$$F_i=\begin{cases} -L(\tilde{u}_D(\bar{\xi}_{ij}))+f(\bar{\xi}_{ij}), & for\ 1\leq i\leq k\\ -\mathbf{n}_i\cdot C\nabla u_D(\bar{\xi}_{ij})_i+h(\bar{\xi}_{ij}), & for\ k+1\leq i\leq n\end{cases} \tag{2.75}$$

In eqs. 2.74, 2.75, n represents the total number of Control Points, while k the number of collocation points in the interior of the domain, letting n-k be the multitude of collocation points on the boundary. Note that in the case of linear elasticity the operator L of eqs. 2.74, 2.75, is defined by

$$L\mathbf{u}+\mathbf{f}=0\Longrightarrow\nabla\cdot(C\nabla^S\mathbf{u})+\mathbf{f}=0 \qquad in\ \Omega \tag{2.76}$$

And in a similar manner the operator B can be defined for the application of Neumann boundary conditions as

$$B\mathbf{u}=\mathbf{h}\Longrightarrow(C\nabla^S\mathbf{u})\cdot\mathbf{n}=\mathbf{h} \qquad in\ \Gamma_N \tag{2.77}$$

In the following subsections, the cases of two-dimensional and three-dimensional stiffness matrices and load vectors will be briefly summarized.

2.4.1 Stiffness matrix 2D

In a similar fashion to the IGA Galerkin approach, for the extraction of the stiffness matrix, a mapping is required for the transformation of parametric quantities to the Cartesian space. Due to the isoparametric concept that also holds in case of IGA collocation methods, the Jacobian transformation is employed.

$$\underset{(2x2)}{[J]} = \begin{bmatrix} R_{1,\xi}(\xi,\eta) & R_{2,\xi}(\xi,\eta) & ... & R_{N,\xi}(\xi,\eta) \\ R_{1,\eta}(\xi,\eta) & R_{2,\eta}(\xi,\eta) & ... & R_{N,\eta}(\xi,\eta) \end{bmatrix} \cdot \begin{bmatrix} X_1 & Y_1 \\ X_2 & Y_2 \\ ... & ... \\ X_n & Y_N \end{bmatrix} \tag{2.78}$$

In collocation methods apart from first derivatives, second derivatives must also be transformed to the physical space thus raising the need for the Hessian transformation matrix, as well as the square derivatives of J_2 of the Jacobian transformation.

$$\underset{(3x2)}{[H]} = \begin{bmatrix} R_{1,\xi\xi}(\xi,\eta) & R_{2,\xi\xi}(\xi,\eta) & ... & R_{N,\xi\xi}(\xi,\eta) \\ R_{1,\eta\eta}(\xi,\eta) & R_{2,\eta\eta}(\xi,\eta) & ... & R_{N,\eta\eta}(\xi,\eta) \\ R_{1,\xi\eta}(\xi,\eta) & R_{2,\xi\eta}(\xi,\eta) & ... & R_{N,\xi\eta}(\xi,\eta) \end{bmatrix} \cdot \begin{bmatrix} X_1 & Y_1 \\ X_2 & Y_2 \\ ... & ... \\ X_n & Y_N \end{bmatrix} \tag{2.79}$$

$$\underset{(3x3)}{[J_2]} = \begin{bmatrix} J_{11}^2 & J_{11}J_{12} & J_{12}^2 \\ 2J_{11}J_{21} & J_{11}J_{22} + J_{12}J_{21} & 2J_{12}J_{22} \\ J_{21}^2 & J_{21}J_{22} & J_{22}^2 \end{bmatrix} \tag{2.80}$$

Given eqs.2.78, 2.79 and 2.80 the transformation from parameter to physical space is realized by

$$J_2 \mathbf{R}_x x = \mathbf{R}_{\xi\xi} - H^T \mathbf{R}_x \tag{2.81}$$

where $\mathbf{R}_x x, \mathbf{R}_{\xi\xi}, \mathbf{R}_x$ represent the vectors of second derivatives in physical and parametric coordinates, and the vector of first derivatives in physical space respectively. Having computed the transformations and given the Lamè constants of elasticity λ and μ, where based on Young's modulus of elasticity E and Poisson's ration ν they are calculated as

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{2.82a}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{2.82b}$$

and the operators of eqs.2.76 and 2.77 can written in matrix form as:

$$L = \begin{pmatrix} (\lambda+2\mu)\frac{\partial^2}{\partial x^2} + \mu\frac{\partial^2}{\partial y^2} & (\lambda+\mu)\frac{\partial^2}{\partial x \partial y} \\ (\lambda+\mu)\frac{\partial^2}{\partial x \partial y} & (\lambda+2\mu)\frac{\partial^2}{\partial y^2} + \mu\frac{\partial^2}{\partial x^2} \end{pmatrix} \tag{2.83}$$

$$B = \begin{pmatrix} (\lambda+2\mu)n_x\frac{\partial}{\partial x} + \mu n_y\frac{\partial}{\partial y} & \lambda n_x\frac{\partial}{\partial y} + \mu n_y\frac{\partial}{\partial x} \\ \lambda n_y\frac{\partial}{\partial x} + \mu n_x\frac{\partial}{\partial y} & (\lambda+2\mu)n_y\frac{\partial}{\partial y} + \mu n_x\frac{\partial}{\partial x} \end{pmatrix} \tag{2.84}$$

where n_x, n_y are the direction cosines of the outward normal vector $\mathbf{n}$.

2.4.2 Stiffness matrix 3D

Once again for the three-dimensional collocation case, the need arises to define the Jacobian transformation from parameter to physical space.

$$\begin{bmatrix} \frac{\partial\phi}{\partial\xi} \\ \frac{\partial\phi}{\partial\eta} \\ \frac{\partial\phi}{\partial\zeta} \end{bmatrix} = \underset{(3x3)}{[J]} \cdot \begin{bmatrix} \frac{\partial\phi}{\partial x} \\ \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial z} \end{bmatrix} \tag{2.85}$$

Analogously to the two-dimensional case, second derivatives must be transformed between physical and parameter space and thus the Hessian matrix and square derivatives matrix are computed as

$$\underset{(6x3)}{[H]} = \begin{bmatrix} R_{1,\xi\xi}(\xi,\eta,\zeta) & R_{2,\xi\xi}(\xi,\eta,\zeta) & ... & R_{N,\xi\xi}(\xi,\eta,\zeta) \\ R_{1,\eta\eta}(\xi,\eta,\zeta) & R_{2,\eta\eta}(\xi,\eta,\zeta) & ... & R_{N,\eta\eta}(\xi,\eta,\zeta) \\ R_{1,\zeta\zeta}(\xi,\eta,\zeta) & R_{2,\zeta\zeta}(\xi,\eta,\zeta) & ... & R_{N,\zeta\zeta}(\xi,\eta,\zeta) \\ R_{1,\xi\eta}(\xi,\eta,\zeta) & R_{2,\xi\eta}(\xi,\eta,\zeta) & ... & R_{N,\xi\eta}(\xi,\eta,\zeta) \\ R_{1,\xi\zeta}(\xi,\eta,\zeta) & R_{2,\xi\zeta}(\xi,\eta,\zeta) & ... & R_{N,\xi\zeta}(\xi,\eta,\zeta) \\ R_{1,\eta\zeta}(\xi,\eta,\zeta) & R_{2,\eta\zeta}(\xi,\eta,\zeta) & ... & R_{N,\eta\zeta}(\xi,\eta,\zeta) \end{bmatrix} \cdot \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ ... & ... & ... \\ X_n & Y_N & Z_N \end{bmatrix} \tag{2.86}$$

$$\underset{(6x6)}{[J_2]} = \begin{bmatrix} J_{11}^2 & J_{12}^2 & J_{13}^2 & 2J_{11}J_{12} & 2J_{11}J_{13} & 2J_{12}J_{13} \\ J_{21}^2 & J_{22}^2 & J_{23}^2 & 2J_{21}J_{22} & 2J_{21}J_{23} & 2J_{22}J_{23} \\ J_{31}^2 & J_{32}^2 & J_{33}^2 & 2J_{31}J_{32} & 2J_{31}J_{33} & 2J_{32}J_{33} \\ J_{11}J_{21} & J_{12}J_{22} & J_{13}J_{23} & (J_{11}J_{22}+J_{21}J_{12}) & (J_{11}J_{23}+J_{21}J_{13}) & (J_{12}J_{23}+J_{22}J_{13}) \\ J_{11}J_{31} & J_{12}J_{32} & J_{13}J_{33} & (J_{11}J_{32}+J_{31}J_{12}) & (J_{11}J_{33}+J_{31}J_{13}) & (J_{12}J_{33}+J_{32}J_{13}) \\ J_{21}J_{31} & J_{22}J_{32} & J_{23}J_{33} & (J_{21}J_{32}+J_{31}J_{22}) & (J_{21}J_{33}+J_{31}J_{23}) & (J_{22}J_{23}+J_{32}J_{23}) \end{bmatrix} \tag{2.87}$$

Matrices [H] and $[J_2]$ are then used in the transformations bellow

$$J\mathbf{R}_x = \mathbf{R}_\xi \tag{2.88a}$$

$$J_2\mathbf{R}_{xx} = \mathbf{R}_{\xi\xi} - H^T\mathbf{R}_x \tag{2.88b}$$

Finally, the operators L and B defined for the computation of the Collocation matrices in the three-dimensional case are defined as follows:

$$L = \begin{pmatrix} (\lambda+2\mu)\frac{\partial^2}{\partial x^2} + \mu\frac{\partial^2}{\partial y^2} + \mu\frac{\partial^2}{\partial z^2} & (\lambda+\mu)\frac{\partial^2}{\partial x\partial y} & (\lambda+\mu)\frac{\partial^2}{\partial x\partial z} \\ (\lambda+\mu)\frac{\partial^2}{\partial x\partial y} & (\lambda+2\mu)\frac{\partial^2}{\partial y^2} + \mu\frac{\partial^2}{\partial x^2} + \mu\frac{\partial^2}{\partial z^2} & (\lambda+\mu)\frac{\partial^2}{\partial y\partial z} \\ (\lambda+\mu)\frac{\partial^2}{\partial x\partial z} & (\lambda+\mu)\frac{\partial^2}{\partial y\partial z} & (\lambda+2\mu)\frac{\partial^2}{\partial z^2} + \mu\frac{\partial^2}{\partial x^2} + \mu\frac{\partial^2}{\partial y^2} \end{pmatrix} \tag{2.89}$$

$$B = \begin{pmatrix} (\lambda+2\mu)n_x\frac{\partial}{\partial x} + \mu n_y\frac{\partial}{\partial y} + \mu n_z\frac{\partial}{\partial z} & \lambda n_x\frac{\partial}{\partial y} + \mu n_y\frac{\partial}{\partial x} & \lambda n_x\frac{\partial}{\partial z} + \mu n_z\frac{\partial}{\partial x} \\ \lambda n_y\frac{\partial}{\partial x} + \mu n_x\frac{\partial}{\partial y} & (\lambda+2\mu)n_y\frac{\partial}{\partial y} + \mu n_x\frac{\partial}{\partial x} + \mu n_z\frac{\partial}{\partial z} & \lambda n_y\frac{\partial}{\partial z} + \mu n_z\frac{\partial}{\partial y} \\ \lambda n_z\frac{\partial}{\partial x} + \mu n_x\frac{\partial}{\partial z} & \lambda n_z\frac{\partial}{\partial y} + \mu n_y\frac{\partial}{\partial z} & (\lambda+2\mu)n_z\frac{\partial}{\partial z} + \mu n_x\frac{\partial}{\partial x} + \mu n_y\frac{\partial}{\partial y} \end{pmatrix} \tag{2.90}$$

3
Isogeometric solution methods

3.1 Introduction

IGA method was developed with the aim of eliminating the geometry approximation introduced by FEM and providing a tighter integration with CAD models. The strongest asset of IGA is the increased polynomial degree of the shape functions, that is accompanied with higher interelement continuity. In case of the CAD counterpart of IGA, this translates to smooth and accurate representation of curves, while for the CAE case, it results into smoother variation of the analysis characteristics such as stresses or strains. This property, however desired it may be regarding the accuracy of the results, it significantly increases the computational cost for the assembly and the solution of the produced linear systems. The increased shape function support results in increased bandwidth and population of the stiffness matrices. As a result, efficient and scalable solution methods are considered open issue towards the establishment of IGA.

A variety of methods have been proposed for addressing the solution of the resulting IGA system of equations. A relationship between solution cost and degrees od freedom is presented in [51], by examining different levels of continuity from C^0 of FEM to full continuity C^{p-1} of IGA. Similar research has been conducted for iterative solvers [49] and specifically with the preconditioned conjugate gradient method, for which various preconditioners, like diagonal-Jacobi, successive-symmetric over-relaxation (SSOR) and incomplete Cholesky factorizations were examined [141, 46, 96] to improve the computational cost for the solution of highly continuous IGA discretizations.

Similarly, DDM solution methods were investigated with the first [55] introducing overlapping additive Schwarz preconditioners for IGA. IETI method was proposed in [106] as an isogeometric variant of the Finite Element Tearing Interconnecting Dual-Primal method. Primal and mixed Schwarz preconditioner for linear elasticity systems were introduced in [179], while [27] studied the BDDC preconditioners for isogeometric scalar elliptic problems.

This chapter, will briefly discuss two categories of solution methods that are utilized latter in this manuscript. Specifically, iterative and domain decomposition solution schemes are described, with PCG and GMRES being the iterative algorithms described and FETI-DP and P-FETI-DP the domain decomposition ones. The reason for the exploration of two different algorithms in each category is that Chapters 4 and 5, address the solution of the isogeometric Galerkin and Collocation methods equivalently, that result into symmetric and non-symmetric stiffness matrices,hence making imperative the need for different algorithms.

3.2 Iterative

The first major category of solution methods considered for the estimation solution arising from structural mechanics problems is iterative solvers. They utilize an initial starting point and approximate the final solution via consecutive iterations. When a certain criterion is met, such as the error tolerance, the iterative method is considered to have converged to the final solution of the linear system. Unfortunately, the efficiency severely depends on the choice of preconditioning technique used. In this subsection, the PCG and GMRES methods will be briefly discussed, as they will be utilized later in the manuscript for the solution of symmetric and non-symmetric linear systems deriving from the isogeometric Galerkin or Collocation method equivalently. Their description is kept as concise as possible as they are considered standard textbook material.

3.2.1 PCG

PCG is considered to be one of the most widely used iterative solvers for the solution of symmetric positive definite systems, such as the ones deriving from isogeometric Galerkin processes [154, 49]. Specifically, a linear system of the form:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \tag{3.1}$$

where $\mathbf{A}$ is a symmetric positive definite matrix and let $\mathbf{x}_0$ be the initial guess solution vector, which in most cases is chosen as the zero vector, the solution $\mathbf{x}_k$ of the *k-th* iteration, the solution of the next iterative step is estimated by

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{3.2}$$

where $\mathbf{p}_k$ defined in eq. 3.2 are the direction vectors pointing at the steepest possible gradient at the point $\mathbf{x}_k$ and α_k a scalar cooeficient. These consecutive direction vectors $\mathbf{p}_k, \mathbf{p}_{k+1}$ must satisfy the following orthogonality principle

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0, \qquad \forall\ i \neq j \tag{3.3}$$

Similar to the solution vectors, the direction vectors are updated in each iteration based on their value in the previous iteration step

$$\mathbf{p}_{k+1} := \mathbf{r}_k + \beta_k \mathbf{p}_k \tag{3.4}$$

The term $\mathbf{r}_k$ of eq. 3.4 defines the residual vector and is calculated as $\mathbf{r}_k = \mathbf{f} - \mathbf{A}\mathbf{x}_k$, while β_k is another scalar coefficient. The preconditioned version of the conjugate gradient method examined here, introduced one additional vector $\mathbf{z}_k$, which represents the preconditioned residual forces vector and is calculated by

$$\mathbf{z}_{k+1} = \overline{\mathbf{A}}^{-1} \tilde{r}_k \tag{3.5}$$

Algorithm 1: PCG algorithm

```
Data: Linear system matrix: $\mathbf{A}$,
      Right hand side: $\mathbf{b}$,
      Initial solution guess: $\mathbf{x}_0$,
      Preconditioner matrix $\tilde{M}$
Result: solution vector: $\mathbf{x}_{k+1}$
$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$;
$\mathbf{z}_0 := \tilde{M}^{-1}\mathbf{r}_0$;
$\mathbf{p}_0 := \mathbf{z}_0$;
k=0;
while (convergence is not met) do
    $\alpha_k := \frac{\mathbf{r}_k^T \mathbf{z}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}$;
    $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k$;
    if $\mathbf{r}_{k+1}$ is not sufficiently small then
        $\mathbf{z}_{k+1} := \tilde{\mathbf{A}}\mathbf{r}_{k+1}$;
        $\beta_k := \frac{\mathbf{z}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{z}_k^T \mathbf{r}_k}$ ;
        $\mathbf{p}_{k+1} := \mathbf{z}_{k+1} + \beta_k \mathbf{p}_k$;
        k=k+1;
    else
        exit loop ;
    end
end
```

The choice of the preconditioning matrix $\tilde{\mathbf{A}}$ is significant for the efficiency of the method. Choosing the preconditioner to be the equal to the matrix $\mathbf{A}$, would result in the PCG converging in a single iteration, however the computational cost would be inexpedient. Thus, the preconditioner matrix is chosen to be sufficiently close to the initial matrix $\mathbf{A}$, yet easily reversible in order to minimize the resulting computational cost per required iteration. The algorithm of PCG method according to [154] is given finally provided in Algorithm 1.

3.2.2 GMRES

GMRES is an iterative solution method that belong to the Krylov subspace family of methods and extends the conjugate gradient solution scheme, for the solution of linear systems that are governed by non-symmetric matrices [154]. In a similar fashion to the PCG method that was previously analyzed, GMRES utilizes of orthogonal vectors, which due to the non-symmetric nature of the matrices must be stored for the computation of the next one. In detail, the orthogonal base of the residual vectors is Arnoldi process given by Alg. 2.

Algorithm 2: Arnoldi algorithm

```
Data: Unit vector $\mathbf{v}_1$
for $j$=1,2,..,$m$ do
    for $i$=1,2,..$j$ do
        $h_{i,j} = (\mathbf{A}\mathbf{v}_j, \mathbf{v}_i)$;
    end
    $\mathbf{w}_j = \mathbf{A}\mathbf{v}_j - \sum_{i=1}^{j} h_{i,j}\mathbf{v}_i$;
    $h_{j+1,j} = \|\mathbf{w}_j\|_2$;
    if $h_{j+1,j} = 0$ then
        exit loop
    end
    $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}}$
end
```

In Alg. 2, the quantities $h_{i,j}$ create the Hessenberg matrix and thus based on the Arnoldi process, the GMRES algorithm is formed in Alg. As a result, the approximate solution of the *i-th* iteration is calculated based on a combination of the vectors generated by the Arnoldi process and the equivalent solutions vectors of previous iterations as follows

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sum_{j=1}^{m} \beta_j \mathbf{v}_j \tag{3.6}$$

Algorithm 3: GMRES algorithm

```
Data: Matrix $\mathbf{A}$
      Right hand side $\mathbf{b}$
      Initial solution guess $\mathbf{x}_0$
$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$;
$\beta := \|\mathbf{r}_0\|_2$;
$\mathbf{v}_1 := \frac{\mathbf{r}_0}{\beta}$ ;
for $j$=1,2,..,$m$ do
    $\mathbf{w}_j := \mathbf{A}\mathbf{v}_j$;
    for $i$=1,...,$j$ do
        $h_{i,j} := (\mathbf{w}_j, \mathbf{v}_i)$;
        $\mathbf{w}_j := \mathbf{w}_j - h_{i,j}\mathbf{v}_i$;
    end
    $h_{j+1,j} = \|\mathbf{w}_j\|_2$ if $h_{j+1,j} = 0$ then
        m:=j ;
        exit all loops;
    end
    $\mathbf{v}_{j+1}\frac{\mathbf{w}_j}{h_{j+1,j}}$;
end
Compute the Hessenberg matrix $\overline{\mathbf{H}}_m$ =;
$\mathbf{y}_m = min(\left\|\beta e_1 - \overline{\mathbf{H}}_m\mathbf{y}\right\|_2)$ ;
$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m$
```

3.3 Domain Decomposition

3.3.1 PSM

Primal Substructuring Method (PSM) is considered to be a fundamental method for the solution of computational mechanics problems with subdomains [71, 33]. The underlying idea on which the method is based on, is initially the segmentation of a domain into subdomains, followed by a static condensation of the internal boundary degrees of freedom of a subdomain to the common boundary degrees of freedom between adjacent subdomains.

Specifically, for each one of the subdomains, a partitioning is performed that separates the dof into boundary, denoted as *(b)* and internal, denoted as *(*i). Boundary dof are defined by those degrees that belong to more than one subdomains. In terms of the subdomain stiffness matrix and forces vector this translates to:

$$\mathbf{K}^s = \begin{bmatrix} K^s_{ii} & K^s_{ib} \\ K^s_{bi} & K^s_{bb} \end{bmatrix} \tag{3.7a}$$

$$\mathbf{f}^s = \begin{bmatrix} f^s_i \\ f^s_b \end{bmatrix} \tag{3.7b}$$

by applying eq. 3.7 to the equilibrium equation $\mathbf{K} \cdot \mathbf{u} = \mathbf{f}$, where the internal dofs are numbered first followed by the boundary dof, it can be written as follows:

$$\begin{bmatrix} \mathbf{K}^{(1)}_{ii} & 0 & 0 & \overline{\mathbf{K}}^{(1)}_{ib} \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \mathbf{K}^{(N_s)}_{ii} & \overline{\mathbf{K}}^{(N_s)}_{ib} \\ \overline{\mathbf{K}}^{(1)}_{bi} & ... & \overline{\mathbf{K}}^{(N_s)}_{bi} & \mathbf{K_{bb}} \end{bmatrix} \cdot \begin{Bmatrix} \mathbf{u}^{(1)}_i \\ \vdots \\ \mathbf{u}^{(N_s)}_i \\ \mathbf{u}_b \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^{(1)}_i \\ \vdots \\ \mathbf{f}^{(N_s)}_i \\ \mathbf{f}_b \end{Bmatrix} \tag{3.8}$$

In eq.3.8 the terms that refer to the boundary problem of PSM, such as $\mathbf{K_{bb}}$, $\mathbf{bf}$ and $\mathbf{f}_b$ are as the sum of contribution of the boundary dofs, mapped to their equivalent global boundary dofs of the PSM problem.

The goal of PSM is to reduce the multitude of the dofs of the initial system only to the boundary dofs of the between the subdomains. This transforms the initial linear system $\mathbf{K} \cdot \mathbf{u} = \mathbf{f}$ to

$$\mathbf{S} \cdot \mathbf{u}_b = \hat{\mathbf{f}}_b \tag{3.9}$$

where the matrix $\mathbf{S}$ is the Schur complement of the initial stiffness matrix $\mathbf{K}$ to the boundary dof between the subdomain. Matrix $\mathbf{S}$ can be assembled by adding the contributions of each of the subdomain matrices as follows:

$$\mathbf{S} = \sum_{s=1}^{N_s} \overline{\mathbf{S}}^s \tag{3.10}$$

where $\overline{\mathbf{S}}$ is the contribution of the subdomain matrix to the boundary problem Schur complement, and N_s the number of subdomains.

$$\overline{\mathbf{S}}^s = (\mathbf{L}_b^s)^T \mathbf{S}^s \mathbf{L}_b^s \tag{3.11}$$

In eq.5.15, the matrix $\mathbf{L}_b$ is an unsigned Boolean matrix that maps the local boundary dof of the subdomain to the global boundary dof of the problem, while the matrix $\mathbf{S}^s$ is calculated by:

$$\mathbf{S}^s = \mathbf{K}_{bb}^s - (\mathbf{K}_{ib}^s)^T (\mathbf{K}_{ii}^s)^{-1} \mathbf{K}_{ib}^s \tag{3.12}$$

The same procedure is applied for the subdomain forces, where the right hand side of eq. 5.13 can be expanded as

$$\hat{\mathbf{f}}_b = \mathbf{f}_b - \sum_{s=1}^{N_s} (\mathbf{L}_b^s)^T (\mathbf{K}_{ib}^s)^T (\mathbf{K}_{ii}^s)^{-1} \mathbf{f}_i^s \tag{3.13}$$

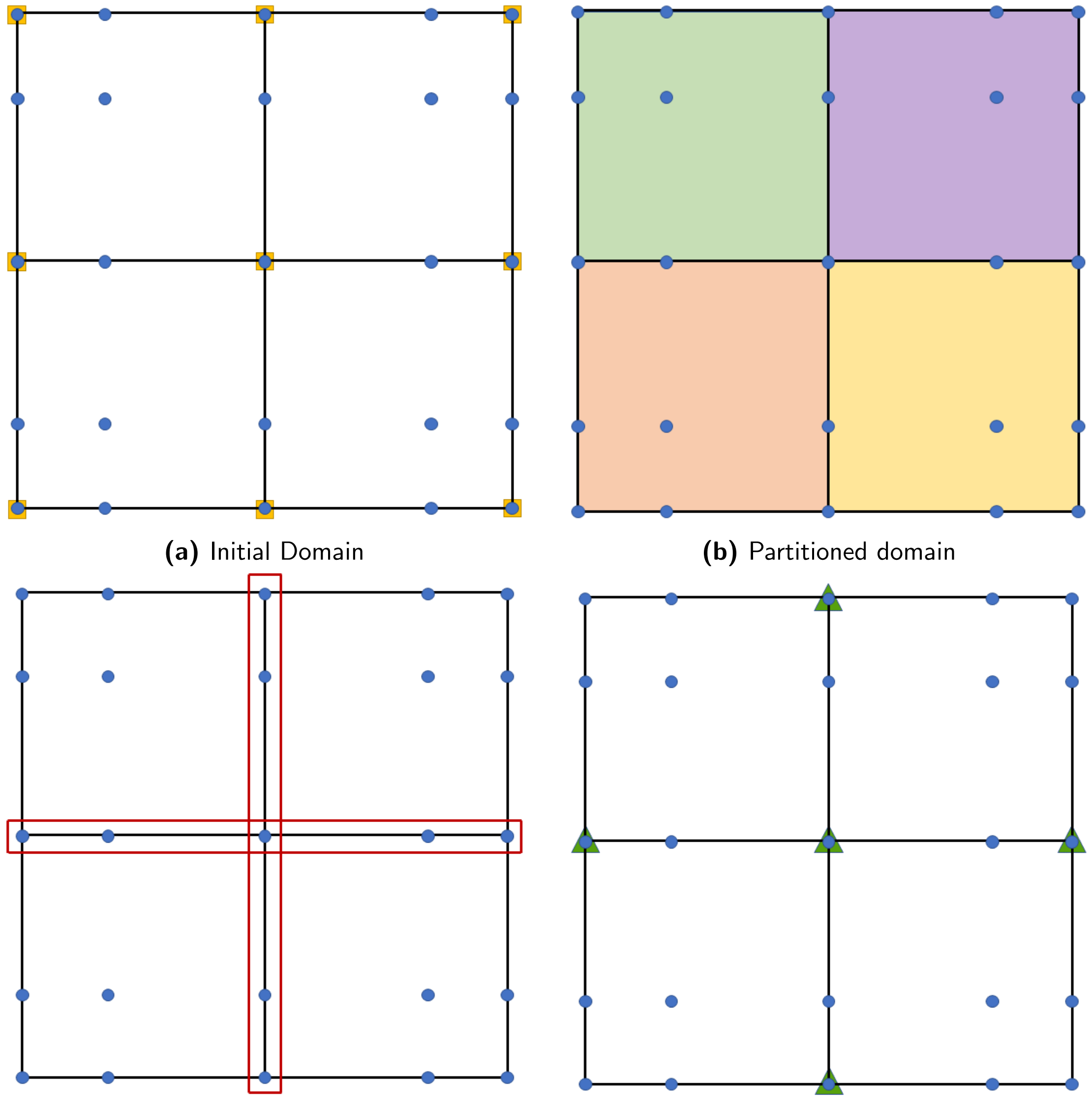
3.3.2 P-FETI-DP

P-FETI-DP method constitutes an enhancement of the aforementioned PSM method, as it introduces a preconditioner $\tilde{A}^{-1}$ to the boundary linear system of eq. 5.13, that is based on the coarse problem of the FETI-DP method. Its expression is given by:

$$\begin{aligned} \tilde{\mathbf{A}}^{-1} = (\tilde{\mathbf{L}}_{pr}^e)^T (\mathbf{K}_{rr}^e)^{-1} \tilde{\mathbf{L}}_{pr}^e + \big({}_b\mathbf{N}_{bc} - (\tilde{\mathbf{L}}_{pr}^e)^T (\mathbf{K}_{rr}^e)^{-1} \mathbf{K}_{rc}^e L_c^e\big) (\mathbf{K}_{cc}^*)^{-1} \\ \big(- (\mathbf{L}_c^e)^T \mathbf{K}_{cr}^e (\mathbf{K}_{rr}^e)^{-1} \tilde{\mathbf{L}}_{pr}^e +_{bc} \mathbf{N}_b \big) \end{aligned} \tag{3.14}$$

In order to accurately describe the mapping matrices introduced by this preconditioner, a brief description of the partitionings appearing in FETI-DP method are provided.

Figure 3.1 provides an example of a bicubic domain. Knot value vectors $\Xi = \{0, 0, 0, 0, 1, 2, 2, 2, 2\}$ and $H = \{0, 0, 0, 0, 1, 2, 2, 2, 2\}$ are utilized for the initial parameterization of the domain, which produces two isogeometric nurbs elements per direction as illustrated in Figure 3.1a. A multitude of $n = m - p - 1 = 9 - 3 - 1 = 5$ Control Points are defined per parametric direction, resulting to a total of 25 Control Points for the two-dimensional domain. In the latter expression, n represents the number of Control Points, m the number of values existing in the knot value vector and p the polynomial degree. The positions of the Control Points on the parametric domain are shown as blue circles, while the Knots that define the element boundaries are derived from the distinct



(a) Initial Domain

(b) Partitioned domain

(c) Boundary - Internal node partitioning

(d) Corner - Remainder node partitioning

Figure 3.1: P-FETI-DP node partitioning

values of the knot value vectors and are depicted with yellow squares in Figure 3.1a. Figure 3.1b shows a partitioning of the parametric domain into subdomains. Note that the subdivisioning is performed in a Control Point basis, not a element basis in order to achieve as uniform distribution among the subdomains as possible. Specifically, each parametric axis is split into two subdomains, considering only a single Control Point as their common interface. The tensor product result of this univariate subdivisioning, produces the partitioned domain, depicted as shaded areas in Figure 3.1b. At the common edges between the produced subdomains, Control Points produce an interface, that is the boundary problem of the PSM method. As already explained by eq. 3.7, this interface between the subdomains leads to a separation of the degrees of freedom into boundary and internal ones. This separation is schematically illustrated in Figure 3.1c, where nodes that belong to the interface problem of PSM are highlighted with the red rectangles. All remaining nodes belong to the internal nodes category.

Another partitioning arises from the coarse problem of FETI-DP method and is based on the definition of corner nodes. Specifically, corner nodes are defined as the nodes that either belong to the interface between more than two subdomains and lie on the interior of definition domain Ω, or belong to the interface between two subdomain and lie on the boundary $\partial\Omega$ of the domain Ω. This definition introduces a new partitioning of the subdomain nodes to corner and remainder ones. In terms of the local subdomain stiffness matrix and forces vector, this translates to :

$$\mathbf{K}^s = \begin{bmatrix} K^s_{rr} & K^s_{rc} \\ K^s_{cr} & K^s_{cc} \end{bmatrix} \tag{3.15a}$$

$$\mathbf{f}^s = \begin{bmatrix} f^s_r \\ f^s_c \end{bmatrix} \tag{3.15b}$$

This definition is illustrated in Figure 3.1d, where corner nodes are depicted as green triangles. As it can be observed only a single Control Point belongs to the case of interior points with multitude of converging subdomains greater than two. All other corner nodes belong to the the domain boundary case, with number of converging nodes equal to two. Having defined all required partitionings, the constituents of the eq. 5.31 can now be defined. Starting from left to right all unknown terms of the preconditioner are explained in detail. $\tilde{\mathbf{L}}^e_{pr}$ matrix is used to map the remainder dofs of the discretized domain and the boundary problem of the PSM method. Its expression is given by

$$\tilde{\mathbf{L}}^e_{pr} = {}_r\mathbf{N}^e_b \mathbf{L}^e_{pb} \tag{3.16}$$

The purpose of matrix ${}_r\mathbf{N}_b^e$ as its subscripts indicate, is the compatibility of matrix dimensions. When subdivided into subdomains it can be written with the following block diagonal expression

$$
{}_r\mathbf{N}_b^e = \begin{bmatrix} {}_r\mathbf{N}_b^{(1)} & & \\ & \ddots & \\ & & {}_r\mathbf{N}_b^{(N_s)} \end{bmatrix} \tag{3.17}
$$

Each of the subdomain contributions ${}_r\mathbf{N}_b^{(N_s)}$ is a Boolean matrix that maps the remainder dofs of the subdomain s to the boundary dofs the same subdomain. The matrix $\mathbf{L}_{pb}^e$ of eq. 5.32 can be similarly written

$$
\mathbf{L}_{pb}^e = \begin{bmatrix} \mathbf{L}_{pb}^{(1)} \\ \vdots \\ \mathbf{L}_{pb}^{(N_s)} \end{bmatrix} \tag{3.18}
$$

where for each subdomain the matrix $\mathbf{L}_{pb}^{(s)}$ is computed as

$$
\mathbf{L}_{pb}^{(s)} = \mathbf{L}_b^s \mathbf{W} \tag{3.19}
$$

In eq. 5.35, the matrix $\mathbf{L}_b^s$ is a Boolean matrix that maps local boundary dofs of the subdomain to the global boundary dofs of the PSM problem. Additionall, the matrix $\mathbf{W}$ is a diagonal matrix, with terms equal to the inverse multiplicity of the boundary dofs. The term multiplicity for the case of boundary dofs, refers to the number of subdomains this dof belongs to. Moving to the next undefined terms $\mathbf{K}_{rr}^e, \mathbf{K}_{rc}^e$, these constitute block diagonal matrices generates from the partitioning of the subdomain matrices to corner and remainder dofs as per eq. 3.15a.

$$
\mathbf{K}_{rr}^e = \begin{bmatrix} \mathbf{K}_{rr}^{(1)} & & \\ & \ddots & \\ & & \mathbf{K}_{rr}^{(N_s)} \end{bmatrix} \tag{3.20a}
$$

$$
\mathbf{K}_{rc}^e = \begin{bmatrix} \mathbf{K}_{rc}^{(1)} & & \\ & \ddots & \\ & & \mathbf{K}_{rc}^{(N_s)} \end{bmatrix} \tag{3.20b}
$$

while the $\mathbf{K}^{*}_{cc}$ matrix is derived as the Schur complement of the remainder dofs of each subdomain, to its boundary corner ones and consequently its mapping to the global boundary dofs of the corner problem define by FETI-DP. The matrix ${}_{b}\mathbf{N}_{bc}$ of eq. 5.31, is another Boolean mapping matrix, that links the boundary dofs of the PSM problem, to the corner dofs of the FETI-DP method. Finally, the matrix $\mathbf{L}_c$ is written in subdomain form as

$$\mathbf{L}^{e}_{c} = \begin{bmatrix} \mathbf{L}^{(1)}_{c} \\ \vdots \\ \mathbf{L}^{(N_s)}_{c} \end{bmatrix} \tag{3.21}$$

and defines an unsigned Boolean matrix that links the local corner dof of each subdomain to the equivalent corner dof of the coarse problem. Note that for all the Boolean matrices involved in this process no matrix vector multiplications are required, rather the utilization of the proper submatrix.

3.3.3 FETI-DP

FETI-DP method, belongs to the category of dual DDM methods, that chooses as a solution quantity for the boundary problem, the traction forces developed between subdomains.

Similar to the aforementioned DDM methods, FETI-DP is based on the partitioning of subdomain matrices based on their geometrical or interface properties with adjacent subdomain. Thus, given a subdomain s, and its stiffness matrix $\mathbf{K}^{s}$, displacement vector $\mathbf{u}^{s}$ and force vector $\mathbf{f}^{s}$, it can be partitioned as follows:

$$\mathbf{K}^{s} = \begin{bmatrix} \mathbf{K}^{s}_{ii} & \mathbf{K}^{s}_{ib} \\ (\mathbf{K}^{s}_{ib})^{T} & \mathbf{K}^{s}_{bb} \end{bmatrix} \tag{3.22a}$$

$$\mathbf{u}^{s} = \begin{Bmatrix} \mathbf{u}^{s}_{i} \\ \mathbf{u}^{s}_{b} \end{Bmatrix} \tag{3.22b}$$

$$\mathbf{f}^{s} = \begin{Bmatrix} \mathbf{f}^{s}_{i} \\ \mathbf{f}^{s}_{b} \end{Bmatrix} \tag{3.22c}$$

The subscripts b and i of eq. 3.22, denote the boundary and internal dofs of the subdomain equivalently. The boundary dofs in analogy to the PSM method, are the degrees of freedom that are shared among multiple subdomains, while internal dof, belong solely to a single subdomain. A further partitioning of the boundary dof, results into the following expression for the boundary displacement terms:

$$\mathbf{u}_b^s = \begin{Bmatrix} \mathbf{u}_{br}^s \\ \mathbf{u}_{bc}^s \end{Bmatrix} \tag{3.23}$$

Again the subscripts c and r, denote the partitioning of the boundary degrees of freedom into corner and remainder ones. Corner nodes and dof, are those that belong to either of the following two categories. The first category includes nodes that lie on the crosspoints between subdomains and as a result this point belong to more than two subdomains. On the other hand, the second category takes into account nodes that lie of the boundary of the domain and at the same time belong to an interface between adjacent subdomains. Given two adjacent subdomains s_1 and s_2, the expression that defines their displacement compatibility is given by

$$\mathbf{u}_b^{(s_1)} - \mathbf{u}_b^{(s_2)} = 0 \tag{3.24}$$

Eq. 3.24 can be extended to include the compatitbility equations for all subdomains

$$\sum_{s=1}^{N_s} \mathbf{B}^s \mathbf{u}^s = 0 \tag{3.25}$$

where $\mathbf{B}^s$ is a signed Boolean matrix with values $(+1, -1, 0)$ that maps the local subdomain boundary degrees of freedom, to the global boundary dofs of the problem. Eq. 3.25 represents the compatibility equation for all subdomains of the structure which is enforced by introducing Lagrange multipliers, while at the same time, the displacement compatibility at the corner nodes is ensured.

To this end, the stiffness matrix $\mathbf{K}^s$, the displacement vector $\mathbf{u}^s$ and force vector $\mathbf{f}^s$ of an arbitrary subdomain s, can be further partitioned as follows:

$$\mathbf{K}^s = \begin{bmatrix} \mathbf{K}^s_{rr} & \mathbf{K}^s_{rc} \\ (\mathbf{K}^s_{rc})^T & \mathbf{K}^s_{cc} \end{bmatrix} \tag{3.26a}$$

$$\mathbf{u}^s = \begin{Bmatrix} \mathbf{u}^s_r \\ \mathbf{u}^s_{bc} \end{Bmatrix} \tag{3.26b}$$

$$\mathbf{f}^s = \begin{Bmatrix} \mathbf{f}^s_r \\ \mathbf{f}^s_{bc} \end{Bmatrix} \tag{3.26c}$$

where the remainder dof include the internal nodes of the subdomain and the boundary nodes that do not belong to the corner category. As a result, the stiffness matrix can be further elaborated as

$$\mathbf{K}^s = \begin{bmatrix} \mathbf{K}^s_{ii} & \mathbf{K}^s_{ibr} & \mathbf{K}^s_{ic} \\ (\mathbf{K}^s_{ibr})^T & \mathbf{K}^s_{brbr} & \mathbf{K}^s_{brc} \\ (\mathbf{K}^s_{ic})^T & (\mathbf{K}^s_{brc})^T & \mathbf{K}^s_{cc} \end{bmatrix} \tag{3.27}$$

Additionally to the global displacement vector, the corner dof displacement vector is introduced.

$$\mathbf{u}_c = \begin{Bmatrix} \mathbf{u}^{(1)}_c \\ \vdots \\ \mathbf{u}^{(N_s)}_c \end{Bmatrix} \tag{3.28}$$

In a similar fashion to eq. 3.25, another compatibility equation is added

$$\sum_{s=1}^{N_s} \mathbf{B}^s_r \mathbf{u}^s_r = 0 \tag{3.29}$$

where $\mathbf{B}^s_r$ is a signed matrix with values $(1, -1, 0)$ that maps the remainder dofs of the subdomain to to the global boundary remainder dofs. In addition, for each subdomain, a new Boolena mapping matrix $\mathbf{L}^s_c$ is introduced which connects the local corner dofs of a subdomain to the global corner dofs as follows

$$\mathbf{L}^s_c \mathbf{u}_c = \mathbf{u}^s_{bc} \tag{3.30}$$

Taking into account all the above equation, the final linear system to be solved takes the following form:

$$\begin{bmatrix} \mathbf{K}^e & (\mathbf{B}^e)^T \\ \mathbf{B}^e & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{u}^e \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^e \\ \mathbf{0} \end{Bmatrix} \tag{3.31}$$

where the matrix $\mathbf{K}^e$ is a block diagonal matrix that contains the local subdomain stiffness matrices, while matrix $\mathbf{B}^e$ is derived based on the compatibility of the remainder dofs of eq. 3.29

$$\mathbf{B}^e = \left[\begin{bmatrix} \mathbf{B}_r^{(1)} & 0 \end{bmatrix} \quad ... \quad \begin{bmatrix} \mathbf{B}_r^{(N_s)} & 0 \end{bmatrix} \right] \tag{3.32}$$

With the aid of eqs. 3.26,3.29,3.30,3.31 the equilibrium equation for the remainder dof is obtained.

$$\mathbf{K}_{rr}^s \mathbf{u}_r^s + \mathbf{K}_{rc}^s \mathbf{L}_c^s \mathbf{u}_c = \mathbf{f}_r^s - (\mathbf{B}_r^s)^T \lambda \tag{3.33}$$

The additional equilibrium equation for the displacements at the corner dof is expressed as:

$$\sum_{s=1}^{N_s} (\mathbf{L}_c^s)^T (\mathbf{K}_{rc}^s)^T \mathbf{u}_r^s + \sum_{s=1}^{N_s} (\mathbf{L}_c^s)^T \mathbf{K}_{cc}^s \mathbf{L}_c^s \mathbf{u}_c = + \sum_{s=1}^{N_s} (\mathbf{L}_c^s)^T \mathbf{f}_{bc}^s = \mathbf{f}_c \tag{3.34}$$

where the matrix $\mathbf{K}_{cc}$ is computed by

$$\mathbf{K}_{cc} = \sum_{s=1}^{N_s} (\mathbf{L}_c^s)^T \mathbf{K}_{cc}^s \mathbf{L}_c^s \tag{3.35}$$

Assuming that selection of corner dof is adequate, so that given their values all subdomains are considered sufficiently supported and thus the matrix $\mathbf{K}_{rr}^s$ is reversible, so eq. 3.33 can be written as:

$$\mathbf{u}_r = (\mathbf{K}_{rr}^s)^{-1} [\mathbf{f}_r^s - (\mathbf{B}_r^s)^T \lambda - \mathbf{K}_{rc}^s \mathbf{L}_c^s \mathbf{u}_c] \tag{3.36}$$

Replacing the latter to eq. 3.29 we obtain:

$$\sum_{s=1}^{N_s} \mathbf{B}_r^s (\mathbf{K}_{rr}^s)^{-1} [\mathbf{f}_r^s - (\mathbf{B}_r^s)^T \lambda - \mathbf{K}_{rc}^s \mathbf{L}_c^s \mathbf{u}_c] = 0 \tag{3.37a}$$

$$\sum_{s=1}^{N_s} \mathbf{B}_r^s (\mathbf{K}_{rr}^s)^{-1} \mathbf{f}_r^s - \sum_{s=1}^{N_s} \mathbf{B}_r^s (\mathbf{K}_{rr}^s)^{-1} (\mathbf{B}_r^s)^T \lambda - \sum_{s=1}^{N_s} \mathbf{B}_r^s (\mathbf{K}_{rr}^s)^{-1} \mathbf{K}_{rc}^s \mathbf{L}_c^s \mathbf{u}_c = 0 \tag{3.37b}$$

For the sake of simplification, the following terms of eq. 3.37 are defined as

$$\mathbf{d}_r = \sum_{s=1}^{N_s} \mathbf{B}_r^s (\mathbf{K}_{rr}^s)^{-1} \mathbf{f}_r^s \tag{3.38a}$$

$$\mathbf{F}_{I_{rr}} = \sum_{s=1}^{N_s} \mathbf{B}_r^s (\mathbf{K}_{rr}^s)^{-1} (\mathbf{B}_r^s)^T \tag{3.38b}$$

$$\mathbf{F}_{I_{rc}} = \sum_{s=1}^{N_s} \mathbf{B}_r^s (\mathbf{K}_{rr}^s)^{-1} \mathbf{K}_{rc}^s \tag{3.38c}$$

which translate eq. 3.37 to

$$\mathbf{F}_{I_{rr}} \lambda + \mathbf{F}_{I_{rc}} \mathbf{u}_c = \mathbf{d}_r \tag{3.39}$$

In a similar fashion, replacing eq. 3.36 to eq. 3.34 the following expression is derived

$$\sum_{s=1}^{N_s} (\mathbf{L}_c^s)^T (\mathbf{K}_{rc}^s)^T (\mathbf{K}_{rr}^s)^{-1} [\mathbf{f}_r^s - (\mathbf{B}_r^s)^T \lambda - \mathbf{K}_{rc}^s \mathbf{L}_c^s \mathbf{u}_c] + \mathbf{K}_{cc} \mathbf{u}_c = \mathbf{f}_c \tag{3.40}$$

which by defining

$$\mathbf{K}_{cc}^* = \mathbf{K}_{cc} - \sum_{s=1}^{N_s} (\mathbf{L}_c^s)^T (\mathbf{K}_{rc}^s)^T (\mathbf{K}_{rr}^s)^{-1} \mathbf{K}_{rc}^s \mathbf{L}_c^s \tag{3.41a}$$

$$\mathbf{f}_c^* = \mathbf{f}_c - \sum_{s=1}^{N_s} (\mathbf{L}_c^s)^T (\mathbf{K}_{rc}^s)^T (\mathbf{K}_{rr}^s)^{-1} \mathbf{f}_r^s \tag{3.41b}$$

turns into

$$(\mathbf{F}_{I_{rc}})^T\lambda - \mathbf{K}^*_{cc}\mathbf{u}_c = -\mathbf{f}^*_c \tag{3.42}$$

The eqs.3.39, 3.42 represent a dual-primal linear system, as its unknown quantities are both the Lagrange multipliers, as well as displacements of the corner dofs

$$\begin{bmatrix} \mathbf{F}_{I_{rr}} & \mathbf{F}_{I_{rc}} \\ (\mathbf{F}_{I_{rr}})^T & -\mathbf{K}^*_{cc} \end{bmatrix} \begin{Bmatrix} \lambda \\ \mathbf{u}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{d}_r \\ -\mathbf{f}^*_c \end{Bmatrix} \tag{3.43}$$

The second hypothesis of FETI-DP method is introduced in this step regarding the corner displacements $\mathbf{u}_c$, which assumes that that if the subdomains are connected only to the corner nodes, then these connections suffice to transfer the loads applied to the strucure to its external supports. This results to the matrix $\mathbf{K}^*_{cc}$ being reversible, thus from the second equation of the linear system in eq.3.43, the displacement is of the corner dof is obtained as

$$\mathbf{u}_c = (\mathbf{K}^*_{cc})^{-1}[\mathbf{f}^*_c + (\mathbf{F}_{I_{rc}})^T\lambda] \tag{3.44}$$

By replacing eq. 3.44 to the first equation of the linear system in eq.3.43 the final equation of FETI-DP method is produced

$$[\mathbf{F}_{I_{rr}} + \mathbf{F}_{I_{rc}}(\mathbf{K}^*_{cc})^{-1}(\mathbf{F}_{I_{rc}})^T]\lambda = \mathbf{d}_r - \mathbf{F}_{I_{rc}}(\mathbf{K}^*_{cc})^{-1}\mathbf{f}^*_c \tag{3.45}$$

4

Non-overlapping domain decomposition solution schemes for structural mechanics isogeometric analysis

4.1 Introduction

This Chapter focuses on the development of a family of solution algorithms for the purposes of isogeometric Galerkin methods that exploit the advantages of an iterative solution scheme, namely PCG coupled with domain decomposition methods. Specifically, by introducing an appropriate modification to the NURBS shape function overlapping nature, a non-overlapping equivalent of the initial domain is introduced, that possess the same geometrical features with the initial one, yet has artificially induced discontinuities in the form of truncated shape functions. This allows the development of a robust and scalable preconditioner that minimizes the computational cost in both sequential as well as high-performance parallel computing environments. In this work, the PCG iterative solver is coupled with the IETI domain decomposition method, in order to investigate the numerical characteristics of the proposed method.

4.2 Existing solution techniques

Due to the capabilities introduced by the IGA, along with the CAD tools, it has gained a wide acceptance by the scientific community. This rapid evolution creates the need for efficient computational methods for solving complex problems. In both FEM and IGA methods the resulting algebraic equation linear system has the form:

$$K \cdot u = f \tag{4.1}$$

where K is the stiffness matrix, f is the external load vector and u the vector of unknown displacements. The computational effort required for the solution of eq. 4.1 highly depends on the size and the sparsity pattern of the stiffness matrix. In the case of IGA, shape functions of higher continuity that enable exact geometrical representation, reduce the sparsity and increase the computational cost for the solution. Specifically, the overlapping of shape functions through several IGA elements leads to denser matrices and higher interelement continuity, while the utilization of higher polynomial degrees leads to a more computational demanding solution procedure compared to FEM for the same dof.

This characteristic feature of IGA may hinder its capability of addressing large-scale problems unless cost-effective solution methods are developed to address the inherent features of IGA. Both direct and iterative solution schemes have been utilized for the efficient treatment of large-scale IGA problems, but their computational efficiency suffers as continuity increases [50]. At the same time, DDM were proposed for coupled subdomains [153, 9, 83, 35] and were implemented for the solution of IGA simulated problems [56, 24, 27, 57, 26, 86, 43]. Furthermore, IETI [106, 86] was among the first attempts to implement DDM for IGA problems. This method, is based on the well-established FETI-DP solver [69, 102, 120, 63, 104] which combines both primal and dual DDM formulations. In IETI, the subdivision in subdomains coincides with the non-overlapping patterns of patches of the model. This makes the number and the size of subdomains geometry-dependent, severely impairing load balancing and scalability in paralle architectures.

4.2.1 The IETI method

IETI was introduced in [106] as an extension to the FETI family of methods [63, 70, 185, 138]. It utilizes the exact geometrical representation of IGA with the advanced subdomain coupling and solution capabilities of FETI methods. Specifically, the NURBS representation of a structure is naturally subdivided into patches, to accurately describe abrupt changes of geometry. The boundary between these patches describes the interface problem of IETI where each patch coincides with a subdomain and thus constitutes a local problem to be solved in parallel.

The stiffness matrix K^s, force vector f^s and displacement vector u^s of each patch (s) can be partitioned as follows:

$$K^s = \begin{bmatrix} K^s_{ii} & K^s_{ib} \\ K^s_{bi} & K^s_{bb} \end{bmatrix} \tag{4.2a}$$

$$f^s = \begin{bmatrix} f^s_i \\ f^s_b \end{bmatrix} \tag{4.2b}$$

$$u^s = \begin{bmatrix} u^s_i \\ u^s_b \end{bmatrix} \tag{4.2c}$$

with the (i) subscript denoting the internal dof and the (b) subscript denoting the boundary dof.

In order to impose continuity between patches, the following equation is introduced:

$$\sum_{s=1}^{N_s} B^s u^s = 0 \tag{4.3}$$

with N_s being the number of patches and B^s being a signed Boolean matrix. The u^s_b vector is further partitioned as:

$$u^s_b = \begin{bmatrix} u^s_{br} \\ u^s_{bc} \end{bmatrix} \tag{4.4}$$

with the subscript (c) denoting corner nodes which are usually defined as nodes belonging to more than two patches. Subsequently, K^s, u^s and f^s can be partitioned as follows:

$$K^s = \begin{bmatrix} K^s_{rr} & K^s_{rc} \\ K^s_{cr} & K^s_{cc} \end{bmatrix} \tag{4.5a}$$

$$f^s = \begin{bmatrix} f^s_r \\ f^s_{bc} \end{bmatrix} \tag{4.5b}$$

$$u^s = \begin{bmatrix} u^s_r \\ u^s_{bc} \end{bmatrix} \tag{4.5c}$$

where

$$f^s_r = \begin{bmatrix} f^s_i \\ f^s_{br} \end{bmatrix} \tag{4.6a}$$

$$u^s_r = \begin{bmatrix} u^s_i \\ u^s_{br} \end{bmatrix} \tag{4.6b}$$

and subscript (r) denoting all dof that do not belong to a corner node. Considering this repartitioning, the continuity between patches can be re-written as:

$$\sum_{s=1}^{N_s} B^s_r u^s_r = 0 \tag{4.7}$$

with B^s_r being the signed Boolean matrix B^s, pertaining to the (r) dof. Moreover, the following global vector of corner dof is introduced

$$u_c = \begin{bmatrix} u^1_c & ... & u^j_c & ... & u^{N_c}_c \end{bmatrix}^T \tag{4.8}$$

with u^j_c denoting all of the displacement dof attached to the j-th global node that is also a corner node of the mesh decomposition and N_c denoting the total number of corner nodes.

In order to connect u_c with u^s_{bc}, the following equation is introduced:

$$B^s_c u_c = u^s_{bc} \tag{4.9}$$

with B_c^s being an unsigned Boolena matrix.

Considering all the above equations, the patch equilibrium is expressed as:

$$K_{rr}^s U_r^s + K_{rc}^s B_c^s u_c = f_r^s - B_r^{sT} \lambda \tag{4.10a}$$

$$\sum_{s=1}^{N_s} B_c^{sT} {K_{rc}^s}^T u_r^s + \sum_{s=1}^{N_s} B_c^{sT} K_{cc}^s u_c = \sum_{s=1}^{N_s} B_c^{sT} f_{bc}^s = f_c \tag{4.10b}$$

with λ being the traction forces between each Patch.

From the above relations, it follows that

$$u_r^s = {K_{rr}^s}^{-1} (f_r^s - B_r^{sT} \lambda - K_{rc}^s B_c^s u_c) \tag{4.11}$$

which, when substituted to eq. 5.23, the following system of equations arises

$$\begin{bmatrix} F_{I_{rr}} & F_{I_{rc}} \\ {F_{I_{rc}}}^T & -K_{cc}^* \end{bmatrix} \begin{bmatrix} \lambda \\ u_c \end{bmatrix} = \begin{bmatrix} d_r \\ f_c^* \end{bmatrix} \tag{4.12}$$

where:

$$F_{I_{rr}} = \sum_{i=1}^{N_s} B_r^s {K_{rr}^s}^{-1} B_r^{sT} \tag{4.13a}$$

$$F_{I_{rc}} = \sum_{i=1}^{N_s} B_r^s {K_{rr}^s}^{-1} K_{rc}^s B_c^s \tag{4.13b}$$

$$K_{cc} = \sum_{i=1}^{N_s} B_c^{sT} K_{cc}^s B_c^s \tag{4.13c}$$

$$K_{cc}^* = K_{cc} - \sum_{i=1}^{N_s} (K_{rc}^s B_c^s)^T {K_{rr}^s}^{-1} (K_{rc}^s B_c^s) \tag{4.13d}$$

$$d_r = \sum_{i=1}^{N_s} B_r^s {K_{rr}^s}^{-1} f_r^s \tag{4.13e}$$

$$f_c^* = f_c - \sum_{i=1}^{N_s} (K_{rc}^s B_c^s)^T {K_{rr}^s}^{-1} f_r^s \tag{4.13f}$$

By eliminating u_c from the baove equations, the following interface problem needs to be solved:

$$(F_{I_{rr}} + F_{I_{rc}} {K_{cc}^*}^{-1} F_{I_{rc}}^T)\lambda = d_r - F_{I_{rc}} {K_{cc}^*}^{-1} f_c^* \tag{4.14}$$

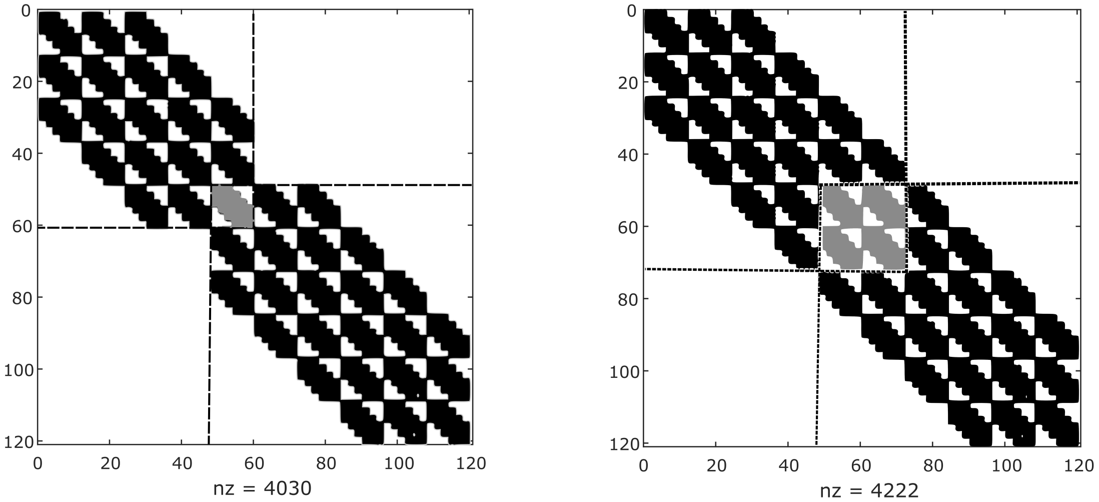
The solution of eq.5.30 is usually performed with an iterative solution algorithm and for the case of structural mechanics where positive definite matrices occur, the PCG algorithm is used.

4.2.2 Handling the interface problem: Multi-patch (IETI-P) and Overlapping IETI (IETI-O)

As described in the previous section, the boundary between the patches of an IGA model describes the interface problem of IETI, with each patch defining a subdomain. Usually, these patches occur in IGA when designing the model and are introduced in order to accurately describe the geometrical characteristics of the model. However, such patches can also be introduced by the designer, in an arbitrary fashion, without disrupting the aforementioned geometrical characteristics. In this work, a multi-patch IETI variant called IETI-P, is considered where the model to be solved is altered with the introduction of additional patches that serve the purpose of defining IETI subdomains. Such introduction can have a positive impact on the IETI's mesh dependent load balancing and scalability, at the expense of introducing more dof.

An alternative approach for the definition of subdomains it the IETI-O variant by applying the overlapping approach as first described in [130], for the case of element-free Galerkin simulation methods, and later in [27, 57] for the BDDC where the resulting interface is referred to as ”fat interface”. Specifically, a physical domain subdivision is performed along Control Points instead of relying to the existing patches. Due to the continuity characteristics of the shape functions, the interface problem (Figure 4.1b) in case of two overlapping subdomains is larger when compared to the interface problem of two patchesin IETI (Figure 4.1a). However, as in the case of IETI-P, this consideration alleviates the problem of path dependency encountered in IETI. The difference of overlapping terms of the stiffness matrix is visualized in Figure 4.1 for a 2D domain. The gray area depicts the elements associated to the interface dofs.

Both IETI-O and IETI-P share the same characteristics, pertaining to their mathematical properties as well as their numerical and parallel scalability, with IETI [106]. This stems from the fact that these IETI variants differ only, either in the magnitude of the interface problem or in the way the domain is divided into subdomains. properties and characteristics of IETI have been throughly investigated in [106, 27, 57, 69, 102, 77].



(a) IETI-P

(b) IETI-O

Figure 4.1: Global stiffness matrix graphs of a 2D domain for p=2 and the interface of two subdomains.

4.3 PCG-IETI-N

Despite its favorite features, IETI has some drawbacks that decrease its efficiency, with the most important one being the dependency between partitioning and structure geometry. Specifically, IETI relies on patches already present in the model to define the subdomains of the structure, making partitioning geometry-dependent and thus severely impairing both load balancing and scalability, especially in fine grain high-performance computing environments. The proposed alternative versions of IETI-P and IETI-O overcome the problem of patch dependency, by either introducing new patches or be extending the interface problem. This comes at the expense of increasing the overlapping domains, since a multitude of new Control points are introduced, and the resulting interface problem can become prohibitively large as shape function continuity increases.

In order to alleviate the aforementioned mesh dependency of partitioning without the associated drawbacks, a solution methodology for structural mechanics problems is proposed thta combines the advantages of both PCG [154] and DDM. In addition to the original model with stiffness matrix K, the proposed algorithm introduces another model with stiffness matrix K_p, referred to as the "non-overlapping model". The procedure for building K_p will be described in the next section. The non-overlapping model has identical properties and geometry to the original model, but its shape functions have been truncated at the boundaries of the subdomains. These discontinuities result in non-overlapping subdomains, but at the same time reduce the accuracy of the original model.

Matrices K and K_p have the same size and due to their similarity, they are connected with the following relation:

$$K = K_p + \Delta K \tag{4.15}$$

where ΔK is a sparse matrix with the size of K and K_p, with terms only near the interface where continuity has been reduced. By substituting eq. 4.15 to eq. 4.1, the linear system at hand can be written as:

$$(K_p + \Delta K)u = f \tag{4.16}$$

Following the rationale of incomplete Cholesky preconditionings[73], ΔK can be considered as the error matrix E contributing to the small terms in the lower triangular matrix produced by the Cholesky factorization, which can be ignored of they do not satisfy a specified magnitude criterion. In the proposed method, the preconditioning matrix K_{prec} of the iterative method used (i.e.:the PCG method[154] in this work) becomes the stiffness matrix K_p of the non-overlapping model. This means that for each iteration, a linear system involving the non-overlapping model needs to be solved , in order to evaluate the preconditioned residual vector $z_{j+1} = {K_{prec}}^{-1} r_{j+1}$.

The solution of the equation, $Ku = f$, where the evaluation of the preconditioned residual at each iteration is performed with IETI, with $K_{prec} = K_p$, constitutes the PCG-IETI non-overlapping (PCG-IETI-N) method. In particular, the residual vector r_{j+1}, becomes the load vector ofr IETI which is split into N_s parts, with N_s being the number of non-overlapping subdomains. The load vector of each subdomain is equal to

$$L^s r_{j+1} = f^s \tag{4.17}$$

with L^s being an unsigned Boolean matrix. By partitioning the non-overlapping subdomain quantities as follows

$$K^s = \begin{bmatrix} K^s_{p_{rr}} & K^s_{p_{rc}} \\ {K^s_{p_{rc}}}^T & K^s_{p_{cc}} \end{bmatrix} \tag{4.18a}$$

$$u^s = \begin{bmatrix} u^s_r \\ u^s_{b_c} \end{bmatrix} \tag{4.18b}$$

$$f^s = \begin{bmatrix} f^s_r \\ f^s_{b_c} \end{bmatrix} \tag{4.18c}$$

the problem of eq. 5.30 is solved with respect to λ. Following the computation of λ, vector u_c is evaluated from the second eq.5.26 as follows:

$$u_c = {K^*_{cc}}^{-1}({F_{I_{rc}}}^T\lambda - f^*_c) \tag{4.19}$$

Then, vector z_{j+1} is computed for each subdomain as:

$$z^s_{j+1} = {K^s_{p_{rr}}}^{-1}(f^s_r - B^{sT}_r)\lambda - K^s_{p_{rc}} B^s_c u_c \tag{4.20}$$

and global vector z_{j+1}, by

$$z_{j+1} = \sum_{s=1}^{N_s} W^s L^s z^s_{j+1} \tag{4.21}$$

where W^s being a scaling diagonal subdomain matrix that accounts for eventual subdomain heterogeneities [152].

The repeated solutions required for the evaluation of the preconditioned residual vector z_{j+1} are treated as problems with multiple right-hand sides, as described in [77, 172, 73, 41], further enhancing the convergence properties of PCG-IETI-N.

4.4 Building the non-overlapping stiffness matrix

In order to further explain the process of constructing the non-overlapping model and its stiffness matrix, a brief overview of Gauss integration and shape function evaluation, as performed in IGA, is provided in the following subsections. Advanced quadrature techniques for IGA are presented in [16, 12, 6, 123, 38, 132, 94, 67].

4.4.1 Gauss integration

According to [93], in order to accurately integrate a piecewise polynomial of degree p, $\frac{p+1}{1}$ and $\frac{p+2}{2}$, Gauss Points are required for odd and even degrees, respectively. Numerical integration is performed for the evaluation of the stiffness matrix [176], which is an integral of the form

$$K = \int_{\Omega} B^T E B d\Omega \tag{4.22}$$

with E being the constitutive matrix and B the deformation matrix which is evaluated from the derivatives of the shape functions. In the one-dimensional case and for polynomial degree p of the shape functions, the differentiation produces a (p-1) degree polynomial. As a result, the stiffness matrix of eq.4.22 is a polynomial of maximum degree $(p-1)+(p-1) = 2p-2$ which is an even number. The minimum number of Gauss Points for integrating an even degree polynomials is equal to:

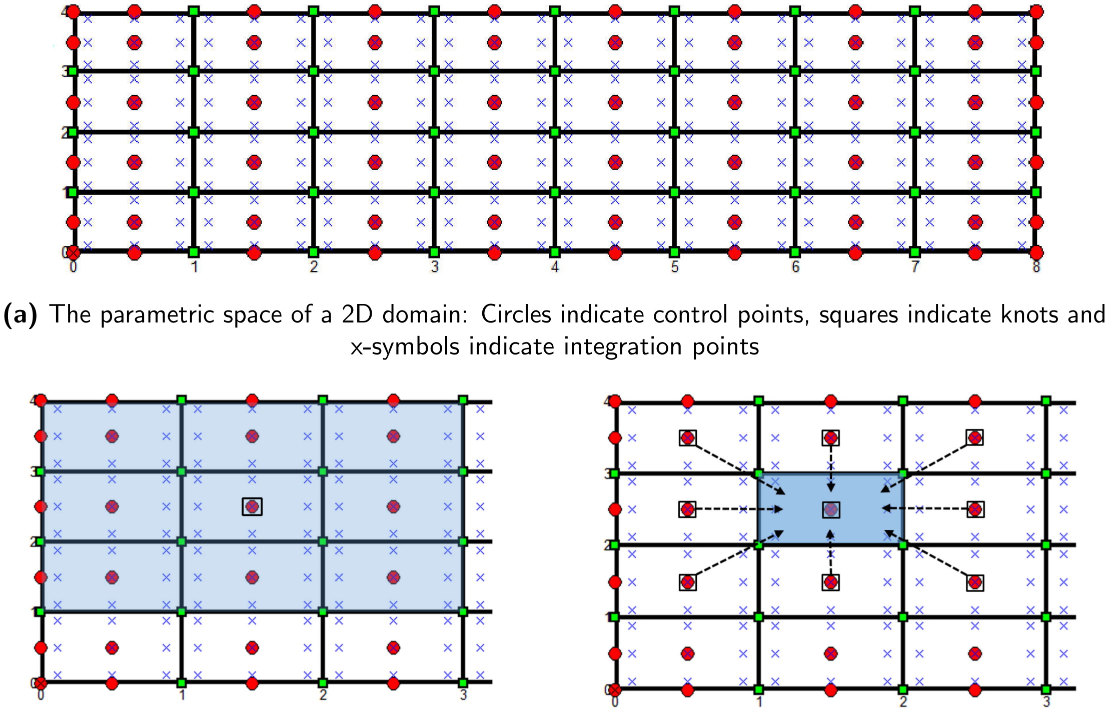
$$\frac{(2p-2)+2}{2} = \frac{2p}{2} = p \tag{4.23}$$

In an analogous fashion, the minimum number of Gauss Points can be derived for 2D and 3D cases. The maximum polynomial degree, needed to accurately compute the shape function partial derivatives used in the deformation matrix, is p. eq.4.22 is a polynomial of maximum degree 2p which is an even number and the minimum number of integration points are computed as:

$$\frac{2p+2}{2} = p+1 \tag{4.24}$$

Figure 4.2a illustrates an example of a 2D domain, modeled with biquadratic shape functions. According to the above discussion, the minimum number of integration points are (p+1)=3 for both parametric directions.

It is also important to notice another property that stems from the definition of B-Spline shape functions; according to [143], each control point is associated to one shape function and their support is equal to (p+1) knot spans. This property is illustrated in Figure 4.2b where the affected area of the indicated control point is shown shaded, while the connectivity of the control point is indicated in Figure 4.2c. Furthermore, each knot span contains (p+1) non-zero shape functions of degree p. In two dimensions, each element that is created from tensor product knot spans, contains $(p+1) \times (q+1)$ shape functions for polynomials degrees p and q that affect an element. Following the same rationale, in three dimensions, $(p+1) \times (q+1) \times (r+1)$ shape functions for polynomials degrees p,q and r affect and element. Inductively, since each Gauss Point belongs to an element, its numerical value is affected by the control points associated with this particular element, as illustrated in Figure 4.3.



(a) The parametric space of a 2D domain: Circles indicate control points, squares indicate knots and x-symbols indicate integration points

(b) Control point - shape function support

(c) Element - control point connectivity

Figure 4.2: A 2D domain modeled with 6x10 control points and biquadratic shape functions (p=2)..

4.4.2 Shape function truncation

Following the presentation of the control point support, a shape function support truncation process will be described. For this purpose, an implementation that truncates the support of shape function along predefined boundaries in the parameter space will be examined.

Let Ω be a one-dimensional NURBS patch, which is required to be split at a certain position ξ_t of parametric axis Ξ. This procedure will remodel that initial domain into two subdomains, namely Ω_{s_1} and Ω_{s_2}. Each subdomain contains a set of shape functions R_{s_1} and R_{s_2} for which:

$$R_{s_1} \cap R_{s_2} \neq 0 \tag{4.25}$$

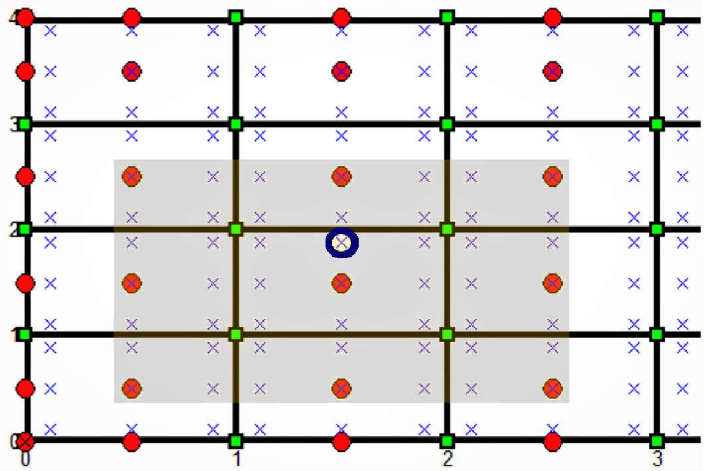


Figure 4.3: 2D domain for p=q=2. N control points (p+1) x (q+1)=9 have non-zero influence to the circled Gauss point.

To these sets of functions, a truncation function $T(\xi_t)$ will be applied which will modify them by reducing the subdomain common support to a single shape function. Since the application of the truncation function will eliminate the contribution of the shape function sets beyond position ξ_t, the truncation functions is chosen to be expressed with the aid of the Heaviside function, which is defined as follows:

$$H(\xi) = \begin{cases} 0, & for\ \xi < 0 \\ 1, & for\ \xi \geq 0 \end{cases} \tag{4.26}$$

As a result, each of the subdomains' shape function sets R_{s_1}, R_{s_2} are truncated as follows:

$$R^*_{s_1} = R_{s_1}[1 - H(\xi - \xi_t)] = R_{s_1} T_L(\xi_t) \tag{4.27a}$$

$$R^*_{s_2} = R_{s_2} H(\xi - \xi_t) = R_{s_2} T_R(\xi_t) \tag{4.27b}$$

An immediate outcome of the above is that, for the subdivision of the initial domain into a multitude of subdomains, each intermediary subdomain can be described as follows:

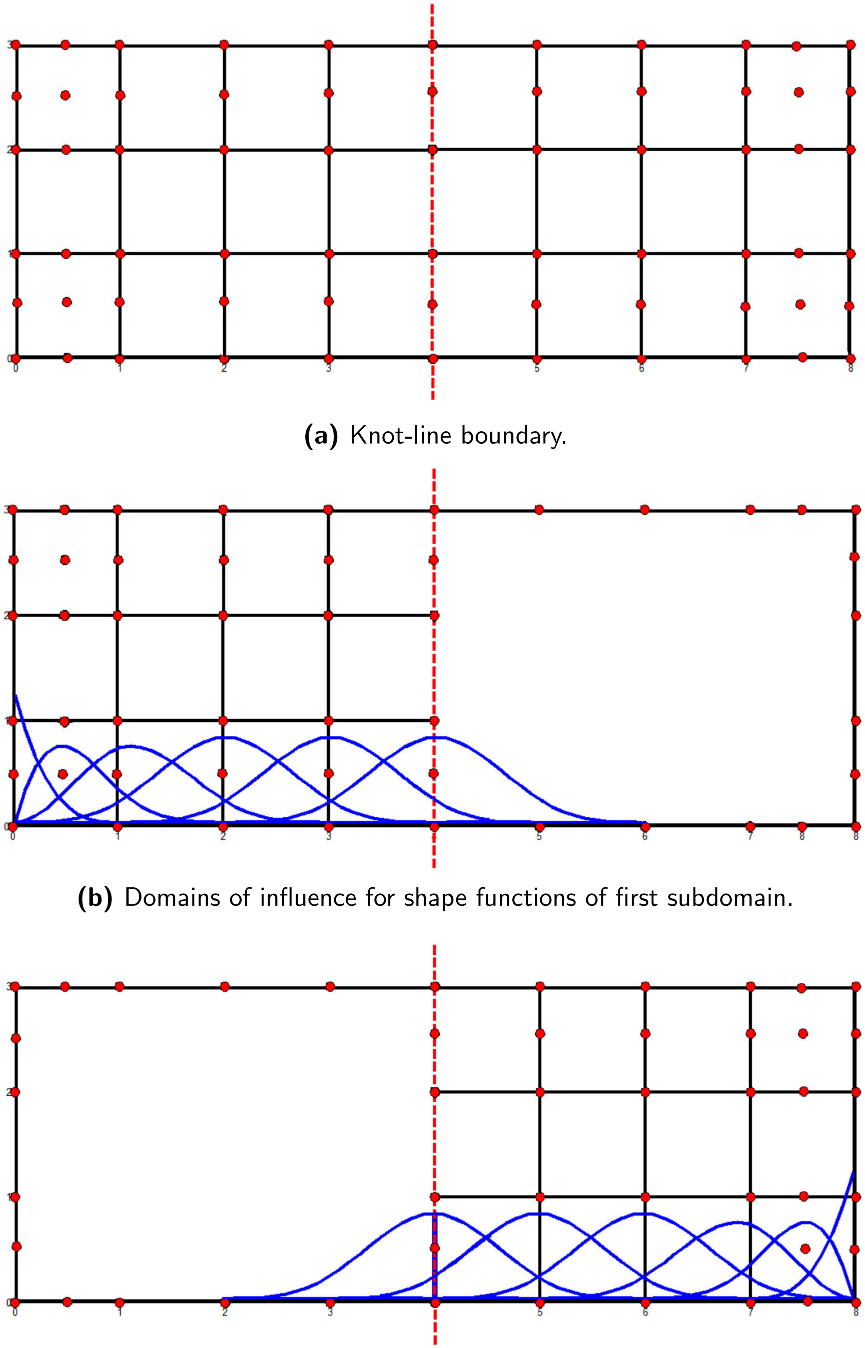
$$R^*_{si} = R_{si} T_L(\xi^a_t) T_R(\xi^b_t) \tag{4.28}$$

with ξ^a_t, ξ^b_t being the equivalent truncation positions. Since $T_L(0)$ and $T_R(1)$ do not affect the contribution of the shape functions, eq. 4.28 can be generalized for the description of the first and last subdomains with knot value vector equal to $\Xi = [\xi_1 = 0, ..., \xi_{n+p+1} = 1]$. Moreover, eq. 4.28 can be expanded for more dimensions with the introduction of similar functions for each of the corresponding parametric directions.

In order to examine the properties of shape functions, both even and odd polynomials degrees are examined for the 2D domain of Figure 4.4a. The subdivision of the domain into two subdomains by the knot line as indicated in Figure 4.4a is considered for p=3, the penetration of the shape functions of the left subdomain to the right one and vice-verca are depicted in Figure 4.4b and 4.4c. The graph of the non-zero tems of its stiffness matrix is demonstrated in Figure 4.5a, while Figure 4.5b depicts the shared dof between the two subdomains before the truncation of the p=3 shape functions.

The parameterization with p=3 corresponds to 6x11 control points. Figure 4.6 shows the domains of influence of the truncated shape functions for the two subdomains of the 2D domain example. The expansion of the shape functions is spatially limited to the knot line, leaving the area beyond the knot line unaffected. Since, for the case of odd polynomial degrees, the control points lie on knot lines where truncation is performed, this results in shared dof between the subdomains, thus retaining the minimum continuity of the domain after the shape function truncation.

For the biqubic polynomial degree case the global stiffness matrix shown in 4.5a, is transformed to the graph of Figure 4.7.



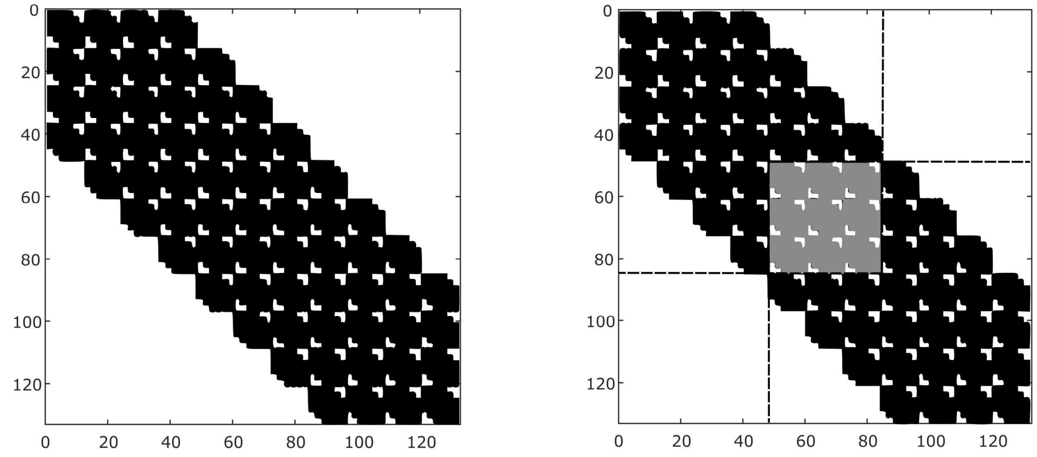
(a) Knot-line boundary.

(b) Domains of influence for shape functions of first subdomain.

(c) Domains of influence for shape functions of second subdomain.

Figure 4.4: 2D domain with bicubic shape functions subdivided into two subdomains by the red dash line. Shape functions depicted are identical across the dashed line due to tensor product property of NURBS.

For the case of even polynomial degrees, the control points lie on the center of elements, as shown in Figure 4.8 for p=2, and the truncated shape functions terminate at the boundary knot line. For this case, the sparsity patterns of the full continuity and reduced continuity stiffness matrices are depicted in Figure 4.9. It can be seen in Figure 4.9b that the two subdomains do not interact since there are no common control points at the boundary knot line.



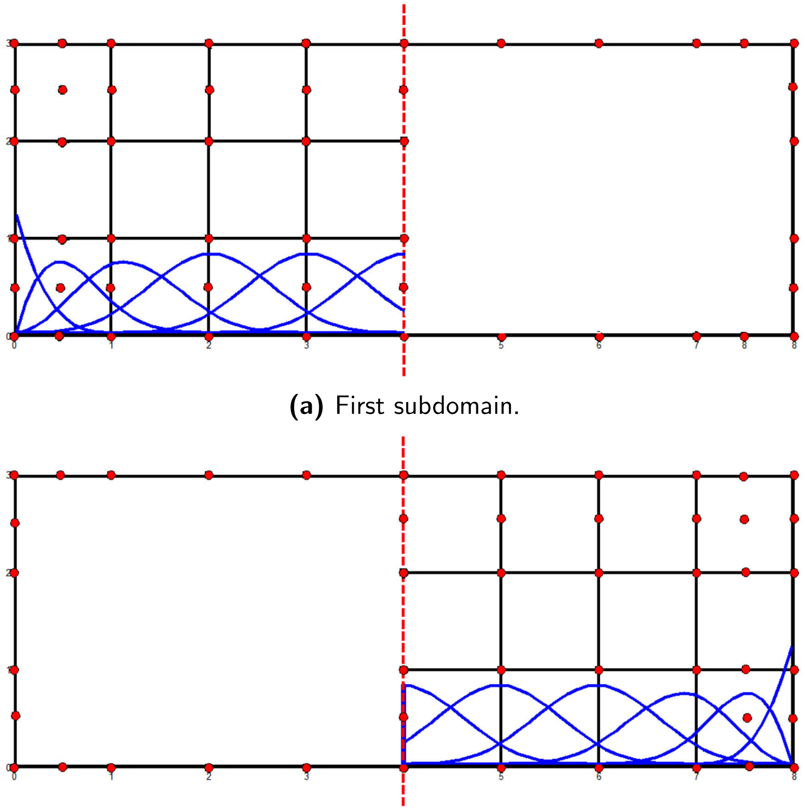
(a) Global stiffness matrix graph of the 2D domain for p=3.

(b) Overlapping boundary terms of the two subdomains.

Figure 4.5: Stiffness matrix of the 2D domain of Figure 4.4.

In order to alleviate this shortcoming, the subdomain boundaries are defined by control point lines, and not by knot point lines, as shown in Figure 4.12. Thus, choosing a control point line as the boundary between two subdomains, the compatibility along the new boundary is restored as shown in Figure 4.10. For an one-dimensional with knot value vector equal to $\Xi = [\xi_0, \xi_1, ..., \xi_{n+p+1}]$, the Greville absissae as described in [161] can be used for the evaluation of the parametric position of the control points which also serve as the possible positions for domain subdivision. Specifically, the parametric coordinates are given by:

$$\xi_i = \frac{\sum_{k=1}^{p} \xi_{i+k}}{p} \tag{4.29}$$



(a) First subdomain.

(b) Second subdomain.

Figure 4.6: Truncated shape functions along the subdomain boundary knot line for bicubic polynomial degree. Knots coincide with controls points on the element vertices.

4.4.3 Benchmark test

In order to evaluate the numerical scalability of the PCG-IETI-N method, two benchmark examples are considered. The first one relates to the numerical scalability of the method, examining the number of iterations related to both the mesh size and the subdomain size, considering a cubic mesh with continuity p=3. The initial guess of the method is the zero vector and iteration count is considered for convergence with tolerance 10^{-6}. The number of iterations required for each case are depicted in Table 4.1.

The second relates to how numerical scalability is influenced by the continuity parameter p in terms of iterations needed for convergence, a parametric study of a 3D cantilever with dof in the range of 3K was considered with shape function polynomial

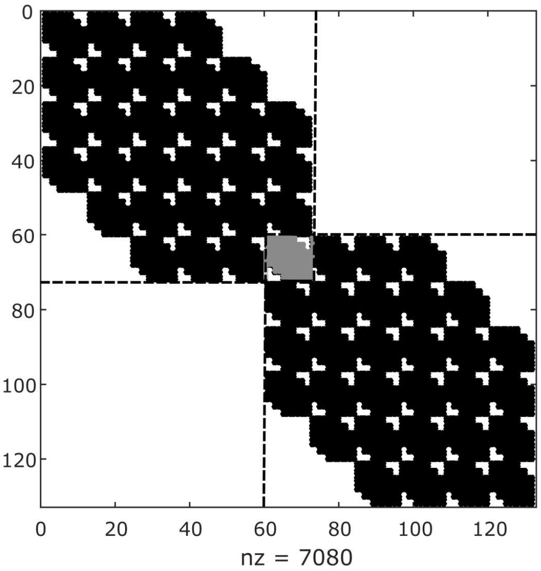
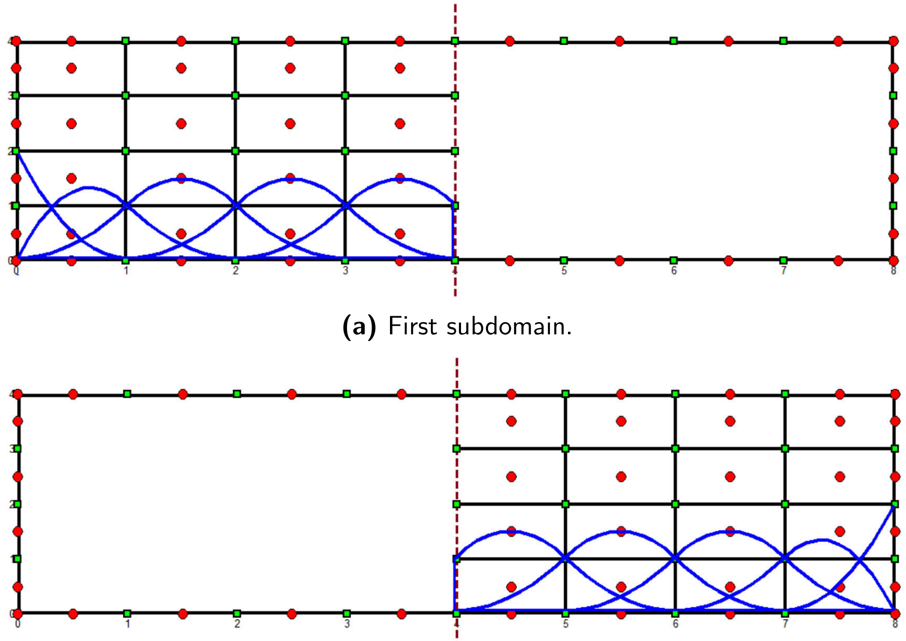


Figure 4.7: Stiffness matrix graph of 2D domain for p=3 with artificially reduced continuity.

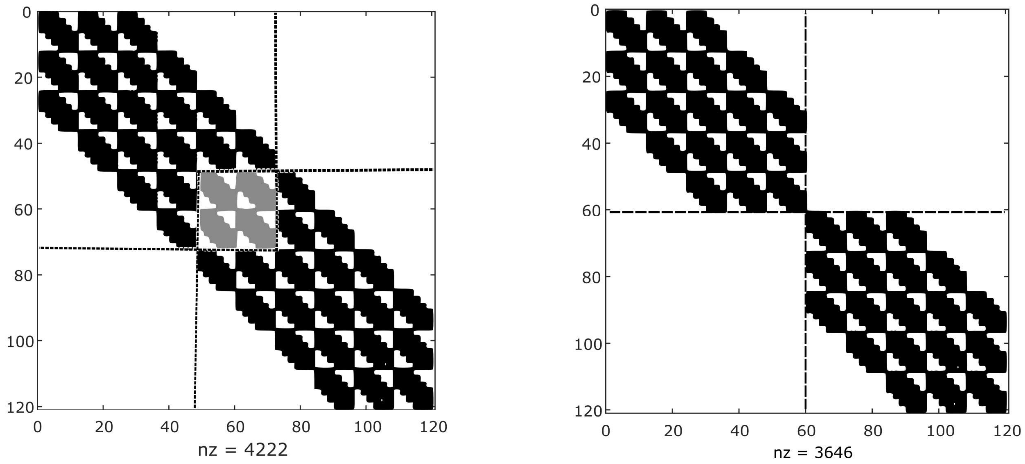
degrees ranging from p=2 to p=5, subdivided into 4 and 16 subdomains. As in the previous test, the initial guess of the method is the zero vector and iteration count is considered for convergence with tolerance 10^{-6}. The number of iterations required and respective condition numbers are depicted in Tables 4.2, 4.3.



(a) First subdomain.

(b) Second subdomain.

Figure 4.8: Truncated shape functions along a knot line for p=2.



(a) Full continuity.

(b) Artificially reduced continuity.

Figure 4.9: Stiffness matrix graphs of a 2D domain for p=2.

N	1/h=8	1/h=16	1/h=32
2×2	10	11	13
4×4	11	11	15
8×8	-	13	17
4×4	-	-	17

Table 4.1: Iteration metrics considering mesh (1/h) and subdomain size (N) for a cubic sample for p=3

p	Unpreconditioned System		Diagonal		Cholesky	
	k2	It.	k2	It.	k2	It.
2	19221	207	22632	192	3902	35
3	25223	234	35824	188	2318	25
4	33449	271	55348	191	2548	21
5	42722	399	77509	234	2081	18

Table 4.2: Iteration and condition number metrics for the case of various PCG preconditioners with p ranging from 2 to 5 for the 3D cantilever example.

The results show a similar behavior of both the condition number and iterations with IETI [27, 43]. This is an indication that the process described has a minor effect on the numerical scalability of the DDM used which for this case is IETI.

The PCG method was also used for the solution of the 3D contilever example for p=2 and p=3 in the range of 10K dof, using the PCG-D and PCG-IC methods which correspond to the diagonal and the incomplete Cholesky factorization preconditioners. The incomplete Cholesky preconditioner by position is implemented with $K = \tilde{L}\tilde{L}^T + E$, in which the sparsity pattern of the lower triangular part of K is retained in $\tilde{L}$ [32]. The iteration required for convergence are shown in Table 4.4 and the convergence history in Figure 4.11.

In both cases, PCG-IETI-N required less iterations than the incomplete Cholesky preconditioner, which is considered a strong precondioner for the examined test case, with the addittional advantage of being amenable to parallelization. In the following section, a set of large IGA models are used for further investigating the performance of the methods discussed.

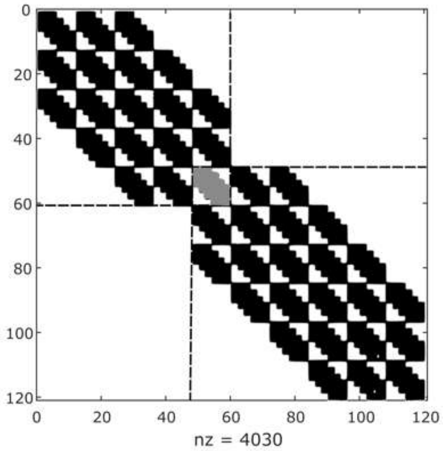
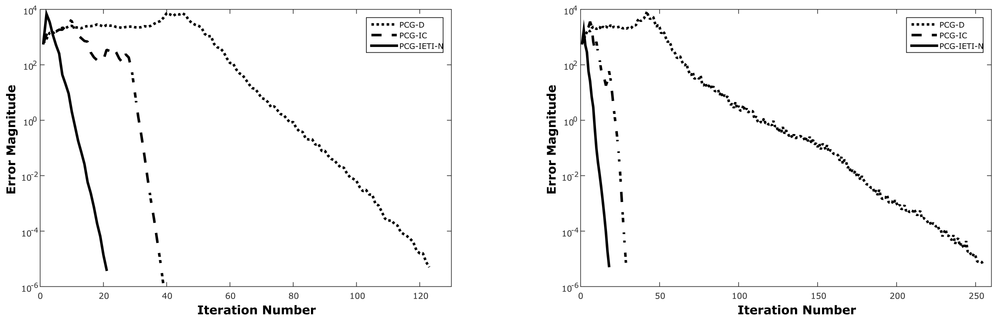


Figure 4.10: Stiffness matrix graph for truncated shape functions along the boundary of the control point line for p=2.



(a) 3D cantilever convergence of PCG schemes for p=2.

(b) 3D cantilever convergence of PCG schemes for p=3.

Figure 4.11: Convergence history of 3D cantilever for various quadratic and cubic polynomial degrees.

p	4 Subdomains Non-Overlapping		16 Subdomains Non-Overlapping	
	k2	It.	k2	It.
2	2011	14	5756	15
3	263	10	804	11
4	3204	24	8615	27
5	4124	19	7717	21

Table 4.3: Iteration and condition number metrics for the case of 4 and 16 subdomains with p ranging from 2 to 5 for the 3D cantilever example.

	p=2	p=3
PCG-D	123	255
PCG-IC	39	29
PCG-IETI-N	21	18

Table 4.4: Iteration metrics PCG-D, PCG-IC and PCG-IETI-N methods for the 3D cantilever example.

4.5 Numerical tests

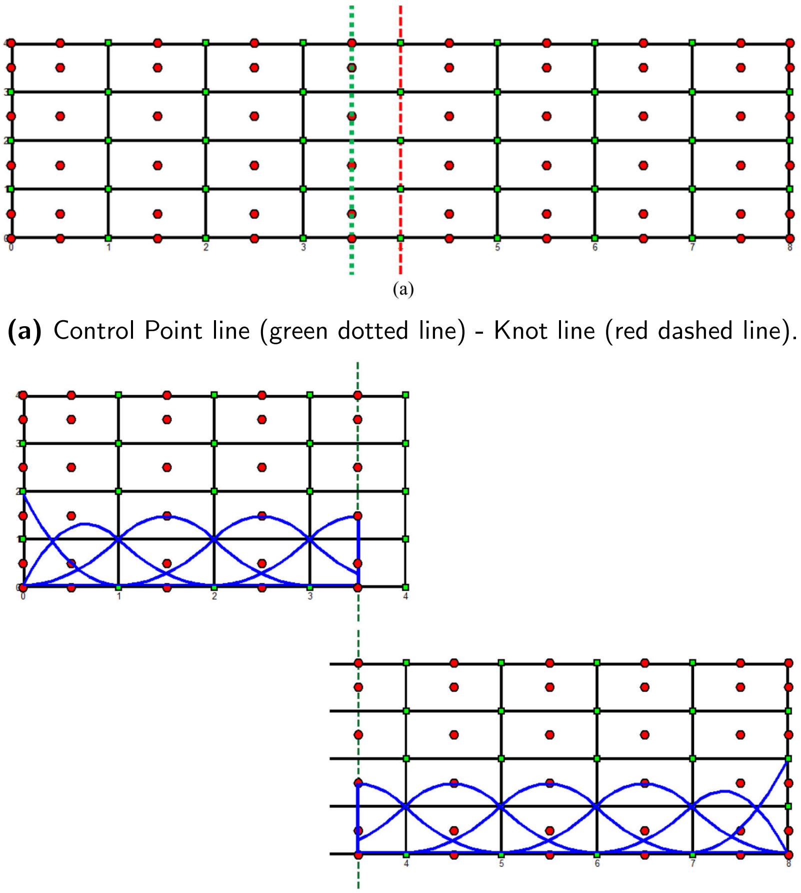
In this section two numerical examples have been chosen to demonstrate the performance of the methods in three dimensional elasticity problems using the in house open-source code *MSolve* developed in the framework of ERC grant Master [MSo].

Model range from 100K to 1000K dof, enabling the assesment of the methods in a large-scale context. The influence of the number of subdomains on the performance of the methods is also considered by performing a parametric study in the range of 9 to 4800 subdomains for the numerical examples considered. For the case of PCG-IETI-N, these subdomains are defined by constructing a non-overlapping model with truncated shape functions as described in the presious section. For the case of IETI, both IETI-P and IETI-O are considered. The computing platform is an Intel Core i7 X980 with 6 cores.

4.5.1 Crossed beams example

The first model is a crossed beams model comprised of approximately 100K dofs and is depicted in Figure 4.13

All control points of the bottom and left faces of the crossed beams are clamped



(a) Control Point line (green dotted line) - Knot line (red dashed line).

(b) Subdomain subdivisioning at a control point line

Figure 4.12: Truncated shape functions along a control point for p=2.

constraining the base displacements. Loads of magnitude 100 KN, are applied to interpolatory control points of the two edges as illustrated in the figure. The material used for the linear elastic analysis has a Young's modulus E of 10^5 KPa and Poisson's ration of 0.3.

Tables 4.5, 4.6 give the convergence behavior of the methods with respect to the number of iterations, the size of the subdomains and the equivalent interface problem. For the case of PCG-IETI-N, both preconditioning step iterations and the global iterations are depicted.

The computing time measured in seconds is provided in Table 4.7 for both sequential and parallel execution, where for the latter, all 6 cores were utilized. As for the case of the pipe model, the PCG-IETI-N outperforms all other methods considered in terms of

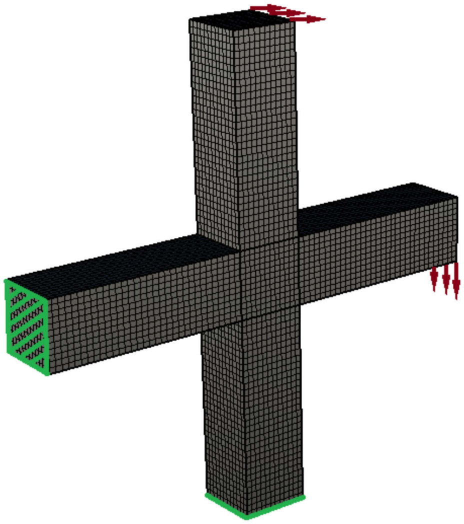


Figure 4.13: Crossed beams with 32.175 control points.

execution time in both sequential and parallel execution modes.

Figure 4.14 demonstrates the parallel execution time for all methods where there is an optimal combination of the size of the local problems and the size of the interface problem. It can be observed, however, that the performance of the PCG-IETI-N is less dependent on the number of subdomains.

Subdomains		IETI-P		IETI-O	
Number	Size(dof)	Iter	Interface(dof)	Iter.	Interface(dof)
9=9x1x1	11925	14	5220	15	20620
36=18x2x1	4185	15	19665	17	42345
104=26x2x2	1963	18	37323	20	63549

Table 4.5: Iterations subdomain metrics of IETI-P and IETI-O methods for the crossed beams model.

4.5.2 Pipe example

In order to assess the performance of the methods in a a larger-scale context, a second model is considered as shown in Figure 4.15

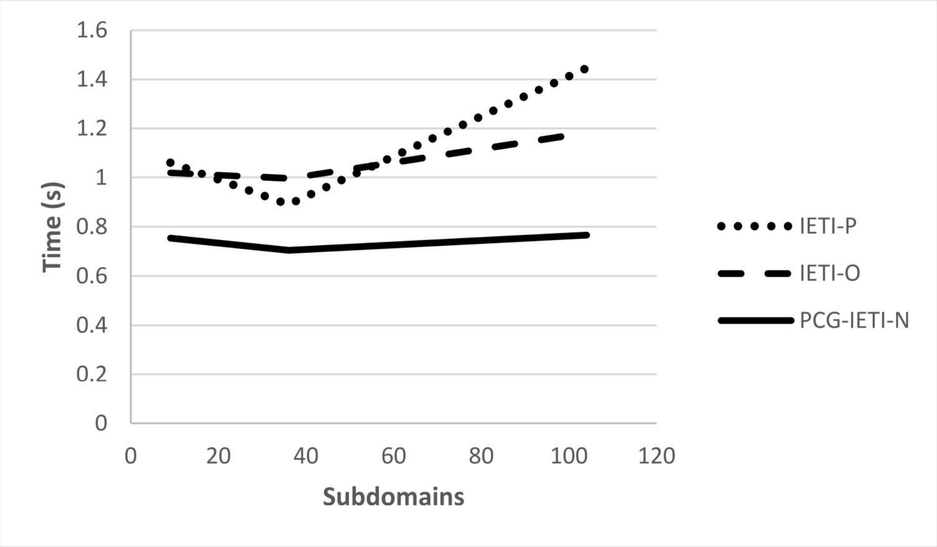


Figure 4.14: Performance of the various IETI methods in terms of parallel execution time for the crossed beams model.

Subdomains		PCG-IETI-N		
Number	Size(dof)	Iter	Prec. iter.	Interface(dof)
9=9x1x1	11925	14	10	7920
36=18x2x1	4185	17	12	30240
104=26x2x2	1963	20	14	47190

Table 4.6: Iterations subdomain metrics of the PCG-IETI-N method for the crossed beams model.

The pipe model was created with three dimensional NURBS using the NURBS Toolbox [M] and consist of 57 control points per parametric axis ξ that creates the circumference of the circle, 9 per parametric axis η that creates the radius of the annulus and 306 per parametric axis ζ that gives height to the pipe. The degree of NURBS shape functions is considered to be consistent in all parametric directions and is equal to p=2. The base of the pipe is clamped by constraining the displacements of the corresponding control points. Concentrated loads are at the control points, interpolatory to the geometry on the top of the pipe with magnitude of 100 KN, introducing both bending and torsional strain to the structure. The material used for the linear elastic analysis is identical with the previous model and has a Young's Modulus E of 10^5 KPa and Poisson's ratio 0.3.

Tables 4.8, 4.9 give the performance of the solution methods considered for the pipe model of approximately 500K dof in terms of number of iterations, equivalent subdomain

Subdomains	IETI-P		IETI-O		PCG-IETI-N	
	Time	Time	Time	Time	Time	Time
	seq.	par.	seq.	par.	seq.	par.
9	5.677	1.061	5.297	1.019	4.072	0.754
36	4.989	0.891	5.625	0.996	3.883	0.705
104	7.729	1.445	6.14	1.181	4.137	0.766

Table 4.7: Performance metrics of the various IETI methods for the crossed beam model.

size and interface problem. For the case of PCG-IETI-N, the preconditioned iterations are also displayed which correspond to the toal number of iterations needed for the estimation of the preconditioned residual.

Subdomains		IETI-P		IETI-O	
Number	Size(dof)	Iter	Interface(dof)	Iter.	Interface(dof)
100=25x4x1	5670	15	61399	16	88939
100=25x4x1	847	16	244135	17	306211
2400=75x16x2	519	18	338004	19	368875

Table 4.8: Iterations subdomain metrics of IETI-P and IETI-O methods for the crossed beams model.

Subdomains		PCG-IETI-N		
Number	Size(dof)	Iter	Prec. iter.	Interface(dof)
100=25x4x1	5670	15	10	67069
100=25x4x1	847	17	10	195751
2400=75x16x2	519	20	12	244183

Table 4.9: Iterations subdomain metrics of the PCG-IETI-N method for the crossed beams model.

Table 4.10 presents the performance of the method in terms of the required computing time in sequential as well as in parallel computing environments. It is evident by this table that the PCG-IETI-N outperforms all other methods considered in terms of execution time in both sequential and parallel implementations. Execution time is measured in seconds, for both sequential and parallel execution, where for the latter, all 6 cores were utilized. Moreover, while the numerical scalability of PCG-IETI-N is in line with the numerical scalability of the other two methods, it outperforms them in terms of parallel scalabilty. Figure 4.16 depicts the parallel execution time for all methods.

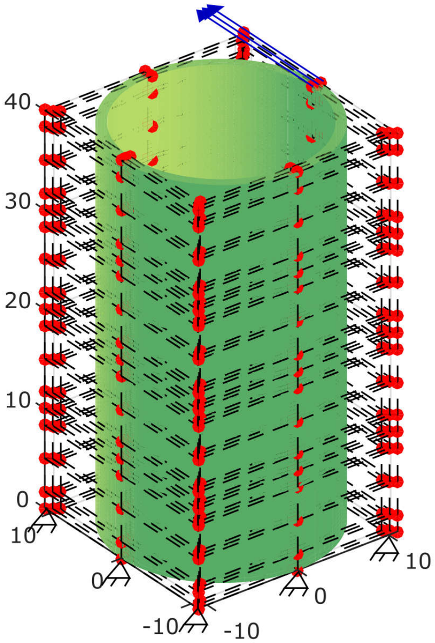


Figure 4.15: Pipe cylinder with coarse discretization.

The curves depict the usual behavior of wall clock time when dealing with DDM, where there is an optimal balance between the size of the local problems and the size of the interface problem.

Subdomains	IETI-P		IETI-O		PCG-IETI-N	
	Time seq.	Time par.	Time seq.	Time par.	Time seq.	Time par.
100	37.25	6.77	36.91	6.71	26.84	4.97
1200	34.92	6.24	34.65	6.42	21.23	3.93
2400	50.71	9.48	35.94	6.78	23.27	4.22

Table 4.10: Performance metrics of the various IETI methods for the pipe model of 500K dof.

The execution time of the various IETI methods on a paralle environment, with respect to the number of subdomains, are graphically represented in Figure 4.16. It is worth noting that there is a significant increase in execution time of IETI-P and to a lesser extends to IETI-O with respect to the number of subdomains. This is attributed in both cases to larger interface problem to be solved by the IETI algorithm.

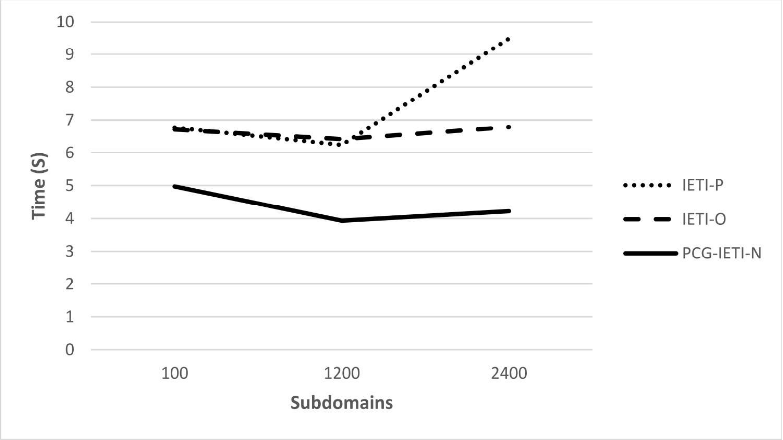


Figure 4.16: Parallel performance of the various IETI methods in terms of the number of subdomains for the pipe model.

To evaluate further the numerical scalability properties of the proposed solution schemes for even larger models, a finer discretization of the previous model is performed, reaching a total 1M dof and 4800 subdomains. Table 4.11 provides a comparison between the computational times of the previous model for the cases of 500K and 1M dof, considering the maximum number of subdomains for each case.

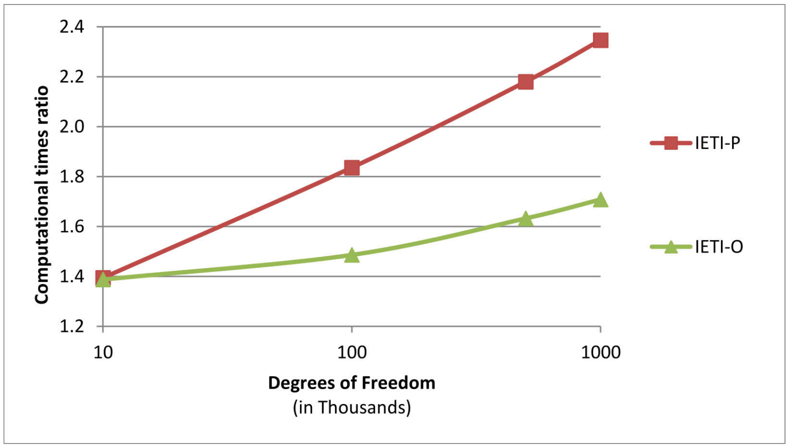


Figure 4.17: Computational improvement of PCG-IETI-N with respect to IETI-O and IETI-P for different model sizes.

Size(dof)	Subdomains	IETI-P		IETI-O		PCG-IETI-N	
		Time seq.	Time par.	Time seq.	Time par.	Time seq.	Time par.
500 K	2400	50.71	9.48	35.94	6.78	23.27	4.22
1 M	4800	116.01	21.82	84.52	16.51	49.48	9.06

Table 4.11: Performance metrics of the various IETI methods for 500K and 1M pipe models.

Figure 4.17 presents a graphical representation of the speedup ratios of the numerical results for the PCG-IETI-N compared to the IETI-P and IETI-O respectively. The ascending trend in the computational efficiency of the PCG-IETI-N when compared to IETI-P and IETI-O becomes more evident in thsi figure. As the models get larger in scale the computational predominance of the proposed solution scheme becomes more pronounced, providing a speedup of more than 2.3x and 1.7x, compared to IETI-P and IETI-O, respectively.

5

Domain Decomposition Solution Schemes for Isogeometric Collocation Methods

5.1 Introduction

This Chapter focuses on the development of a family of solution algorithms for the purposes of isogeometric collocation methods, that exploit the advantages of an iterative solution scheme, name the GMRES coupled with domain decomposition methods. By introducing an appropriate decomposition of the NURBS overlapping nature, non-overlapping decomposition of the initial domain is performed, that allows the utilization of established DDM methods. Both primal and dual alternatives are examined for the preconditioner of GMRES method, with P-FETI-DP being the most suitable one to serve as a robust and scalable preconditioner.

5.2 Isogeometric methods

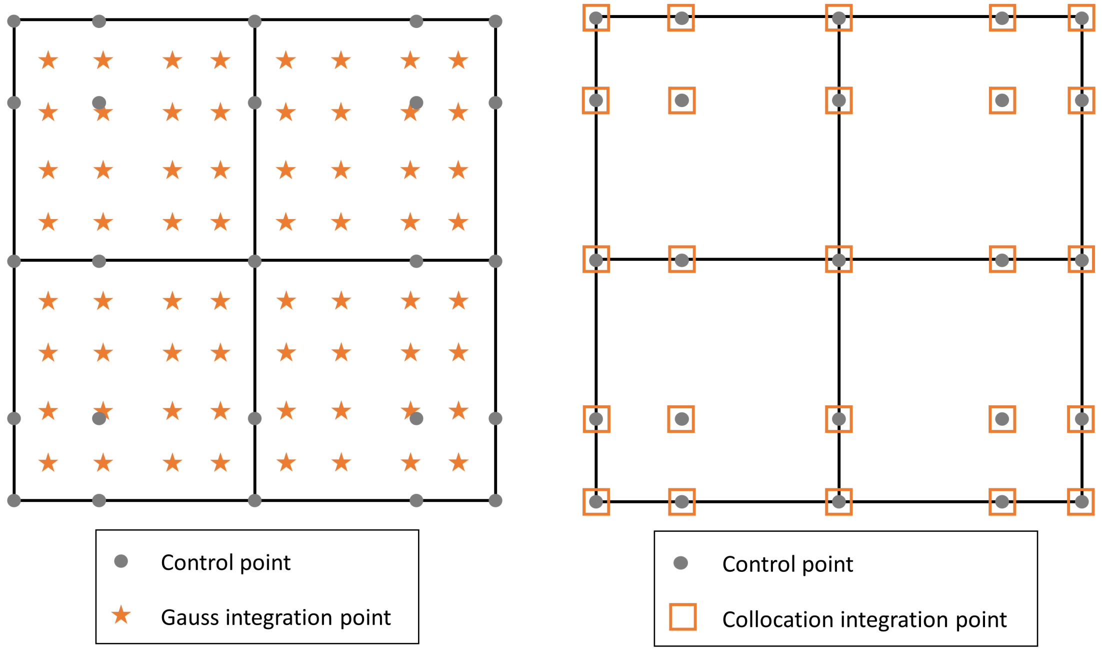
5.2.1 Galerkin vs Collocation

The analysis of linear elasticity PDEs with the isogeometric Galerkin method has been proven to provide results with significantly increased accuracy per degree of freedom compared to its FEM counterparts. This is mainly attributed to the increased continuity of the shape functions, that consequently lead to minimized errors and increased smoothness of second order characteristics such as stresses and strains. Unfortunately, this comes at an increased cost for the assembly of the resulting matrices. As a result, a number of studies has been conducted for implementation of alternative more efficient integration schemes, such as the isogeometric collocation method [19, 13]. A comparison between isogeometric Galerkin and collocation methods was conducted in [161] where collocation was proven to be significantly more efficient than the corresponding Galerkin implementation in terms of the computation of the resulting stiffness matrix.

Specifically, for each collocation point, the cost of its contribution to the stiffness matrix is of magnitude $O(p^d)$, where p is the polynomial degree and d the dimensionality of the problem. This cost is mainly attributed to the requirement for evaluation of shape functions second derivatives at each collocation integration point, due to the increased continuity requirements of the collocation discretizations. On the contrary, Galerkin method requires $O(p^{2d})$ computational effort for the stiffness contribution at each Gauss integration point. Apart from the reduced cost per collocation point, collocation method requires the evaluation of only a single collocation point per shape function, while fully integrated isogeometric Galerkin methods require $(p+1)^d$ Gauss point evaluations per isogeometric element, thus dramatically increasing the stiffness matrix computation cost.

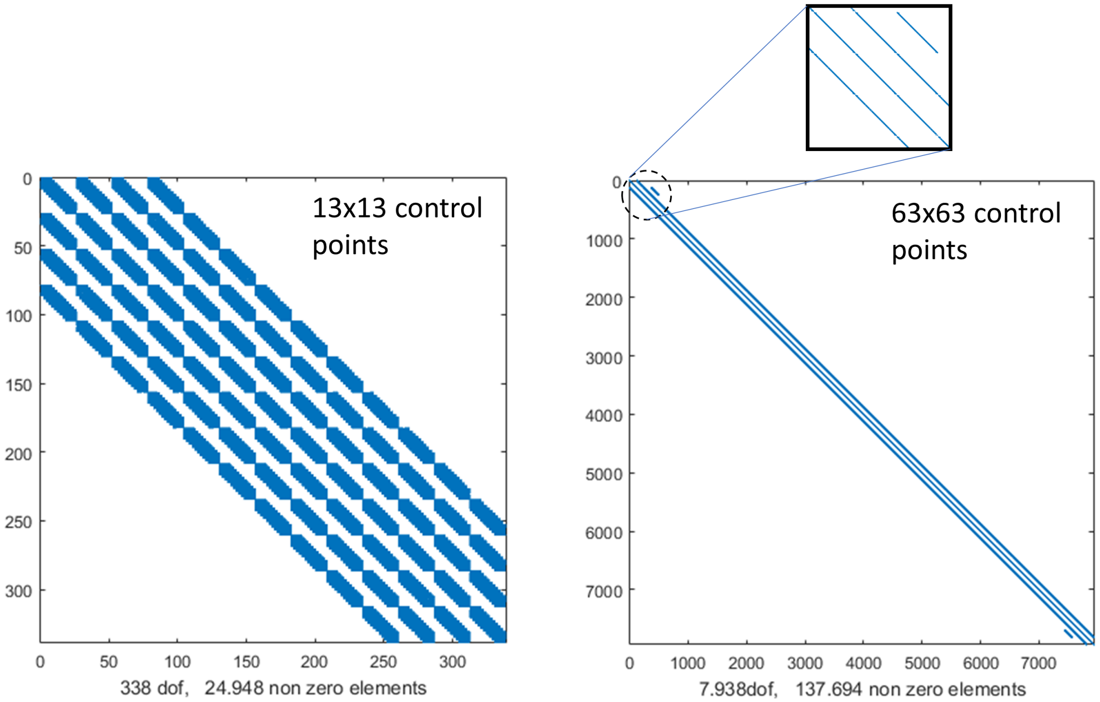
An example of the number of Gauss and collocation points needed for the computation of the stiffness matrix, in each case respectively, is illustrated in Figure 5.1. Namely, for the bicubic domain depicted in Fig. 5.1a of the IGA Galerkin method, a multitude of 16 Gauss Points per isogeometric element are needed leading to a total of 64 Gauss Points for all four isogeometric elements. On the contrary, the same domain discretized with the isogeometric collocation method (Fig. 5.1b) requires only 25 collocation integration points, thus significantly reducing the computational burden for the assembly. However, the accuracy deficit of the isogeometric collocation method can be more than two orders of magnitude, for a minimum polynomial degree p=3 and grows more profound as the polynomial degree rises.

Despite this markedly advantage of collocation over Galerkin IGA methods, they have shortcomings that need to be addressed. In order for the collocation methods to match the accuracy of the Galerkin methods, they have to resort to excessively refined meshes or to higher polynomial degrees that exponentially the number of dof, thus leading to larger non-symmetric matrices. Furthermore, the condition number of the collocation stiffness matrices grows rapidly in case of mesh refinement or polynomial degree elevation [25]. A comparison of the resulting stiffness matrices for Galerkin and collocation for a square domain for the same accuracy is depicted in Figure 5.2. In case of Galerkin, the domain is discretized with 13 control points per parametric axis. In order to attain the same level of accuracy, the domain must be discretized with 63x63 control points in case of collocation discretization. This leads to 338 dof in the Galerkin case and 7938 dof in the collocation case, indicating the increased computational effort required for the solution of the equivalent collocation systems. This deficiency makes imperative the introduction of efficient and robust solution techniques for addressing the problem of solving isogeometric non-symmetric linear systems associated with the collocation methods.



(a) Galerkin **(b)** Collocation

Figure 5.1: Integration points needed for the computation of the stiffness matrix of 2x2 isogeometric Galerkin and collocation discretizations.



(a) Galerkin

(b) Collocation

Figure 5.2: Stiffness matrix sparsity patterns for isogeometric Galerkin and collocation discretizations for accuracy 10^{-2}.

5.3 Existing solution techniques

The non-symmetric nature of the resulting matrices with high condition numbers, creates the need for efficient computational methods to address the solution of complex and large-scale problems. The resulting linear systems of isogeometric collocation, described in the previous section, can be represented in a matrix form as

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{F} \tag{5.1}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{F}$ is the external load vector and $\mathbf{u}$ the displacement vector. The computational effort needed for the solution of eq. (5.1), depends highly on the size, the ellipticity and the sparsity pattern of the stiffness matrix. Specifically,the number of non-zero entries per row, for a vector field with k unknown quantities per node, are $O(k \cdot p^d)$ [161], for both isogeometric collocation and finite elements, while for isogeometric Galerkin is $O((k \cdot 2p)^d)$, where p is the shape function polynomial degree and d the dimensionality of the problem. In case of IGA, the CAD shape functions utilized for the exact geometrical representation of the geometry, leads to a reduced sparsity patterns due to their increased interelement continuity, while high polynomial degrees increase the expected bandwidth of the matrices. All the above, in conjunction to the non-symmetric nature of matrices, lead to a computationally demanding solution procedure.

This deficit of isogeometric collocation, significantly hinders its capability to address complex large-scale problems, unless efficient solution techniques capable of alleviating the solution computational cost are proposed. In this context, multi-frontal parallel direct solver was proposed in [116], for the solution of one-dimensional nanolithography collocation problem. Optimal multilevel preconditioners for isogeometric collocation methods were introduced in [48], where a Bramble-Pasciak-Xu and multigrid V-cycle accelerated GMRES is proposed for the solution of collocation resulted systems of equations. The use of the overlapping Schwarz preconditioners in [25] is the first attempts to introduce domain decomposition methods in isogeometric collocation. It is based on an algebraic decomposition of the stiffness matrix $\mathbf{K}$ into subdomain matrices that share multiple rows of control points per parametric direction. At the same time, a coarse problem is introduced based on an p-th degree interpolation, where p is the polynomial degree of the shape functions corresponding to selected collocation points of the domain.

5.3.1 Direct Solvers

The most widely applied methods for the solution of non-symmetric problems are direct solvers. They are based on a factorization of the stiffness matrix, with the most common factorization methods being the LU factorization [145].

$$\mathbf{K} = \mathbf{L} \cdot \mathbf{U} \tag{5.2}$$

where $\mathbf{L}$ and $\mathbf{U}$ are a lower and an upper triangular matrix. After factorization of the stiffness matrix, the solution is obtained with a forward and backward substitution with the aid of $\mathbf{L}$ and $\mathbf{U}$ matrices, with minimal cost. This renders LU factorization ideal in case of problems with multiple right hands sides, yet factorization cost and memory requirements of the method are its main limitations. Despite, the reduced bandwidth form of isogeometric collocation matrices compared to Galerkin methods, its non-symmetric nature significantly increases the required memory allocation. This shortcoming, in connection to the parallelization difficulties involved in fine grain parallel computer implementations, renders direct solvers not efficient for the solution of large scale problems. A parallel implementation of a direct multi-frontal solver for the univariate case is proposed in [115].

5.3.2 Iterative Solvers

One of most widespread iterative methods for non-symmetric linear systems is the GMRES method [155]. In line with all iterative methods, the convergence of the GMRES method can be significantly improved by introducing an appropriate preconditioner. Preconditioners should abide by the rule to enhance the approximation of the solution at each iteration step with the minimum computational cost. The main focus of all proposed collocation iterative solution schemes is the introduction of an efficient preconditioner for accelerating the convergence of the GMRES method. While its algorithmic description is considered standard textbook material, it is provided below since it will be utilized in this work as solver of the global non-symmetric equation system.

5.3.3 Overlapping Schwarz preconditioner

A brief introduction to the overlapping Schwarz method will be provided based on [25], as it will serve as a comparison basis for the method proposed in the next section.

Specifically, in the univariate case, given a knot value vector $\xi = \{\xi_1 = 0, ..., \xi_{n+p+1} = 1\}$, where n is the number of control points and p is the polynomial degree, a subset of knot values ξ_i^s are selected, that partition the domain into similar span intervals. This subset can be written in a vector as

$$\Xi^{interface} = \{\xi_1^s = 0, ..., \xi_i^s, ..., \xi_{N+1}^s = 1\} \tag{5.3}$$

As apparent by eq. (5.3), the selection of these knot values, splits the univariate domain into N consecutive spans $S_i = \{\xi_k^s, \xi_{k+1}^s\}$, whose union is the initial reference interval. Note that the initial and final values of the domain ξ_1^s, ξ_{N+1}^s are the boundary of the reference interval, while all internal values define the interface among adjacent subdomains.

Algorithm 4: GMRES preconditioned algorithm

```
**Data:** Matrix $\mathbf{K}$
          Right hand side $\mathbf{f}$
          Initial solution guess $\mathbf{u}_0$
$\mathbf{r}_0 = \mathbf{K}_{prec}^{-1}(\mathbf{f} - \mathbf{K}\mathbf{u}_0)$;
$\beta := \|\mathbf{r}_0\|_2$;
$\mathbf{v}_1 := \frac{\mathbf{r}_0}{\beta}$ ;
for j=1,2,..,m do
    $\mathbf{w}_j := \mathbf{K}_{prec}^{-1}\mathbf{K}\mathbf{v}_j$;
    for i=1,...,j do
        $h_{i,j} := (\mathbf{w}_j, \mathbf{v}_i)$;
        $\mathbf{w}_j := \mathbf{w}_j - h_{i,j}\mathbf{v}_i$;
    end
    $h_{j+1,j} = \|\mathbf{w}_j\|_2$ if $h_{j+1,j} = 0$ then
        m:=j ;
        exit all loops;
    end
    $\mathbf{v}_{j+1}\frac{\mathbf{w}_j}{h_{j+1,j}}$;
end
Compute the Hessenberg matrix $\overline{\mathbf{H}}_m$;
$\mathbf{y}_m = min(\|\beta e_1 - \overline{\mathbf{H}}_m\mathbf{y}\|_2)$ ;
$\mathbf{u}_m = \mathbf{u}_0 + \mathbf{V}_m\mathbf{y}_m$
```

For each one of the interface knot values ξ_k^s between two adjacent subdomains, a shape function n_k is selected, whose support intersects both subdomains.

Since the decomposition needs to be overlapping, a non-negative integer value m defines the shared shape functions between the subdomains such that

$$A_k = \{n_k - m \leq i \leq n_{k+1} + m\}, \qquad k = 1, 2, ..N \tag{5.4}$$

Note that eq. (5.4) has two boundary cases for the initial and final subdomain, since the integer i of eq. (5.4) takes values that belong within the multitude of shape function of the univariate reference domain, i.e. $i \in \{1, ..., n\}$. As eq. (5.4) implies, $2m + 1$ shape functions are shared between adjacent subdomain, thus the minimum number is given for $m = 0$ and is equal to 1 shared shape function. The extention to multiple dimensions is derived in a straightforward manner by employing tensor products. Specifically, in the two-dimensional case, the subdomains are defined as

$$S_i = \{\xi^s_k, \xi^s_{k+1}\} \tag{5.5a}$$

$$S_j = \{\eta^s_l, \eta^s_{l+1}\} \tag{5.5b}$$

$$S_{ij} = S_i \times S_j \tag{5.5c}$$

In a similar fashion the function indices that affect each subdomain in the bivariate case can be derived, by extending eq. (5.4) as

$$A_{kl} = \{n_k - m \leq i \leq n_{k+1} + m,\ s_l - m \leq j \leq s_{l+1} + m\} \tag{5.6}$$

where $k \in \{1, 2, ..., N\}$ and $l \in \{1, 2, ..., M\}$. The extension to the three-dimensional case can be produced is a similar manner.

Based on eq. (5.6) and taking into account the degrees of freedom corresponding at each control point, a mapping matrix $\mathbf{L}^s_i$ can be defined. This mapping matrix provides the connection between the global degrees of freedom and those included only in the overlapping subdomain partitioning, as defined by eq. (5.5). It is an unsigned Boolean matrix whose rows denote the degrees of freedom of the subdomain, while the columns represent the degrees of freedom of the whole domain. By applying these mapping matrices on the global stiffness matrix $\mathbf{K}$ of eq. (5.1), the matrices of each of the overlapping subdomains can be derived as follows:

$$\mathbf{K}_{kl} = {\mathbf{L}^s_{kl}}^T \mathbf{K} \mathbf{L}^s_{kl} \tag{5.7}$$

Based on eq. (5.7) suffices for the extraction of the one-level overlapping additive Schwarz preconditioner is provided by:

$$\mathbf{K}^{(1)}_{prec} = \sum_{k=1}^{N} \sum_{l=1}^{M} \mathbf{L}^s_{kl} {\mathbf{K}_{kl}}^{-1} {\mathbf{L}^s_{kl}}^T \tag{5.8}$$

The preconditioner of eq. (5.8) can be further enhanced by introducing a coarse problem to it. In an analogy to the domain subdivision, the extraction of the coarse problem for the overlapping Schwarz preconditioner will be performed in a univariate domain which will then be expanded to higher dimensions. As before, a selection of knot values is performed from the knot value vector of the reference domain, in a way that the distance between consecutive distinct knot values are at a distance equal to the span of the subdomains. This leads to the generation of a coarse knot vector of the form

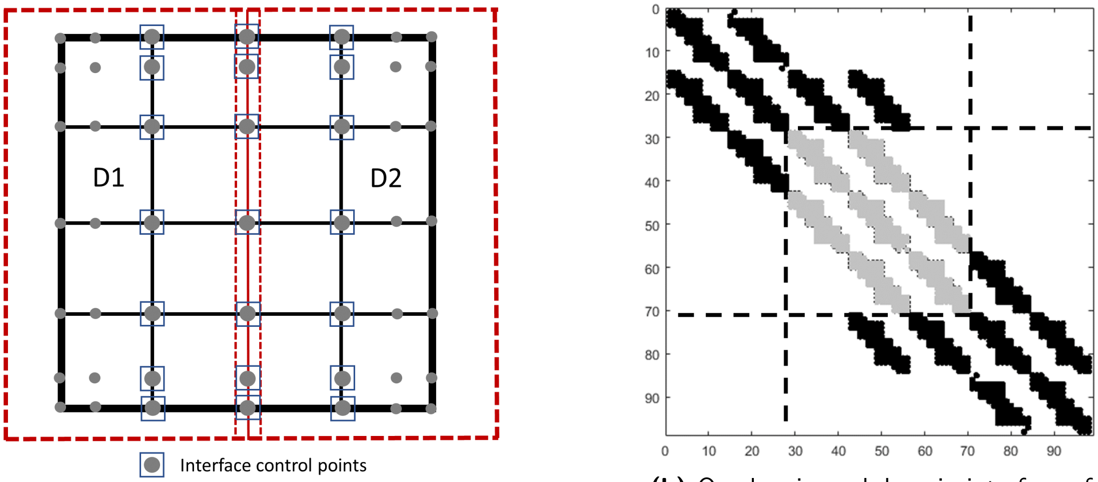
$$\Xi^c = \{\xi_1^c, ..., \xi_{N_c+p+1}^c\} \tag{5.9}$$

Using the tensor product property, eq. (5.9) is extended to determine the multidimensional domain that will serve as the coarse problem of the overlapping Schwarz method. According to [25], multiple procedures can be followed for the formation of the coarse space matrices, yet the one followed here is by utilizing the interpolation operator $\mathbf{L}_0$ from the coarse space to the reference space of the structure. As a result, the coarse space matrix is derived as follows:

$$\mathbf{K}_0 = \mathbf{L}_0{}^T \mathbf{K} \mathbf{L}_0 \tag{5.10}$$

thus extending the preconditioner of eq. (5.8) to

$$\mathbf{K}_{prec}^{(2)} = \mathbf{R}_0 \mathbf{K}_0{}^{-1} \mathbf{L}_0^T + \sum_{k=1}^{N} \sum_{l=1}^{M} \mathbf{L}_{kl}^s \mathbf{K}_{kl}{}^{-1} \mathbf{L}_{kl}^s{}^T \tag{5.11}$$



(a) Overlapping interface

(b) Overlapping subdomain interface of the stiffness matrix

Figure 5.3: Overlapping Additive Schwarz 2x1 domain partitioning of a 4x4 collocation discretized domain

Figure 5.3, provides an example of a 2x1 domain partitioning of a 4x4 discretized domain for the overlapping additive Schwarz method. According to [25], $2m+1$ shape functions are shared among adjacent subdomains, with $m=0$ being the non-overlapping case. In case of $m=1$, then $2m+1=3$ shape functions are shared at the common subdomain interface, as depicted in Figure 5.3a. Specifically, the domain is partitioned in middle of axis X, as indicated by the dashed line. Since, three overlapping control point columns are required, apart from the column defined by the dashed line, the two adjacent columns form the interface problem. The control points that belong to the interface problem denoted with circles in Figure 5.3a. This translates to a major part of the final stiffness matrix being shared among subdomains, with this situation deteriorating as the overlapping index m grows. The effect of the common interface on the stiffness matrix is depicted in Figure 5.3b.

5.4 GMRES and Non-overlapping Domain Decomposition Methods

Despite its favorite features, overlapping additive Schwarz (OAS) method has some drawbacks that decrease its efficiency, with the most notable one the size of the overlapping submatrices, as well as the definition of the underlying coarse problem. Specifically, OAS relies on an overlapping decomposition of the initial domain, fact that leads to a significantly increased computational cost for the solution of the equivalent subdomain problems in each preconditioning step. These interface problems can become prohibitively large for increased polynomial degrees as interelement continuity increases.

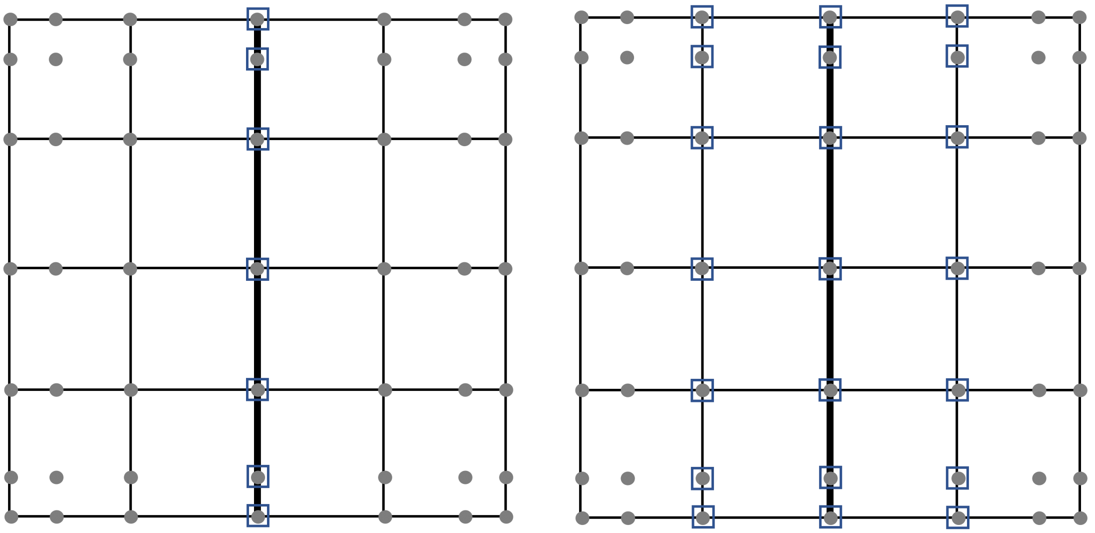
In addition, based on its definition, the extraction of the coarse problem matrices relies heavily on an interpolation operator between the reference and the coarse problem domains. In [25] various polynomial degrees for the interpolation between the coarse and reference domain were examined, from linear interpolation to p^{th} degree interpolation, p being the polynomial degree of the shape functions of the reference domain. In terms of performance, the p^{th} degree interpolation case was proven to be the most effective and thus it chosen for thee evaluation of the interpolation operator in the numerical comparison performed in this work. Note that, in contrast to $\mathbf{L}_{kl}^{s}$ operators that perform only a mapping, $\mathbf{L}_0$ interpolation operator requires a matrix vector multiplication for coarse problem contributions, adding an overhead to the solution procedure.

In order to alleviate the aforementioned deficiencies of the OAS method, an alternative solution methodology is proposed that combines both GMRES and non-overlapping primal and dual domain decomposition methods. In addition to the original model with stiffness matrix $\mathbf{K}$, the proposed algorithm introduces another stiffness matrix $\mathbf{K}_p$, referred to as the "non-overlapping matrix". Matrices $\mathbf{K}$ and $\mathbf{K}_p$ have the same size and are connected as follows:

$$\mathbf{K} = \mathbf{K}_p + \Delta\mathbf{K} \tag{5.12}$$

In the proposed method, the preconditioning matrix $\mathbf{K}_{prec}$ of the iterative GMRES method becomes the $\mathbf{K}_p$ matrix of the non-overlapping model. This means that for each iteration, a linear system involving the non-overlapping stiffness matrix needs to be solved in order to evaluate the preconditioned quantity $\mathbf{w}_j = \mathbf{K}_{prec}^{-1}\mathbf{v}_j$ of algorithm 4.

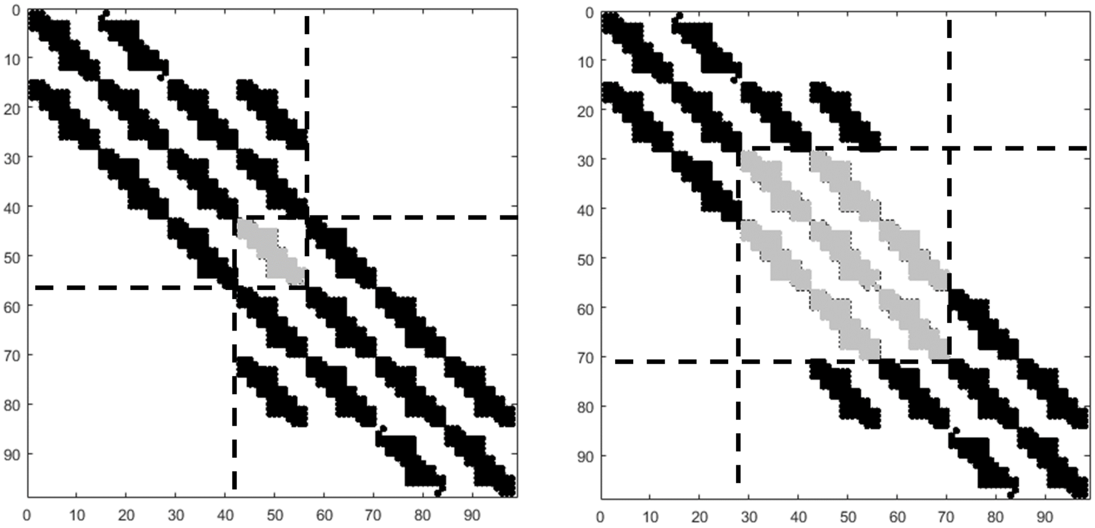
Figure 5.5, illustrates the major difference between the overlapping Schwarz method compared the non-overlapping alternative. Figures 5.4 a,b depict the interface problem, in case of a non-overlapping (Figure 5.4a) and a non-overlapping (Figure 5.4b). In the first case, 7 control points compose the subdomain interface, while in the second case 21. The effect of the increased interface problem to the matrix decomposition is provided in Figures 5.5a,b. As expected the interface problem in the non-overlapping case is significantly reduced. Specifically, in case of the OAS method, each preconditiong step of GMRES method is evaluated by superposition of the solution obtained by each



(a) Interface control point of non-overlapping 2x1 domain decomposition.

(b) Interface control points of overlapping 2x1 domain decomposition.

Figure 5.4: Overlapping vs Non-overlapping domain partitioning effect on the interface problem.



(a)

(b)

Figure 5.5: Interface stiffness elements for (a) Non-overlapping and (b) Overlapping decompositions.

one of the subdomain matrices. This leads to the solution of the interface dof for each subdomain. On the contrary, the non-overlapping domain decomposition methods requires a single evaluation of the Schur complement of the stiffness matrix to the interface problem, where the preconditioning step is now evaluated. The benefits of the proposed alternatives are profound, as the introduced errors depend only on the neglected terms of the non-overlapping decomposition, while the solution of the interface problem required for the evaluation of the preconditioning step is significantly smaller and thus less computationally intensive than the solution of all subdomain matrices required in OAS.

5.4.1 PSM

PSM [71, 33], emanating from the method of substructures [146] is considered to be the standard DDM method. This method subdivides the domain into subdmomains and after elimination of internal degrees of freedom of each subdomain the resulting equations refer to the interface problem among subdomains. The goal of PSM is to reduce the multitude of the dofs of the initial system only to the boundary dof between the subdomains. This transforms the initial linear system $\mathbf{K} \cdot \mathbf{u} = \mathbf{f}$ to

$$\mathbf{S} \cdot \mathbf{u}_b = \hat{\mathbf{f}}_b \tag{5.13}$$

where the matrix $\mathbf{S}$ is the Schur complement of the initial stiffness matrix $\mathbf{K}$ to the interface dof between the subdomain. Matrix $\mathbf{S}$ can be assembled by adding the contributions of each of the subdomain matrices as follows:

$$\mathbf{S} = \sum_{s=1}^{N_s} \overline{\mathbf{S}}^s \tag{5.14}$$

where $\overline{\mathbf{S}}$ is the contribution of the subdomain matrix to the boundary problem Schur complement, and N_s the number of subdomains:

$$\overline{\mathbf{S}}^s = (\mathbf{L}_b^s)^T \mathbf{S}^s \mathbf{L}_b^s \tag{5.15}$$

In eq. (5.15), the matrix $\mathbf{L}_b$ is an unsigned Boolean matrix that maps the local boundary dof of the subdomain to the global boundary dof of the problem, while the matrix $\mathbf{S}^s$ is calculated by:

$$\mathbf{S}^s = \mathbf{K}_{bb}^s - (\mathbf{K}_{ib}^s)^T (\mathbf{K}_{ii}^s)^{-1} \mathbf{K}_{ib}^s \tag{5.16}$$

The same procedure is applied for the subdomain forces, where the right hand side of eq. (5.13) can be expanded as

$$\hat{\mathbf{f}}_b = \mathbf{f}_b - \sum_{s=1}^{N_s} (\mathbf{L}_b^s)^T (\mathbf{K}_{ib}^s)^T (\mathbf{K}_{ii}^s)^{-1} \mathbf{f}_i^s \tag{5.17}$$

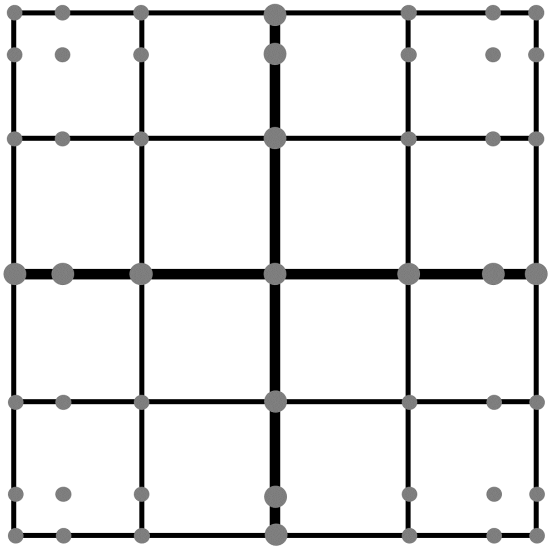
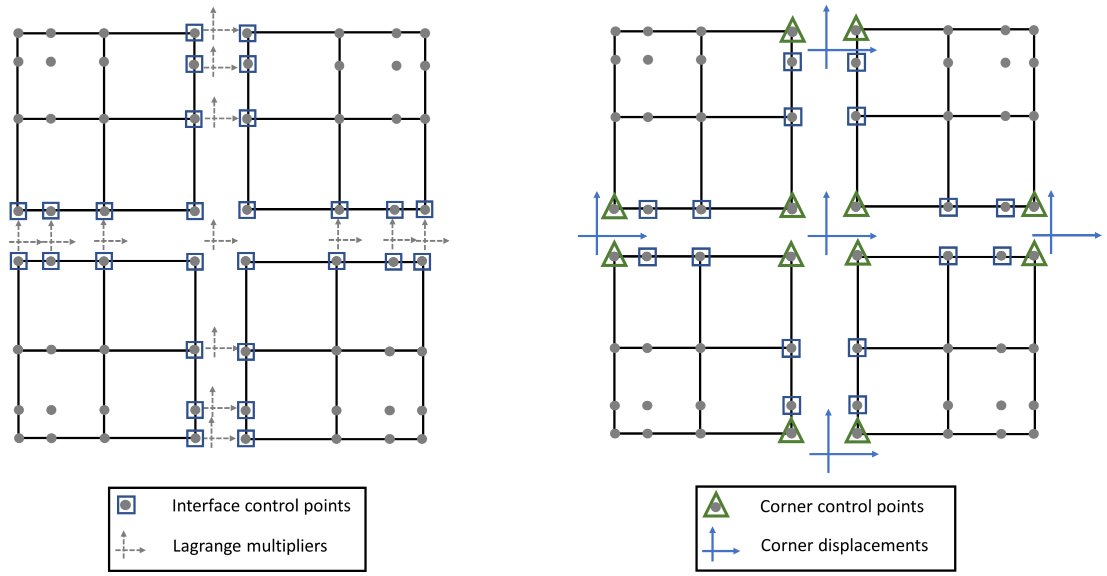


Figure 5.6: Non-overlapping decomposition of a unit square domain with PSM method.

Figure 5.6, schematically illustrates the PSM domain decomposition method. Specifically, for a unit square domain partitioned into 4x4 isogeometric elements and 2x2 subdomains. The interface problem consists of 13 control points. All dof of the initial domain (Figure 5.6a) are condensed to the interface dof and subsequently the interface problem is solved, as described by the PSM procedure above.

5.4.2 FETI-DP

FETI-DP [68, 103, 121, 64, 105] is a well established domain decomposition solver for the case of FEM that utilizes both primal PSM and dual FETI, DDM formulations. Its performance in IGA was studied in [107] and was extended to discontinuous Galerkin problems in [87].



(a) Lagrange multipliers **(b)** Coarse problem displacements

Figure 5.7: Non-overlapping decomposition of a unit square domain with FETI-DP method.

Figure 5.7 illustrated schematically displacements and traction forces acting on the interface of FETI-DP. Specifically, the initial domain of Figure 5.6a, is discretized with 4x4 isogeometric elements,and is partitioned into 2x2 subdomains. In contrast to PSM method, the unknown values are no longer the displacements of the interface problem, but their dual quantities the traction forces of the interface nodes. The traction forces are the Lagrange multipliers, illustrated in Figure 5.7a, which are necessary to enforce continuity between the subdomains. In order to accelerate the propagation of information among the subdomains, the displacements of the coarse problem are taken into account as well. The coarse problem is formed by the nodes that belong to more than two subdomains aswell as by the nodes of the boundary belonging to two subdomains in the 2D case. An example of the coarse problem displacements of FETI-DP is shown in Figure 5.7b. The unknown quantities at the corner nodes are their displacements.

The algorithmic stages of the method are briefly described below. The stiffness matrix K^s, force vector f^s and displacement vector u^s of each patch (s) can be partitioned as follows:

$$K^s = \begin{bmatrix} K^s_{ii} & K^s_{ib} \\ K^s_{bi} & K^s_{bb} \end{bmatrix} \tag{5.18a}$$

$$f^s = \begin{bmatrix} f^s_i \\ f^s_b \end{bmatrix} \tag{5.18b}$$

$$u^s = \begin{bmatrix} u^s_i \\ u^s_b \end{bmatrix} \tag{5.18c}$$

with subscript (i) denoting the internal dof and (b) the boundary dof.

In order to impose continuity between subdomains, the following equation is introduced:

$$\sum_{s=1}^{N_s} B^s u^s = 0 \tag{5.19}$$

with N_s being the number of subdomains and B^s being a signed Boolean matrix. The u^s_b vector is further partitioned as:

$$u^s_b = \begin{bmatrix} u^s_{br} \\ u^s_{bc} \end{bmatrix} \tag{5.20}$$

with the subscript (c) denoting the coarse problem nodes or "corner" nodes. Subsequently, K^s, u^s and f^s can be partitioned as follows:

$$K^s = \begin{bmatrix} K^s_{rr} & K^s_{rc} \\ K^s_{cr} & K^s_{cc} \end{bmatrix} \tag{5.21a}$$

$$f^s = \begin{bmatrix} f^s_r \\ f^s_{bc} \end{bmatrix} \tag{5.21b}$$

$$u^s = \begin{bmatrix} u^s_r \\ u^s_{bc} \end{bmatrix} \tag{5.21c}$$

where

$$f_r^s = \begin{bmatrix} f_i^s \\ f_{br}^s \end{bmatrix} \tag{5.22a}$$

$$u_r^s = \begin{bmatrix} u_i^s \\ u_{br}^s \end{bmatrix} \tag{5.22b}$$

and subscript (r) denoting all dof that do not belong to a corner node.

Considering this repartitioning, the continuity between patches can be re-written as:

$$\sum_{s=1}^{N_s} B_r^s u_r^s = 0 \tag{5.23}$$

with B_r^s being the signed Boolean matrix B^s, pertaining to the (r) dof. Moreover, the following global vector of corner dof is introduced

$$u_c = \begin{bmatrix} u_c^1 & ... & u_c^j & ... & u_c^{N_c} \end{bmatrix}^T \tag{5.24}$$

with u_c^j denoting all displacement dof attached to the j-th global node that is also a corner node of the mesh decomposition and N_c denoting the total number of corner nodes.

In order to connect u_c with u_{bc}^s, the following equation is introduced:

$$L_c^s u_c = u_{bc}^s \tag{5.25}$$

with B_c^s being an unsigned Boolean matrix.

Considering all the above equations, the patch equilibrium condition is expressed as:

$$K_{rr}^s U_r^s + K_{rc}^s L_c^s u_c = f_r^s - B_r^{sT} \lambda \tag{5.26a}$$

$$\sum_{s=1}^{N_s} L_c^{sT} {K_{rc}^s}^T u_r^s + \sum_{s=1}^{N_s} L_c^{sT} K_{cc}^s u_c = \sum_{s=1}^{N_s} L_c^{sT} f_{bc}^s = f_c \tag{5.26b}$$

with λ being the traction forces between each patch.

From the above relations, it follows that

$$u_r^s = (K_{rr}^s)^{-1}(f_r^s - B_r^{sT}\lambda - K_{rc}^s L_c^s u_c) \tag{5.27}$$

which, when substituted to eq. (5.23), the following system of equations arises

$$\begin{bmatrix} F_{I_{rr}} & F_{I_{rc}} \\ {F_{I_{rc}}}^T & -K_{cc}^* \end{bmatrix} \begin{bmatrix} \lambda \\ u_c \end{bmatrix} = \begin{bmatrix} d_r \\ f_c^* \end{bmatrix} \tag{5.28}$$

where:

$$F_{I_{rr}} = \sum_{i=1}^{N_s} B_r^s {K_{rr}^s}^{-1} B_r^{sT} \tag{5.29a}$$

$$F_{I_{rc}} = \sum_{i=1}^{N_s} B_r^s {K_{rr}^s}^{-1} K_{rc}^s B_c^s \tag{5.29b}$$

$$K_{cc} = \sum_{i=1}^{N_s} B_c^{sT} K_{cc}^s B_c^s \tag{5.29c}$$

$$K_{cc}^* = K_{cc} - \sum_{i=1}^{N_s} (K_{rc}^s B_c^s)^T {K_{rr}^s}^{-1} (K_{rc}^s B_c^s) \tag{5.29d}$$

$$d_r = \sum_{i=1}^{N_s} B_r^s {K_{rr}^s}^{-1} f_r^s \tag{5.29e}$$

$$f_c^* = f_c - \sum_{i=1}^{N_s} (K_{rc}^s B_c^s)^T {K_{rr}^s}^{-1} f_r^s \tag{5.29f}$$

By eliminating u_c from the baove equations, the following interface problem needs to be solved:

$$(F_{I_{rr}} + F_{I_{rc}} {K_{cc}^*}^{-1} F_{I_{rc}}^T)\lambda = d_r - F_{I_{rc}} {K_{cc}^*}^{-1} f_c^* \tag{5.30}$$

The solution of eq. (5.30) is usually performed with an iterative solution algorithm and for the case of isogeometric collocation method where non-symmetric matrices occur with, the GMRES algorithm has been used.

5.4.3 P-FETI-DP

P-FETI-DP [76] method constitutes an enhancement of the aforementioned PSM method, as it introduces a preconditioner $\tilde{A}^{-1}$ to the interface system of eq. (5.13), that is based on the coarse problem of the FETI-DP method. Its expression is given by:

$$\begin{aligned}\tilde{\mathbf{A}}^{-1} = (\tilde{\mathbf{L}}^e_{pr})^T(\mathbf{K}^e_{rr})^{-1}\tilde{\mathbf{L}}^e_{pr} + \big({}_b\mathbf{N}_{bc} - (\tilde{\mathbf{L}}^e_{pr})^T(\mathbf{K}^e_{rr})^{-1}\mathbf{K}^e_{rc}L^e_c\big)(\mathbf{K}^*_{cc})^{-1} \\ \big(- (\mathbf{L}^e_c)^T\mathbf{K}^e_{cr}(\mathbf{K}^e_{rr})^{-1}\tilde{\mathbf{L}}^e_{pr} +_{bc} \mathbf{N}_b\big)\end{aligned} \tag{5.31}$$

Starting from left to right all components of the preconditioner are explained below: $\tilde{\mathbf{L}}^e_{pr}$ matrix is used to map the remainder dof of the discretized domain and the boundary problem of the PSM method. Its expression is given by

$$\tilde{\mathbf{L}}^e_{pr} = {}_r\mathbf{N}^e_b\mathbf{L}^e_{pb} \tag{5.32}$$

The purpose of matrix ${}_r\mathbf{N}^e_b$ as its subscripts indicate the compatibility of matrix dimensions. When subdivided into subdomains it can be written with the following block diagonal expression

$${}_r\mathbf{N}^e_b = \begin{bmatrix} {}_r\mathbf{N}^{(1)}_b & & \\ & \ddots & \\ & & {}_r\mathbf{N}^{(N_s)}_b \end{bmatrix} \tag{5.33}$$

Each of the subdomain contributions ${}_r\mathbf{N}^{(N_s)}_b$ is a Boolean matrix that maps the remainder dof of the subdomain (s) to the boundary dof the same subdomain. The matrix $\mathbf{L}^e_{pb}$ of eq. (5.32) can be similarly written

$$\mathbf{L}^e_{pb} = \begin{bmatrix} \mathbf{L}^{(1)}_{pb} \\ \vdots \\ \mathbf{L}^{(N_s)}_{pb} \end{bmatrix} \tag{5.34}$$

while for each subdomain the matrix $\mathbf{L}^{(s)}_{pb}$ is computed as

$$\mathbf{L}^{(s)}_{pb} = \mathbf{L}^s_b\mathbf{W} \tag{5.35}$$

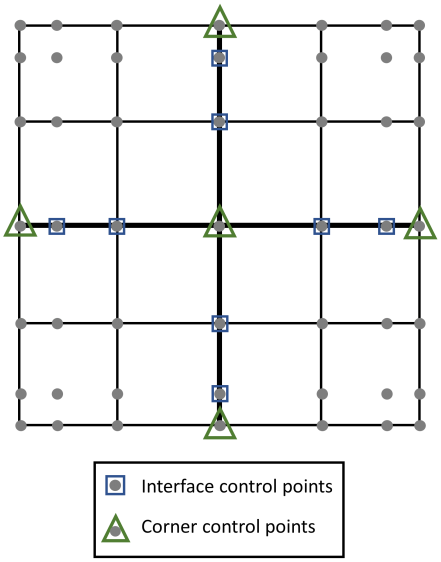


Figure 5.8: P-FETI-DP domain partitioning

In eq. (5.35), the matrix $\mathbf{L}_b^s$ is a Boolean matrix that maps local boundary dofs of the subdomain to the interface dof of the PSM problem. Additionall, the matrix $\mathbf{W}$ is a diagonal matrix, with terms equal to the inverse multiplicity of the interface dof. The term multiplicity for the case of interface dof, refers to the number of subdomains this dof belongs to. Moving to matrices $\mathbf{K}_{rr}^e, \mathbf{K}_{rc}^e$, these constitute block diagonal matrices generated from the partitioning of the subdomain matrices to corner and remainder dof as per eq. (5.21d).

$$\mathbf{K}_{rr}^e = \begin{bmatrix} \mathbf{K}_{rr}^{(1)} & & \\ & \ddots & \\ & & \mathbf{K}_{rr}^{(N_s)} \end{bmatrix} \tag{5.36a}$$

$$\mathbf{K}_{rc}^e = \begin{bmatrix} \mathbf{K}_{rc}^{(1)} & & \\ & \ddots & \\ & & \mathbf{K}_{rc}^{(N_s)} \end{bmatrix} \tag{5.36b}$$

while the $\mathbf{K}_{cc}^*$ matrix is denoted as the Schur complement of the remainder dof of each subdomain, to its boundary corner ones and consequently its mapping to the global boundary dof of the corner problem defined by FETI-DP. The matrix ${}_bN_{bc}$ of eq. (5.31), is also Boolean mapping matrix, that links the boundary dof of the PSM problem, to the corner dof of the FETI-DP method. Finally, the matrix $\mathbf{L}_c$ is written in subdomain form as

$$\mathbf{L}_c^e = \begin{bmatrix} \mathbf{L}_c^{(1)} \\ \vdots \\ \mathbf{L}_c^{(N_s)} \end{bmatrix} \tag{5.37}$$

and defines an unsigned Boolean matrix that links the local corner dof of each subdomain to the equivalent corner dof of the coarse problem. Note that for all Boolean matrices involved in this process, no matrix vector multiplications are required to be performed, rather than the utilization of the proper submatrix.

Figure 5.8, illustrates schematically the P-FETI-DP method. Specifically, for a unit square domain discretized with 4x4 isogeometric elements and 7x7 control points, a 2x2 decomposition is performed. The interface problem control points are indicates with squares. All internal dof of the subdomains are condensed to the interface dof, as described by the PSM procedure. In order to accelerate the solution of the PSM interface problem of eq.(5.13), a preconditioner is crated based on the coarse problem of FETI-DP method operated on the interface control points. This strategy allows for a faster convergence of the interface problem, of the multitude of subdomain or the size of the interface problem.

5.4.4 Primal and dual non overlapping domain decomposition preconditioners for IGA collocation methods

For assessing the performance of non-overlapping domain decomposition-based preconditioners for the solution of IGA problems, we considered both primal and dual variants. Primal variants, such as the PSM method described above, introduce a subdomain interface problem, whose unknown quantities match the unknown field of the PDE. In the case of linear elasticity, this means that the unknowns of the interface quantities are the displacements of the interface nodes among adjacent subdomains. On the contrary, dual methods, create an interface problem, whose unknown quantities are the duals of the displacements, which are the subdomain interaction forces of the interface nodes. In the literature, often these dual quantities that represent the interaction forces are denoted as Lagrange multipliers, due to the mathematical theory used for their development.

While both the aforementioned variants worked as expected for the Galerkin IGA implementation, the dual variants were problematic when stiffness matrices were composed using the collocation procedure. In particular, all the dual DD variants examined produced erroneous results while some of them failed to converge. This behavior can be explained by examining the mechanical interpretation of these two family of solution methods. Dual variants perform iterations on the Lagrange multiplier vector λ, as opposed to the interface displacements of the primal ones. For Galerkin-composed stiffness matrices, each matrix component K_{ij} represents the force required to be exerted on the degree of freedom i to cause a unit displacement for degree of freedom j. For elasticity problems as considered in this work, the opposite also applies according to Betti's law which means that a unit displacement applied in degree of freedom j will result in this unit force reaction in degree of freedom i. This leads to a physical explanation of symmetric nature of the Galerkin-composed stiffness matrices. However, collocation-composed matrices are not symmetric and this mechanical interpretation does not apply. As a result, if the same process considered for the formulation of dual domain decomposition methods for Galerkin-composed stiffness matrices is applied for collocation-composed matrices, the resulting solver fails.

5.4.5 Stiffness matrix-based non-overlapping preconditioner

Following the presentation of the P-FETI-DP method, this section will provide the process of constructing the non-overlapping decomposition of the reference domain, as well as the resulting stiffness matrix which will be used as the preconditioner to the GMRES method. In the univariate case, given an axis defined by the knot value vector $\Xi = \{\xi_1 = 0, ..., \xi_{n+p+1}\}$, where p being the polynomial degree and n the number of control points, a subset of the control points are selected as the interface entities between adjacent subdomains.

$$S_k = \{i \in \mathbb{N} : g_m \leq i \leq g_{m+1}\}, \qquad \forall\ k \in \{1, ..., N\} \tag{5.38}$$

where N is the number of subdomains. The indices of these control points $g_m \in \{1, n\}$ are selected so that each subdomain partitioning S_k contains a similar multitude of control points. Following to the introduction of the interface control points, a diagonal weighting matrix $\mathbf{W}$ is introduced. Its order being the number of degrees of freedom of the univariate domain, while its values equal to the inverse number of subdomains each degree of freedom belongs to.

The aforementioned univariate partitioning can be extended in the bivariate case as a tensor product of the univariate decomposed axes. This results to equations analogous to eq. (5.38) which can be expressed as follows:

$$S_{kl} = \{(i,j), \in \mathbb{N}^2 : g_m \leq i \leq g_{m+1}, h_m \leq j \leq j_{m+1}\} \tag{5.39}$$

where $k \in \{1, N\}$ and $l \in \{1, M\}$, N,M being the number of subdomains per parametric direction. The weighting matrix $\mathbf{W}$ is also computed by the inverse multiplicity of the subdomains each dof belongs to. The extension to the trivariate case is also straightforward. As a result, the subdomain stiffness matrices are calculated by

$$\mathbf{K}_{kl} = \mathbf{L}_w \mathbf{K} \mathbf{L}_w^T \tag{5.40}$$

where $\mathbf{L}_w$ being a scaling matrix given by

$$\mathbf{L}_w = \mathbf{L}_s \mathbf{W} \tag{5.41}$$

and the matrix $\mathbf{L}_s$ maps the dof of the subdomain to the dof of the global domain. Based on the coarse points definition provided for the FETI-DP method, their definition can be generalized for the case of isogeometric analysis. For a two-dimensional domain, coarse problem control points are defined as those points whose multitude of converging subdomains, in the non-overlapping subdomain decomposition, either equals to four or equal to two for those belonging to the boundary edges of the bivariate domain.

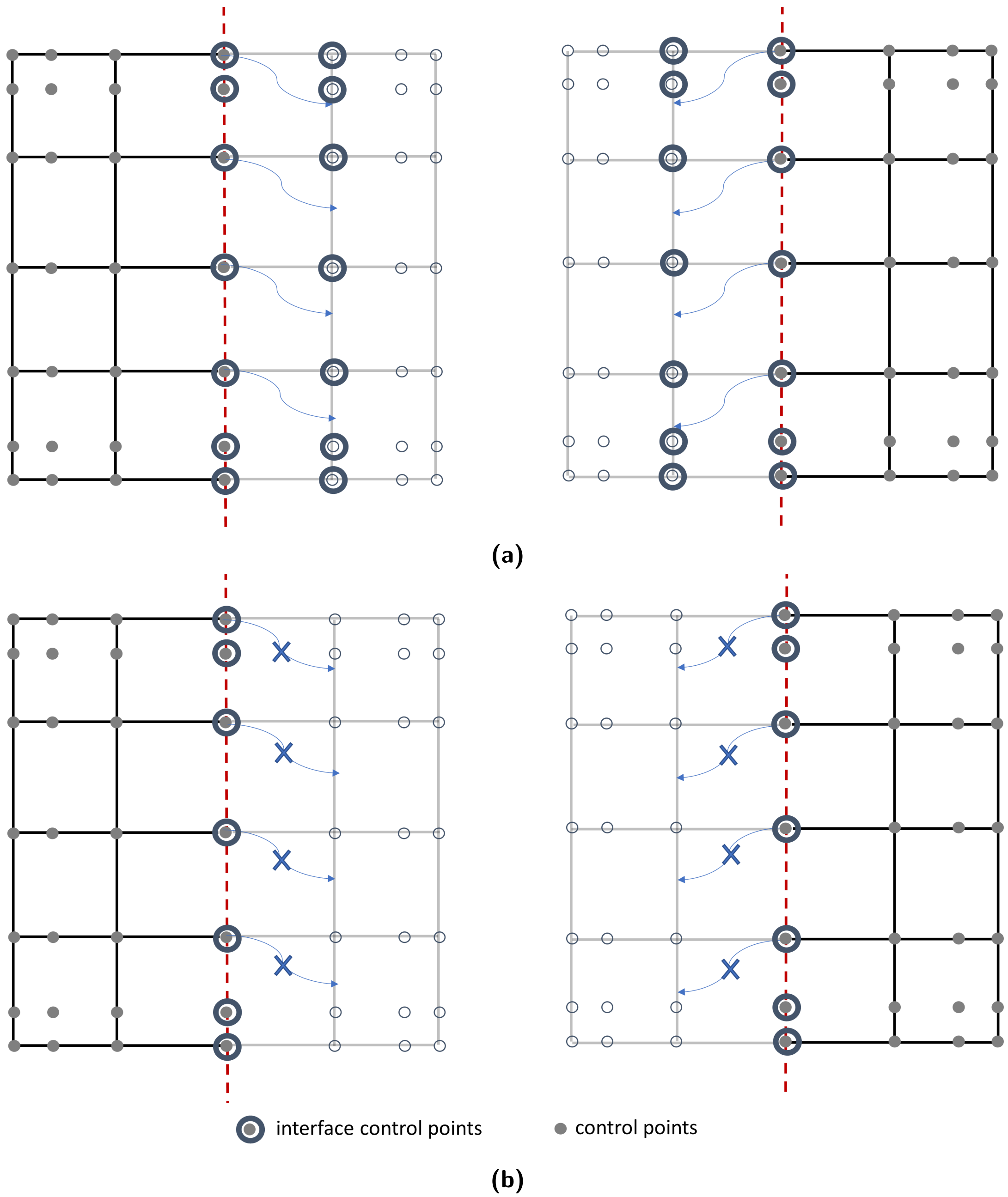


Figure 5.9: (a) Overlapping (b) non-overlapping domain partitionings

The domain decomposition process, described above differs from the partitioning of FEM.Specifically, in the context of finite elements, the subdomain partitioning was performed by requiring all subdomain to have similar number of elements. Especially, when the domain was discretized with the same element types, the equal element number among subdomains guaranteed an equal distribution of the degrees of freedom and ultimately load balancing. On the other hand, in case of IGA, the domain partitioning is performed according to the multitude of control points each subdomain contains. It is apparent that this partitioning does split the domain into equally spaced areas due to the nature of isogeometric shape functions, yet ensures properly load balanced subdomains in terms of their matrix size.

Another difference among the two methods is that the partitioning of IGA requires an averaging of the matrices at the interface dof. The reason for this is that each integration quantity such as the element contributes to matrices areas outside its definition domain and thus the stiffness of a control point cannot be divided into its elemental contribution in a straight forward manner. This averaging process is further explained in Figure 5.9, by partitioning axis X into two subdomains, each one containing a multitude of four control points along this axis. The final control point column of the first subdomain (Figure 5.9a) and first column of the second subdomain (Figure 5.9b) create the interface problem. As can observed, by retaining the existing interelement continuity among subdomain would result in shape functions penetrating neighbouring subdomains. This property of isogeometric shape functions is no longer desired as this would to increase influence among subdomains. To alleviate this, any such influence is truncated as depicted in Figures 5.9c,d. In terms of stiffness matrix, this translated to the decomposition of Figure 5.10, where the gray area illustrates the interface among subdomains. Since the interface matrix are belong to two adjacent subdomains, an averaging is performed that splits the interface contribution to two equal parts, assigning each to one to the corresponding subdomain.

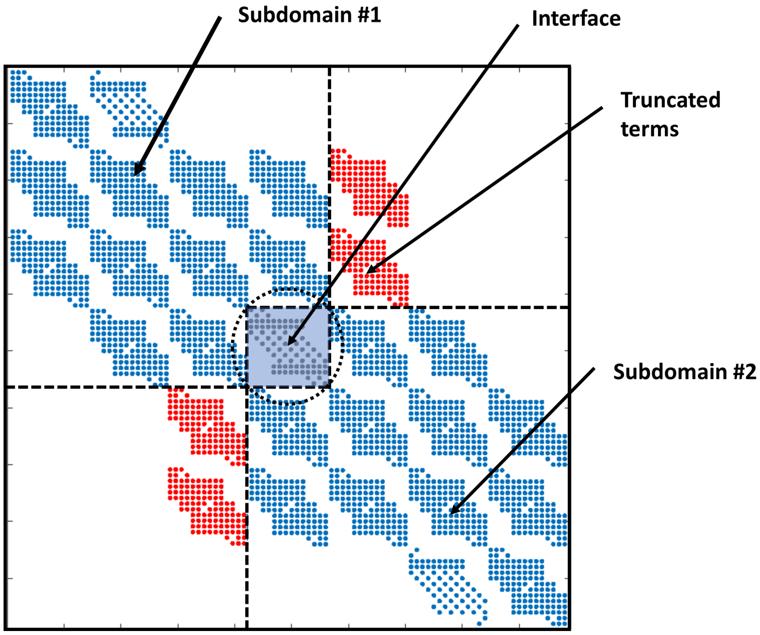


Figure 5.10: Non-overlapping decomposition of a collocation stiffness matrix generated for a unit square domain with 7x7 control points and polynomial degree p=4.

Apart from the averaging performed at the interface among adjacent subdomains, Figure 5.10 showcases the truncated terms that are ignored during the creation of the preconditioner via the non-overlapping decomposition. Specifically, as can be observed, stiffness matrix terms that do not belong to regions defined by the domain partitioning are disregarded. This allows the non-overlapping decomposition to retain a minimal interface problem among adjacent subdomains, thus allowing the computationally efficient solution of the respective preconditioner.

5.5 Numerical results

In this section, several numerical tests are performed in order to test the convergence properties of the P-FETI-DP preconditioner accelerated with GMRES, when utilized for the solution of the linear systems deriving from linear elasticity PDEs. The reference domain utilized for the numerical examples is a unit square, with homogeneous Dirichlet boundary conditions aplied to all degrees of freedom of its left edge and vertical load of

100 KN applied to upper right interpolatory control point of the domain.

The domain is discretized by utilizing isogeometric NURBS shape functions, with parameters h denoting the mesh size, polynomial degree of the shape functions p, subdivided into N subdomains per parametric direction. The Young's modulus is 100 MPa and Poisson's ratio $\nu = 0.3$. The reference linear systems are solved with the aid of the GMRES iterative method, with zero initial guess and a normalized displacement convergence criterion of 10^{-6}.

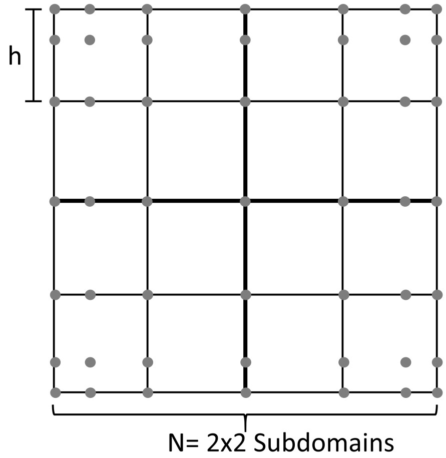


Figure 5.11: Unit square domain utilized for the scalability tests of the numerical tests.

5.5.1 Preconditioned GMRES performance metrics

In this subsection, a comparison of the performance, in terms of both iterations as well as CPU times in sequential execution environment, is performed between the proposed preconditioner method and its strongest competitor, the overlapping Schwarz method [25]. Numerical tests were performed in large-scale models, with dof ranging from 10^2K to 10^3K dof and subdomains numbers in the span between 16 and 1024. For all numerical tests, a convergence accuracy of 10^{-6} is considered for both the convergence of GMRES global problem as well as the interface problem in the case of the proposed method. The computing platform used is an Intel Core i7 X980 with 24Gb RAM memory. In all cases, for the solution of the resulting linear system, equivalent preconditioners are studied. Specifically, both one-level preconditioners, such as the the PSM-D method and

OAS with out coarse problem are provided, as well as 2-level preconditioners like the P-FETI-DP method and 2-level OAS.

The first comparison is performed for the case of approximately 130K dof model. Table 5.1, as well as Figures 5.12a,b present all the data pertaining to the aforementioned model. Figure 5.12a schematically illustrates the iterations behavior with an increasing number of subdomains. As it can be observed, both one level preconditioner alternatives present incremental rise of the number of iteration needed for convergence, with the PSM-D method requiring significantly more iterations compared all other methods with the P-FETI-DP method to require the least number of iterations to converge to the desired accuracy.

256x256			OAS 1-level		OAS 2-level		PSM-D			P-FETI-DP		
N	Interface	Coarse	Gmres	Time	Gmres	Time	Gmres	Prec	Time	Gmres	Prec	Time
	dof	dof	it.	seq. (s)	it.	seq. (s)	it.	it.	seq. (s)	it.	it.	seq. (s)
4x4	3060	36	129	32.71	81	19.29	1	297	14.32	1	26	7.59
8x8	7084	140	188	47.12	102	21.41	1	454	10.35	1	34	4.14
16x16	14940	540	280	102.56	126	25.45	2	690	27.7	2	33	4.33
32x32	29884	2108	442	355.33	149	30.07	2	964	132.37	2	31	10.62

Table 5.1: Performance metrics for the solution 100K dof problem with various preconditioners to the GMRES method.

In terms of wall-clock time metrics, the situation is reversed among the PSM-D and OAS 1-level methods. Specifically, a 2.28x speedup is noted for the minimum case of 16 subdomains, which slowly rises to 2.36x in the case of 1024 subdomains. This speedup of PSM-D method is attributed to the nature of the non-overlapping preconditioner proposed, as the solution of the global problem is transformed to the solution of the interface problem among adjacent subdomains, which is orders of magnitude smaller than the original problem, as can be seen in Table 5.1.

In a similar fashion, P-FETI-DP method outperforms the OAS 2-level alternative. The biggest speedup of P-FETI-DP method is observed for a multitude of 64 subdomains and is 5.17x faster than the equivalent OAS 2-level alternative. Note that P-FETI-DP wall-clock time metric are affected by the multitude of subdomains, as a descending path is followed by an ascending one. The reason for this is that the method requires to strike a balance between the size of subdomain problems and the corresponding coarse problem, so that a small interface problem is maintained, while the coarse problem is large enough to provide an effective preconditioner. This behavior was expected, as it has been observed in previous publications regarding non-overlapping domain decomposition methods [76, 173].

A similar behavior to the 100K dof case, is observed for the 500K test case. In terms of iteration count, the PSM-D preconditioner performs poorly, while P-FETI-DP method outperforms all other preconditioners as depicted in Figure 5.13a. In terms of wall-clock time, both PSM-D and OAS 1-level preconditioners exhibit an incremental rise of the wall-clock time with number of subdomains, with the first being up to 2.6x times faster. The same observation holds for the two-level methods. An inspection of the results of Table 5.2 and Figure 5.13b, yield the same descending, ascending behavior of the P-FETI-DP preconditioner. The highest speedup provided by P-FETI-DP, is observed for 256 non-overlapping subdomains and is 8.9x times faster than its OAS counterpart. Note that, for the comparison to be fair among the two-level methods, the overlapping between subdomains in the OAS case, is kept at a minimum of one overlapping shape function.

512x512			OAS 1-level		OAS 2-level		PSM-D			P-FETI-DP		
N	Interface dof	Coarse dof	Gmres it.	Time seq. (s)	Gmres it.	Time seq. (s)	Gmres it.	Prec it.	Time seq. (s)	Gmres it.	Prec it.	Time seq. (s)
4x4	6132	36	179	282.45	92	130.09	1	426	192.38	1	27	93.09
8x8	14252	140	265	265.02	120	136.96	2	645	94.4	2	36	37.37
16x16	30300	540	393	760.62	157	215.55	2	983	157.31	2	36	24.26
32x32	61628	2108	550	1965.27	176	246.33	3	1361	758.54	3	33	45.25

Table 5.2: Performance metrics for the solution 500K dof problem with various preconditioners to the GMRES method.

Finally, for the 1000K dof case, the former observations are validated. In overall, the same behavior regarding the iteration metrics is observed for all preconditioner cases, with two-level methods significatly outperforming the computational cost of one-level methods and P-FETI-DP being the most computationally efficient. Regarding the wall-clock execution time, P-FETI-DP achieves a 5.7x times speedup compared to the OAS 2-level method, hence proving its efficiency in isogeometric collocation problems of various scales.

Note that the parallelization of the aforementioned preconditioners is not bound by the number of subdomains. This means that the number of cores that can be utilized can exceed the total number of subdomains, as apart from the utilization of one core per subdomain for the matrix vector operations of eq. (5.16), the undelying solution of K_{ii} with the subdomain vectors can be further parallelized with block direct or iterative solvers that spawn the solution of the internal problems across multiple cores [76]. Thus,

768x768			OAS 1-level		OAS 2-level		PSM-D			P-FETI-DP		
N	Interface	Coarse	Gmres	Time	Gmres	Time	Gmres	Prec	Time	Gmres	Prec	Time
	dof	dof	it.	seq. (s)	it.	seq. (s)	it.	it.	seq. (s)	it.	it.	seq. (s)
4x4	9204	36	218	977.81	113	336.25	1	523	756.97	2	28	358.93
8x8	21420	140	323	1270.14	149	373.24	2	794	542.42	2	38	163.65
16x16	45660	540	478	1892.42	233	410.23	2	1207	467.72	3	37	72.026
32x32	93372	2108	669	4655.56	257	447.22	3	1670	1729.4	3	35	102.81

Table 5.3: Performance metrics for the solution 1000K dof problem with various preconditioners to the GMRES method.

the speedup of the P-FETI-DP preconditioner, compared to its overlapping competitors, can be retained even in fine grain parallel computing environments. Note that the current approach for the solution of the subdomain internal problems, in both P-FETI-DP and OAS preconditioners, is by an LU factorization of the respective matrices.

5.5.2 P-FETI-DP scalability in N

In this section, the scalability of the proposed method is studied. GMRES iterations required for the convergence of the global and equivalent interface problems are reported in Table 5.4. For a unit square domain, different mesh partitionings are considered ranging from $1/h = 8$ elements to $1/h = 128$ elements for each direction. These meshes are the subdivided into N subdomains with multitude ranging from 4 to 1024. As it can be observed, the non-overlapping model serves as excellent preconditioner for the solution of the global linear system. Iteration count for the global GMRES are minimal, which is attributed to the minimization of the discarded stiffness matrix values. An in-depth investigation of the polynomial degree influence on iteration count is provided in a later section.

GMRES with P-FETI-DP preconditioner (p=3)										
	1/h=8		1/h=16		1/h=32		1/h=64		1/h=128	
N	GMRES	Interface	GMRES	Interface	GMRES	Interface	GMRES	Interface	GMRES	Interface
	it.	it.	it.	it.	it.	it.	it.	it.	it.	it.
2x2	1	7	1	8	1	9	1	9	1	10
4x4	-	-	1	9	2	10	2	11	2	12
8x8	-	-	-	-	1	9	2	11	2	11
16x16	-	-	-	-	-	-	3	10	2	10
32x32	-	-	-	-	-	-	-	-	2	9

Table 5.4: Iteration metrics

Table 5.4 examines the scalability of the PSM method when preconditioned with the first step of the coarse problem of the FETI-DP which is briefly described as the P-FETI-DP method. As it can be observed, iteration count for the solution of the global problem is kept at a minimum due to the non-overlapping model of the interface problem introduced, while the iteration count for the solution of the interface problem remains unaffected by the mesh partitiong $1/h$ as well as by the number of subdomains N. This renders the P-FETI-DP method an ideal candidate for the solution of linear systems deriving from isogeometric collocation methods. For all numerical tests of Table 5.4 the required preconditioner accuracy was set to 10^{-6} for both global and interface GMRES solutions.

5.5.3 P-FETI-DP robustness with respect to preconditioner accuracy

This numerical investigation, addresses the influence of interface problem preconditioner accuracy to the overall performance of the proposed iterative method. Table 5.5 shows the effect of the accuracy of the interface for various subdomain partitionings ranging from 4 to 1024 subdomains In agreement to the iteration count presented in Table 5.4, the minimal number of external iterations required for the solution of the global problem seems to be unaffected by the relaxation of the solution accuracy of the interface problem.

In case of the P-FETI-DP method, the decrease of the preconditioner accuracy reduces the number of required iterations by half, while resulting to a minimal effect on the total iterations of the global problem.

GMRES with P-FETI-DP preconditioner (1/h=150)										
N	Subd.	Interface	GMRES	Prec.	GMRES	Prec.	GMRES	Prec.	GMRES	Prec.
	dofs	dofs	it.	it.10^{-6}	it.	it.10^{-5}	it.	it.10^{-4}	it.	it.10^{-3}
2x2	11026	596	1	10	2	9	2	7	2	6
4x4	2594	1776	1	12	2	10	2	8	3	5
8x8	612	4088	2	12	2	9	2	8	2	5
16x16	171	7980	2	10	2	7	2	6	2	5

Table 5.5: Iteration metrics investigation of the preconditioner accuracy effect of the P-FETI-DP method.

5.5.4 P-FETI-DP dependence on p

In order to study the effect of polynomial degree on the efficiency of the method, a numerical investigation was performed for the domain with $1/h = 64$ mesh size, subdivided into 16 non-overlapping subdomains. Table 5.6, provides the total iterations count needed for the solution of the domain defined, by utilizing as a preconditioner both aforementioned methods PSM-D and P-FETI-DP. As expected, all methods provide an effective and robust preconditioner regarding the iteration count of the global problem, utilizing GMRES. The augmentation of the polynomial degree has a minor effect of the total number of global problem iterations. For polynomial degree $p = 4$ to $p = 7$ an increase of only 10 external iterations is observed, while for all chosen interface problem preconditioners, the total number of iterations required for attaining a 10^{-6} remain unaffected by the polynomial degree increase. This can be attributed to the fact that the introduced non-overlapping model abolishes the augmented interelement continuity. The truncated terms of the initial stiffness matrix only affect the external GMRES iterations, due to the deviation of the truncated model compared to initial one that becomes more pronounced as matrix bandwidth and interelement continuity increase along with the polynomial degree.

h=1/64, N =4x4						
	PSM Unprec.		PSM-D		P-FETI-DP	
p	GMRES it.	Interface it.	GMRES it.	Interface it.	GMRES it.	Interface it.
4	28	42	28	25	28	5
5	32	40	32	21	32	5
6	38	38	38	21	38	5
7	41	41	41	22	41	5

Table 5.6: Effect of polynomial degree of the shape function to the total iteration count.

5.5.5 P-FETI-DP robustness with respect to ill-conditioning

In this section, the effect of ill-conditioning to the proposed preconditioner is studied. Table 5.7, presents the iterations metrics of a unit square domain with different mesh partitionings ranging from $1/h = 16$ to $1/h = 128$ and three cases of Poisson's ratio, namely 0.3,0.45 and 0.4995. As it can be observed, the non-overlapping model that is used for preconditioning the solution, is unaffected by the increase of the ill-conditioning of the resulting stiffness matrices. On the contrary, interface problem iteration metrics increase with both Poisson's ratio with mesh partitioning. The P-FETI-DP preconditioner case exhibits reduced iteration metrics, that validate the robustness of the proposed method even in extreme ill-conditing cases.

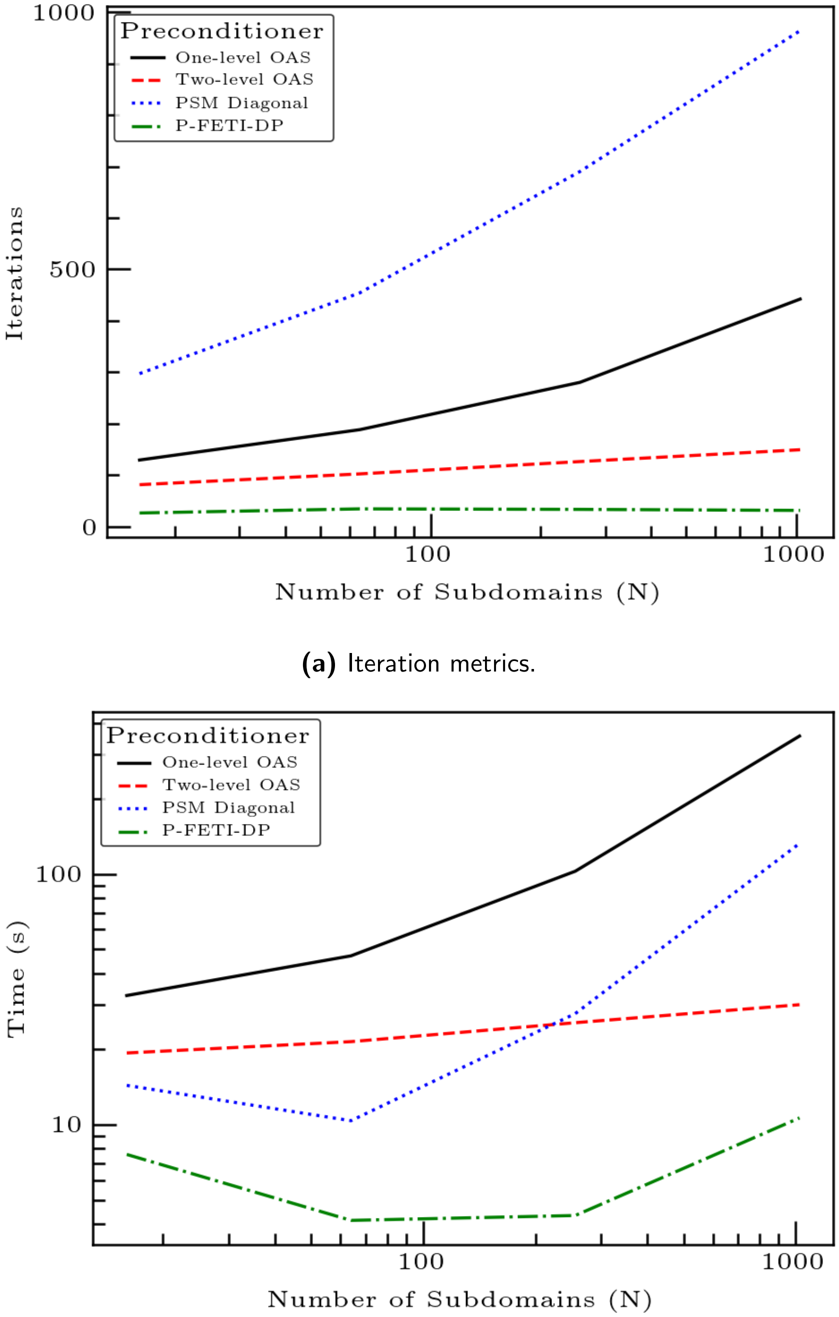
GMRES with P-FETI-DP preconditioner (N=4x4)								
	1/h=16		1/h=32		1/h=64		1/h=128	
ν	GMRES it.	Interface it.	GMRES it.	Interface it.	GMRES it.	Interface it.	GMRES it.	Interface it.
0.3	1	9	2	10	2	11	2	12
0.45	1	14	2	21	2	26	1	35
0.4995	1	20	1	31	1	41	1	45

Table 5.7: Iteration metrics for unit square domain with various mesh partitionings and Poisson's ratio, preconditioned with P-FETI-DP.

GMRES with OAS preconditioner (N=4x4)				
	1/h=16	1/h=32	1/h=64	1/h=128
ν	GMRES it.	GMRES it.	GMRES it.	GMRES it.
0.3	22	27	45	99
0.45	22	29	47	106
0.4995	22	30	47	108

Table 5.8: Iteration metrics for unit square domain with various mesh partitionings and Poisson's ratio, preconditioned with P-FETI-DP.

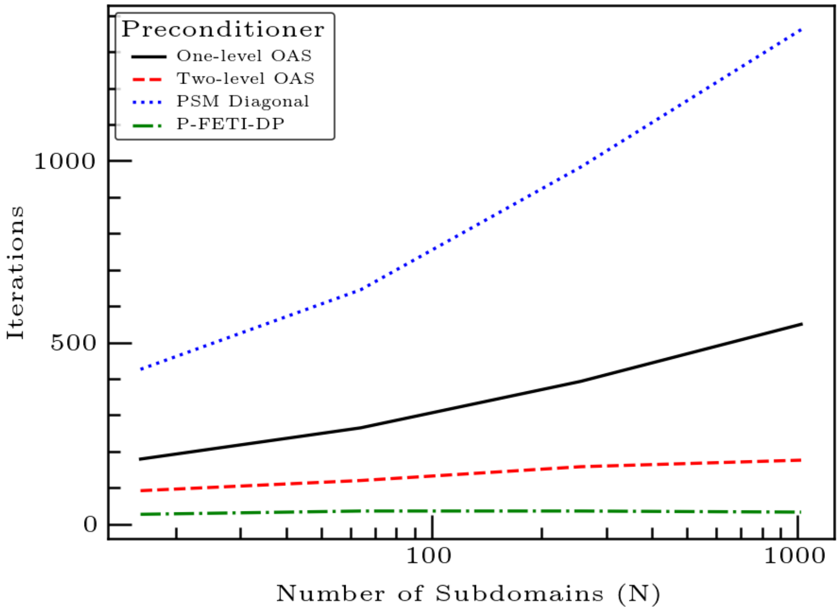
To compare the robustness of the P-FETI-DP with OAS 2-level preconditioner, the same numerical investigation with repsect to ill-conditioning was performed with respect to ill conditioning. Table 5.8, show the effect of ill conditioning on the performance of OAS method. As it can be observed, the increase of the Poisson's ratio has a minor effect on the iterations of the global problem. Yet, in contrast to the equivalent metrics of the P-FETI-DP method, a significant increase of the iteration metrics is observed as the mesh becomes finer, thus rendering OAS 2-level preconditioner less effective with increasing mesh.



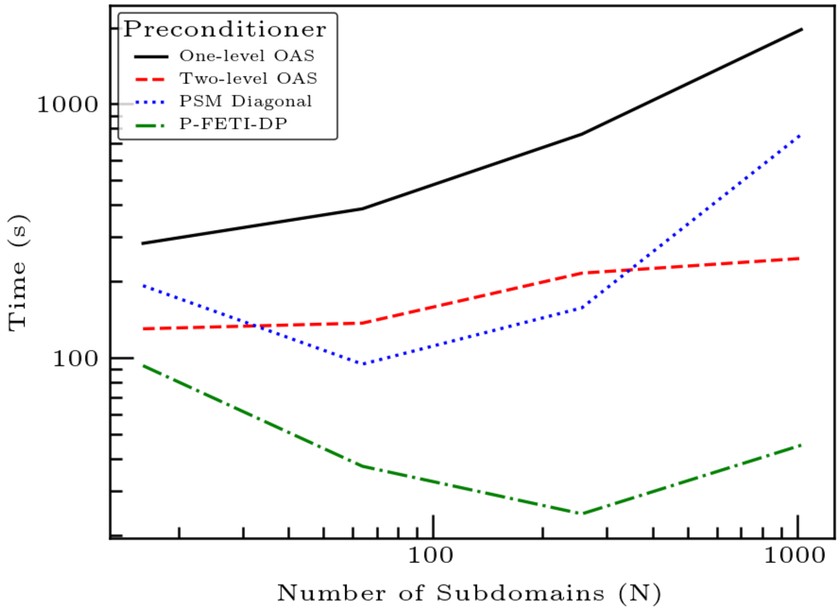
(a) Iteration metrics.

(b) Wall-clock time metrics.

Figure 5.12: Iteration and wall-clock time metrics of the different preconditioners with increasing subdomain partitioning. 100K problem size.

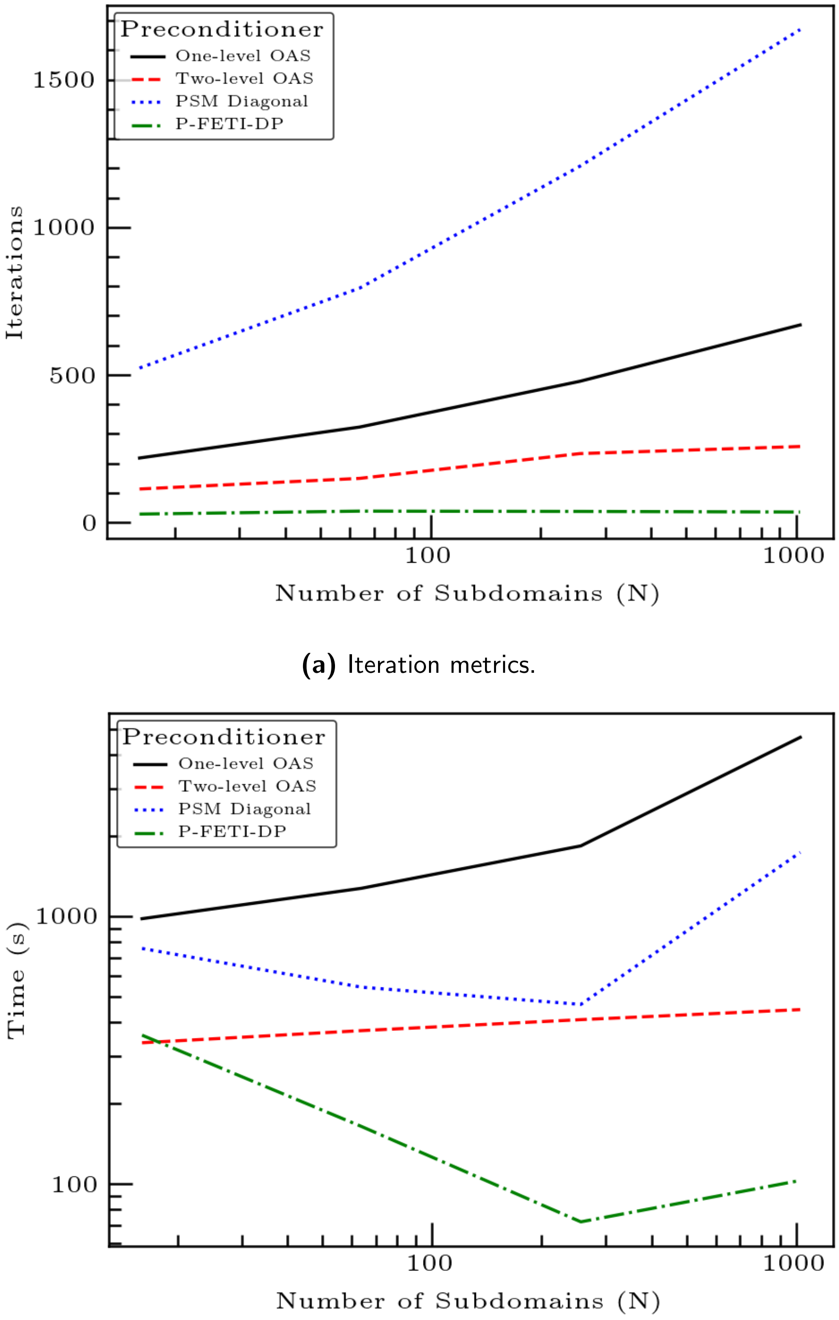


(a) Iteration metrics.



(b) Wall-clock time metrics.

Figure 5.13: Iteration and wall-clock time metrics of the different preconditioners with increasing subdomain partitioning. 500K problem size.



(a) Iteration metrics.

(b) Wall-clock time metrics.

Figure 5.14: Iteration and wall-clock time metrics of the different preconditioners with increasing subdomain partitioning. 1000K problem size.

6

Isogeometric Kirchhoff-Love shell formulation for multiscale material simulations

6.1 Introduction

Thin lightweight structures are often modeled either with the Kirchhoff-Love theory, in case of thin shells, or with Reissner-Midlin theory, in case of thick shells. Even though most product belong to thin or very thin shell category, the computational limitations of thin shell theory when formulated in the framework of the FEM method, made Reissner-Midlin theory dominant in most FEM codes. This is attributed to the prerequisite of Kirchhoff-Love shells for a minimum C^1 continuity between adjacent elements, which classical FEM could not easily provide, thus resulting to non-conforming meshes. Special care must be taken for thin shell theory in the context of FEM, by applying non-local formulations, nodal enforcementof the C^1 continuity, or even penalty method applied to selected material points. On the other hand, IGA, enables for a straightforward implementation of Kirchhoff-Love theory, as continuity is raised from typical C^0 to C^{p-1}, where p being the polynomial degree of the shape functions used.

This Chapter aims to extend existing isogeometric thin shell formulations to incorporate constitutive laws generated by a stochastic mutliscale analysis. A nested IGA-FEM multiscale analysis scheme is proposed, in which IGA is used for the discretization at the macroscopic level and standard FEM are used for the discretization of the RVEs assigned at each integration point of the macrostructure. The discretization of the macroscale structure is performed with T-Splines, which are favored over NURBS, as they allow the extraction of arbitrarily complex geometries without a laborious pre-processing stage.

6.2 A two-scale nested IGA-FEM formulation

The models considered in the present work are macroscale models of shell structures consisted in composite materials that contain arbitray inclusions and cavities. These heterogeneities are located in the microstructure of the material and are thus undetectable at the macroscale level of the model. As a result, the material is considered to be homogeneous in the macroscale, while highly disparate in its microstructure. To determine the influene of the material impurities to its actual constitutive relations, a nested IGA-FEM solution scheme is proposed, which can be split into three discrete steps as described in Figure 6.1.

In the first step, IGA discretization of the macroscale model takes place, which is performed using T-Splines as described in Chapter 1 (Figure 6.1a). The Bezier extraction of the T-Spline model is automatically performed via a CAD tool and resulting Bezier elements are utilized for the computation of the stiffness matrix. For each mid-surface integration position (point A_0 in Figure 6.1b), a multitude of through thickness integration points (points A_i in Figure 6.1c) are considered, each one corresponding to the material zone of a subsection in which plane stress conditions are assumed. This layered representation of the shell's section enables the modelling of through thickness material variability.

In the second step, the material constitutive relations are extracted at each material integration point via stochastic computational homogenization. For each integration point A_i, a random 3D RVE is generated as a prismatic mesoscale model describing the local microstructural topology of the material. Arbitrary volumetric inclusions comprised of either cavities and/or surface/line inclusions are accurately specified. Three-dimensional elasticity continuum finite elements are used for the discretization of the matrix, while inclusions are modeled as structural elements such as shells and beams embedded in the matrix. The random RVE generator is described in detail in Section

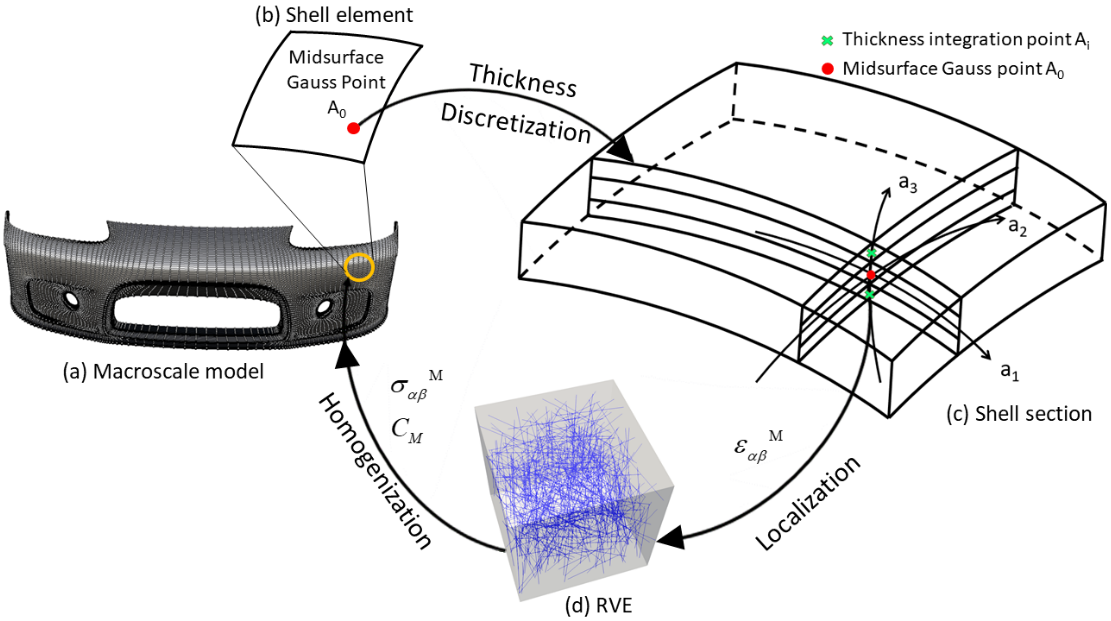

(b) Shell element
Midsurface Gauss Point A_0
Thickness
Discretization
✖ Thickness integration point A_i
● Midsurface Gauss point A_0
a_3
a_2
a_1
(a) Macroscale model
(c) Shell section
$\sigma_{\alpha\beta}{}^M$
C_M
Homogenization
$\varepsilon_{\alpha\beta}{}^M$
Localization
(d) RVE

Figure 6.1: Schematic representation of the nested IGA-FEM flow for the analysis of Kirchhoff-Love shells (a) IGA discretization (b) Shell element discretization (c) Through-thickness discretization at midsurface point A_0 of a shell element (d) Random RVE of local microstructural material totpology at thickness integration point A_i.

6.2.4. For the computational homogenizations, the total macroscopic in-plane strains $\epsilon_{\alpha\beta}^M$ of equation 2.56 at a distance ζ of the mid-surface, are assigned to the corresponding section integration material point (i.e. point A_i in Figure 6.1c) represented by a 3D RVE of the composite material (localization step of Figure 6.1). The in-plane strains $\epsilon_{\alpha\beta}^M$ are enforced as equivalent displacements to the 3D RVE, following the transformation described in Section 6.2.2. This way, an RVE microscale boundary value problem is formulated which computes averaged equivalent material properties, namely the stress $\sigma_{\alpha\beta}^M$ and the corresponding plane stress consitutitve matrix C_M and returns them back to the macroscale Gauss Point A_0.

The final step of the two-scale nested scheme integrates through the thickness of the shell element and computes the stress resultants at the mid-surface point A_0 of the macroscopic model as well as their sensitivities. A detailed description of this step is provided in Section 6.2.3. Note that for the purposes of this work a linear response of the RVE is assumed. However, the proposed scheme can be readily applied to non-linear cases in straightforward manner. The steps of the proposed IGA-FEM scheme are presented in detail in the following Sections.

6.2.1 Microscale boundary value problem

Neglecting inertia forces, we assume that the RVE in Figure 6.1d deforms in a state of equilibrium [108]:

$$\nabla \cdot \sigma = 0 \qquad in\ \Omega_{RVE} \tag{6.1}$$

One of the basic assumptions of the proposed formulation is that the total strain of the macro-continuum equals the volume average of the microscopic strain:

$$\epsilon_M = \frac{1}{V_{RVE}} \int_{\Omega_{RVE}} \epsilon_m dV \tag{6.2}$$

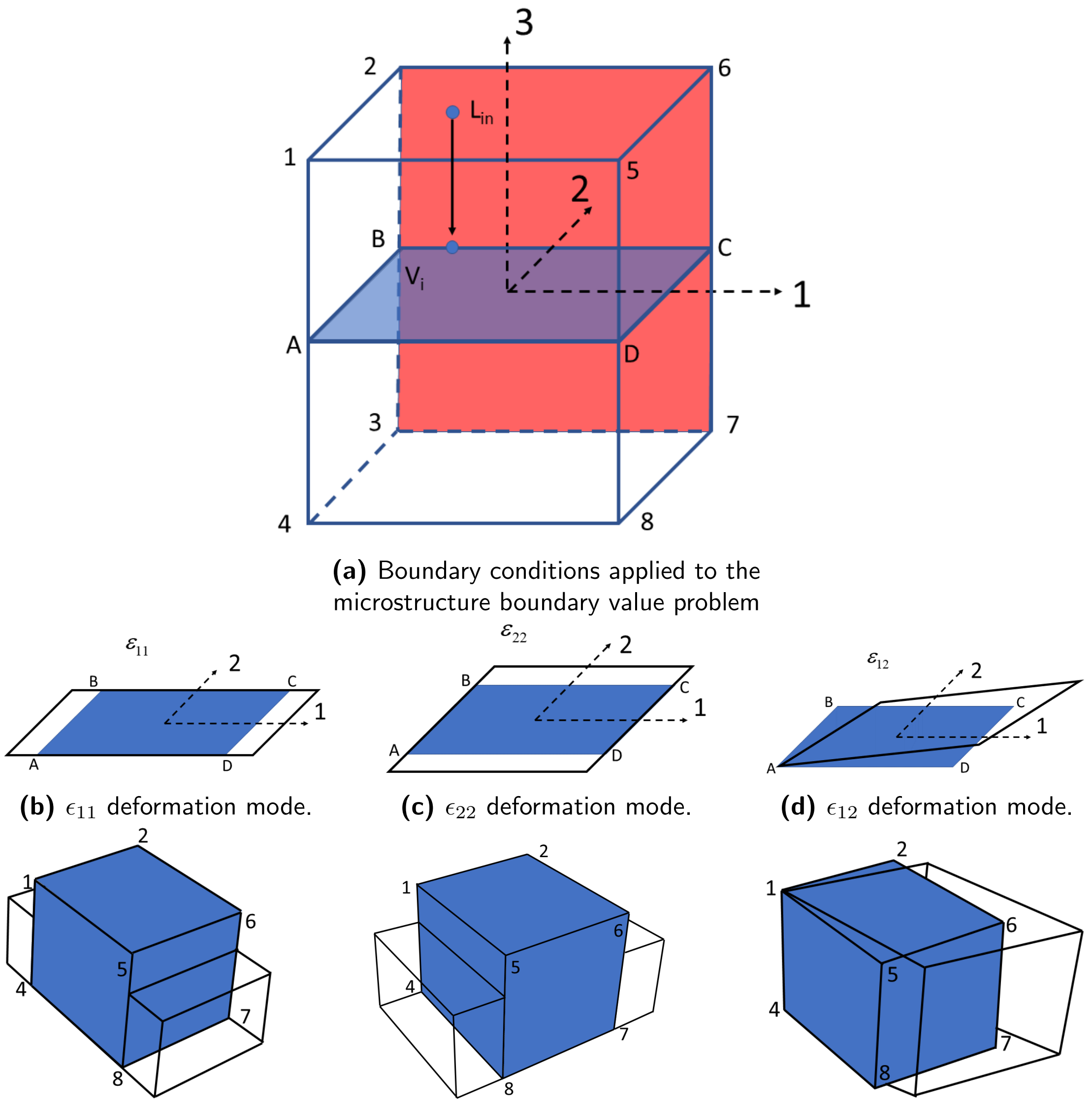
For the case of thin shells, this classic strain averaging condition is translated to an in-plane constraint only. Accordingly, in the following equation, a circumflex denotes restriction of the strain tensor to its in-plane coefficients only that are given be equation 2.56and they refer to the covariant basis of the undeformed configuration.

$$\hat{\epsilon}_M = \frac{1}{V_{RVE}} \int_{\Omega_{RVE}} \hat{\epsilon}_m dV \tag{6.3}$$

This condition is met by imposition of appropriate constraints in the 3D RVE boundary value problem. Specifically, a virtual plane stress element (element ABCD in Figure 6.2a) is embedded at the mid-surface of the RVE as shown in Figure 6.2. Then, in-plane dispplacements are equal to

$$\hat{u}_m = \hat{\epsilon}_M \hat{\mathbf{X}}_m \tag{6.4}$$

are prescribed to all perimeter nodes of the virtual element ABCD, while all peripheral mesh nodes of the 3D RVE are subjected to the same displacementsas their projection nodes on the perimeter of the virtual element. More specifically, with reference to Figure 6.2, and edge nodal point L_{in} of the RVE with coordinates X_1, X_2, X_3 is prescribed with the same u_1 and u_2 displacement values , as dictated by eq. 6.4, with those of the corresponding vertex node V_i of the virtual element, i.e. with the node having the same



(a) Boundary conditions applied to the microstructure boundary value problem

(b) ϵ_{11} deformation mode.

(c) ϵ_{22} deformation mode.

(d) ϵ_{12} deformation mode.

(e) RVE ϵ_{11} deformation mode.

(f) RVE ϵ_{22} deformation mode.

(g) RVE ϵ_{12} deformation mode.

Figure 6.2: Microstructure boundary value problem on a 3D RVE.

X_1, X_2 coordinates.

$$u_1^{edge}(X_1, X_2, X_3) = \hat{u}_1^{vertex}(X_1, X_2) \tag{6.5a}$$

$$u_2^{edge}(X_1, X_2, X_3) = \hat{u}_2^{vertex}(X_1, X_2) \tag{6.5b}$$

This way, the in-plane enforcd displacements of the virtual element ABCD, make the 3D RVE to deform in a pattern that is consistent to the plane stress conditions. The basic 3D deformation modes consistent to plane stress $\epsilon_{11}^M, \epsilon_{22}^M, \epsilon_{12}^M$ are shown in Figure 6.2b-d. As for the out-of-plane strain coefficients ϵ_{33}^M, there is no need of imposing it as a aconstraint, since its value results indirectly from the out-of-plane deformation of the 3D RVE. As a result, out-of-plane shear strains $\epsilon_{3\alpha}^M$ are automatically set to zero. Special care is taken to constraint the out-of-plane rigid body motions. This is attained by constraining the displacements of three nodes of the bottom face, that are parallel to axis 3. Specifically, with reference to Figure 6.2a, vertical displacements of nodes 3,4,7 of the RVE are constrained. This boundary conditions allows the RVE to deform freely in the in-plane directions, while at the same time eliminating possible out-of-plane rigid body motions.

The following strain averaging relation concerning the stress holds [108]:

$$\hat{\sigma}_M = \frac{1}{V_{RVE}} \int_{\Omega_{RVE}} \hat{\sigma}_m dV = \frac{1}{V_{RVE}} \int_{\partial\Omega_{RVE}} sym[\hat{t} \otimes x] dA \tag{6.6}$$

where the divergence theorem has been applied and $\hat{t} = \sigma \cdot n$ is the field of the developed tractions on the boundary of the RVE. Out of plane traction t are identically equal to zero, as no constraint is enforced in the direction 1 and the plane stress condition $\sigma_{33}^M = 0$ holds. By combination of eqs. 6.2 and 6.6 and taking into account eq. 6.4, the 3D macroscopic stress power equals to the volume average of the microscopic stress power.

$$\sigma_M : \dot{\epsilon} = \int_{\Omega_{RVE}} \frac{1}{V_{RVE}} \sigma_m : \dot{\epsilon}_m dV \tag{6.7}$$

This relation satisfies the Hill-Mandel principle of macro-homogeneity [84]. Thsi way, the macroscopic energetic conjugate strain and stress measures can be replaced by their averaged microscopic counterparts in the expression of the principle of virtual work at the macroscale level.

6.2.2 Enforcing plane stress conditions

A standard finite element solution scheme is employed for the boundary value problem of Figure 6.2, expressed at the microlevel in eq. 6.2. Upon discretization of the displacement field u_m and after constraining the rigid body moves of the 3D RVE model, the remaining degrees of freedom are partitioned into two categories, the constrained ones enforced by eq. 6.4 and the internal nodes that are totally free. The constrained nodes, denoted as e, include the in-plane displacement degrees of freedom of the peripheral nodes of the RVE and the internal, denotes as p, are the remainder internal nodes.

$$u = \begin{bmatrix} u_e \\ e_p \end{bmatrix} \tag{6.8}$$

The same partitioning applies to the internal nodal forces P and the stiffness matrix K as well:

$$f = \begin{bmatrix} f_e \\ f_p \end{bmatrix} \tag{6.9a}$$

$$K = \begin{bmatrix} K_{ee} & K_{ep} \\ K_{pe} & K_{pp} \end{bmatrix} \tag{6.9b}$$

For a peripheral node q of the RVE, the in-plane displacement vector u_q is derived from the corresponding projection of the RVE on ABCD and is given as:

$$u_q = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T = D_q^T \hat{\epsilon} \tag{6.10}$$

where D_q and $\hat{\epsilon}^T$ are given as:

$$D_q := \frac{1}{2} \begin{bmatrix} 2x_1 & 0 \\ 0 & 2_x2 \\ x_2x_1 & \end{bmatrix} \tag{6.11a}$$

$$\hat{\epsilon}^T = \begin{bmatrix} \epsilon_{11}^M & \epsilon_{22}^M & 2\epsilon_{12}^M \end{bmatrix} \tag{6.11b}$$

Assembling a global matrix D associated with all peripheral nodes P, the discretized form of the constraint eq. 6.4 is given as:

$$u_e = D^T \hat{\epsilon}, \qquad D = [D_1, d_2, ..., D_p] \tag{6.12}$$

Regarding the reaction forces acting on the internal degrees of freedom "e", an appropriate Lagrange multiplier δ is applied to enforced eq. 6.12. Then the discretized boundary value problem takes the form of the following algebraic equations:

$$f_e = \delta \tag{6.13a}$$

$$f_p = 0 \tag{6.13b}$$

$$u_e - D^T \hat{\epsilon} = 0 \tag{6.13c}$$

Upon solution of the micro-level boundary value problem of Figure 6.2, the macroscopic stress tensor can be calculated for eq.6.6 that takes the discretized form:

$$\hat{\sigma} := \frac{1}{|V|} D\delta \tag{6.14}$$

For an equilibrium state of the microstructure and for an infinitesimal macroscopic strain increment $\Delta\hat{\epsilon}$, the corresponding increments of the displacements $\Delta u_e, \Delta u_p$, Lagrange multiplier $\Delta\delta$ and the total macroscopic strain $\Delta\hat{\sigma}$, is given as follows:

$$\Delta u_e = D^T \Delta\hat{\epsilon} \tag{6.15a}$$

$$\Delta u_p = -{K_{pp}}^{-1} K_{pe} \Delta u_e \tag{6.15b}$$

$$\Delta\delta = \tilde{K}_{ee} D^T \Delta\hat{\epsilon} \tag{6.15c}$$

$$\Delta\hat{\sigma} := \frac{1}{|V|} D\Delta\delta \tag{6.15d}$$

where $\tilde{K}_{ee} = K_{ee} - K_{ep}{K_{pp}}^{-1}K_{pe}$. Hence the macroscopic moduli are given as:

$$C = \frac{1}{|V|} D\tilde{K}_{ee} D^T \tag{6.16}$$

6.2.3 Macroscopic shell stiffness matrix

The macroscopic shell stiffness matrix matrix is derived from the principle of virtual work which can be written as

$$W = \int_V \sigma_M : \delta\epsilon_M dV = \int_V f \cdot \delta u_M dV \tag{6.17}$$

where the virtual displacement δu_M is in the macroscopic level. The expression of internal virtual work at the left hand side is further elaborated as:

$$\int_V \sigma_M : \delta\epsilon_M dV = \int_A (\mathbf{n} : \delta\epsilon_M + \mathbf{m} : \delta\kappa_M) dA \tag{6.18}$$

where through thickness integration delivers the macroscopic stress resultants as

$$n^{\alpha\beta} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_M^{\alpha\beta} d\zeta m^{\alpha\beta} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_M^{\alpha\beta} \zeta d\zeta \tag{6.19a}$$

where α, β take the values 1,2. Similar expressions are used for ”thickness integrated” material matrices [100]. The moduli at each thickness integration point are obtained directly from eq.6.16 as:

$$C_A = \int_{-\frac{h}{2}}^{\frac{h}{2}} C d\zeta \tag{6.20a}$$

$$C_B = \int_{-\frac{h}{2}}^{\frac{h}{2}} C\zeta d\zeta \tag{6.20b}$$

$$C_D = \int_{-\frac{h}{2}}^{\frac{h}{2}} C\zeta^2 d\zeta \tag{6.20c}$$

The differentials of membrane forces and bending moments are computed from:

$$dn^{\alpha\beta} = C_{M_A}^{\alpha\beta\gamma\delta} d\epsilon_{\gamma\delta} + C_{M_B}^{\alpha\beta\gamma\delta} d\kappa_{\gamma\delta} \tag{6.21a}$$

$$dm^{\alpha\beta} = C_{M_B}^{\alpha\beta\gamma\delta} d\epsilon_{\gamma\delta} + C_{M_D}^{\alpha\beta\gamma\delta} d\kappa_{\gamma\delta} \tag{6.21b}$$

Since equilibrium must be met for any variation of the displacement matrices δu_r it can be written that

$$\delta W = \frac{\partial W}{\partial u_r}\delta u_r = 0 \tag{6.22}$$

This leads to the following internal and external forces:

$$F_r^{internal} = \int_A (\mathbf{n} : \frac{\partial \epsilon_M}{\partial u_r} + \mathbf{m} : \frac{\kappa_M}{\partial u_r})dA \tag{6.23a}$$

$$F_r^{external} = \int_A f \cdot \frac{\partial u}{\partial u_r} dA \tag{6.23b}$$

Subsequently, the stiffness matrix is obtained as:

$$K_{rs}^{internal} = \int_A (\frac{\partial \mathbf{n}}{\partial u_s} : \frac{\partial \epsilon_M}{\partial u_r} + \frac{\partial \mathbf{m}}{\partial u_s} : \frac{\partial \kappa_M}{\partial u_r})dA \tag{6.24}$$

Substitution of eq.6.21 in eq.6.24 yields the final expression for the shell's macroscopic stiffness matrix:

$$K_{rs}^{internal} = \int_A (\mathbf{C}_{M_A} : \frac{\partial \epsilon_M}{\partial u_s} + \mathbf{C}_{M_B} : \frac{\partial \kappa_M}{\partial u_s}) : \frac{\partial \epsilon_M}{\partial u_r} + (\mathbf{C}_{M_B} : \frac{\partial \epsilon_M}{\partial u_s} + \mathbf{C}_{M_D} : \frac{\partial \kappa_M}{\partial u_s}) : \frac{\partial \kappa_M}{\partial u_r} dA \tag{6.25}$$

6.2.4 Random RVE generator

As mentioned above, a stochastic RVE generator is developed to account for material heterogeneities at the mesoscale level of RVEs. The generator is based on two uncertainty type, namely the inclusion dispersion and orientation inside the matrix as well as their spatial variability inside the RVE described with a variable volume fraction parameter [10].

RVE with embedded inclusions

Given the RVE dimensions, i.e. length L_{matrix}, width W_{matrix} and height H_{matrix} and the type of inclusions, the RVE generator distributes the inclusions in the matrix either uniformly in a unidirectional pattern (Figure 6.4), or with random orientations (Figure 6.10). The coordinates of an insertion point (Point A in Figure 6.3a) of an inclusion are randomly generated inside the matrix assuming a uniform distribution described by the uncorrelated random variables X_A, uniformly distributed in the range $[0, L_{matrix}]$, Y_A uniformly distributed in the range $[0, W_{matrix}]$ and Z_A, uniformly distributed in the range $[0, H_{matrix}]$.

To define the orientation of the inclusions, an additional point is inserted (Point B in Figure 6.3a) defining an additional orientation random variable, namely the Euler angler ϕ, which for the purposes of this work is also assumed uniformly distributed in $[0, 2\pi]$. This angle, together with a fixed length parameter completely define the coordinates of insertion point B.

The spatial variability of the Volume Fraction (VF) is taken into account by assigning a different RVE at each integration point of the domain. Specifically, a multitude of RVEs with varying VFs are constructed and randomly distributed throughout the macroscopic domain. The spatial varability of the VF used for the construction of the RVEs at different domain locations is modeled using random fields and simulated via series expansion of the Karhunen-Loève method [78]

$$w(\mathbf{x}, \theta) = \sum_{n=0}^{\infty} \sqrt{\lambda_n} \xi_n(\theta) f_n(\mathbf{x}) \tag{6.26}$$

where $\xi_n(\theta)$ is a set of uncorrelated Gaussian random variables and λ_n and $f_n(\mathbf{x})$ are the eigenvalues and eigenvectors of the autocorrelation functions, respectively.

Both the matrix and the inclusions are discretized independently, i.e. without common nodes. For the purposes of the numerical examples, solid finite elements are used for the discretization of the RVE matrix, while the inclusions in the form of carbon nanotubes are discretized with two-noded beam elements. The contribution of each inclusion to the stiffness of the RVE is calculated via the embedded element technique (Figure 6.3a), in which the discretized inclusion degrees of freedom are kinematically constrained to the degrees of freedom of the surrounding matrix. This compatibility between the matrix and the inclusions degrees of freedom, is demonstrated in the three-dimensional case, by utilizing a single spatial beam elements, representing a CNT, embedded in an n-noded continuum finite element. The degrees of freedom of the CNT element are given by

$$U_{beam} = \{u_1^b, u_2^b, u_3^b, ..., u_{12}^b\} \tag{6.27}$$

and the equivalent degrees of freeedom of the continuum finite element by:

$$U_{continuum} = \{u_1^c, u_2^c, u_3^c, ..., u_{3n}^c\} \tag{6.28}$$

The kinematic relation between the inclusion degrees of freedom and those of the solid finite element can be expressed as:

$$U_{inclusion} = [T_{embedded}]U_{matrix} \tag{6.29}$$

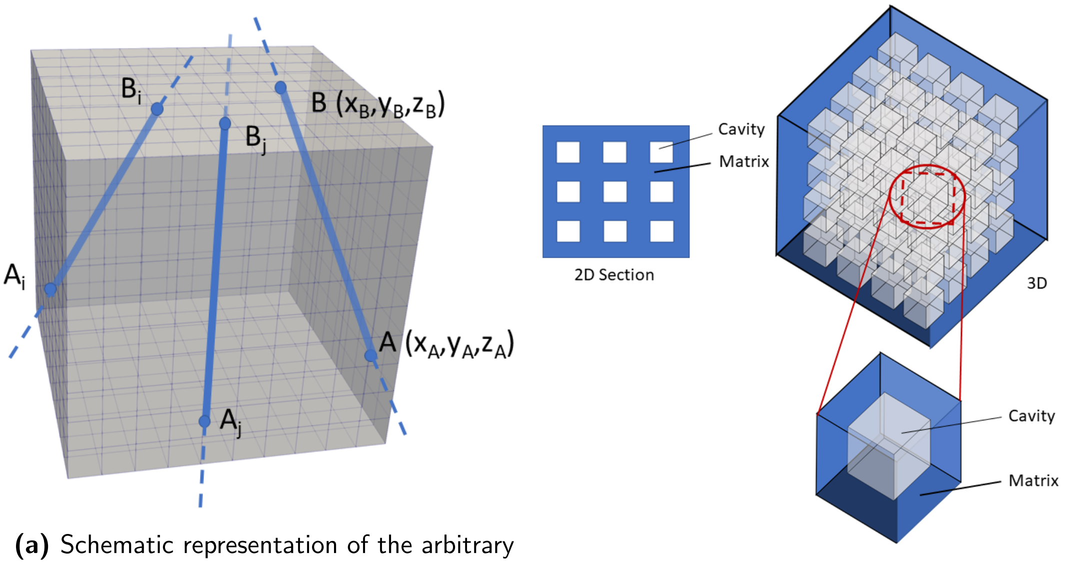
where T is a compatibility matrix that contains the shape function values at each of the inclusion nodes, and is given by

$$[T_{embedded}] = \begin{bmatrix} \mathbf{T}_A^{(1)} & \mathbf{T}_A^{(2)} & ... & \mathbf{T}_A^{(n)} \\ \mathbf{T}_B^{(1)} & \mathbf{T}_B^{(2)} & ... & \mathbf{T}_B^{(n)} \end{bmatrix} \tag{6.30}$$

where A, B denote the initial and final node of the beam element according to Figure 6.3a. $\mathbf{T}_K^{(j)}$ is a matrix that transforms the translational and rotational degrees of freedom of a beam element to the translational degrees of freedom of the matrix element, as follows:

$$\mathbf{T}_K^{(j)} = \begin{bmatrix} N_j(\xi_K, \eta_K, \zeta_K) & 0 & 0 \\ 0 & N_j(\xi_K, \eta_K, \zeta_K) & 0 \\ 0 & 0 & N_j(\xi_K, \eta_K, \zeta_K) \\ 0 & -\frac{1}{2}\frac{\partial N_j(\xi_K,\eta_K,\zeta_K)}{\partial z} & -\frac{1}{2}\frac{\partial N_j(\xi_K,\eta_K,\zeta_K)}{\partial y} \\ -\frac{1}{2}\frac{\partial N_j(\xi_K,\eta_K,\zeta_K)}{\partial z} & 0 & -\frac{1}{2}\frac{\partial N_j(\xi_K,\eta_K,\zeta_K)}{\partial x} \\ -\frac{1}{2}\frac{\partial N_j(\xi_K,\eta_K,\zeta_K)}{\partial y} & -\frac{1}{2}\frac{\partial N_j(\xi_K,\eta_K,\zeta_K)}{\partial x} & 0 \end{bmatrix} \tag{6.31}$$

The terms $\frac{\partial N_j}{\partial x}, \frac{\partial N_j}{\partial y}, \frac{\partial N_j}{\partial z}$ represent the shape function derivatives, evaluated at the coordinates ξ_K, η_K, ζ_K of the beam node at the local coordinate system of the solid element. As a result, the stiffness contribution of the inclusion to the matrix is given as follows:



(a) Schematic representation of the arbitrary inclusions discretized with beam elements, embedded in a 3D matrix, discretized with solid continuum finite elements.

(b) Schematic representation of metamaterial microstructure in 2D and 3D and its decomposition into 3D RVEs.

Figure 6.3: Examples of 3D RVEs.

$$K_{embedded} = T_{embedded}^{T} \cdot K_{inclusion} \cdot T_{embedded} \tag{6.32}$$

The final stiffness matrix of the RVE is calculated as the sum of the contributions of the continuum finite elements and the equivalent contributions of the beam elements transformed to the nodal degrees of freedom of their host elements with the aid of eq. 6.32. The use of the embedded FEM technique [127] allows for the use of relatively coarse and simple meshes, with respect to a standard FEM discretization. A standard FEM discretization would require fully conforming meshes in the matrix-inclusion interphases that would lead to complicated and laborious preprocessing of the RVE FEM models. This is avoided in the present work with the use of the embedded FEM technique. As a result, the proposed discretization procedure reduces significantly the computational cost for the solution of the RVE.

Metamaterial RVEs

Another typical material microstructure used in advanced composites is the metamaterial illustrated in Figure 6.3b. A metamaterial is created as a combination of a matrix with volumetric inclusions of different material properties or cavities. The volumetric nature of these inclusions does not allow their modelling as beams or shell elements. As a result, both the matrix and the inclusion are discretized with 3D continuum finite elements, ensuring that the mesh at the boundary between them is conforming. This might lead to increased RVE mesh size compared to the embedded inclusions presented in the previous section., yet it is inevitable in case of complex inclusion geometries. Since these materials do not exist in nature and are manufactured, they are commonly placed in a periodic lattice as Figure 6.3b illustrates. A single periodic portion of the lattice is extracted (Figure 6.3b) and serves as the RVE geometry to be used throughout the model. The stochasticity for this RVE case is introduced as the variability of the material properties of the matrix or the incorporated inclusions. Similar to the embedded inclusions case, the spatial variability of the material properties is dictated by the Karhunen-Loève method as provided in eq. 6.26.

6.3 Numerical results

This section provides benchmark examples that demonstrate the merits of the proposed computational methodology. The first example verifies the accuracy of the imposed plane stress conditions as described in Section6.2.2, followed by two real-scale examples that demonstrate the applicability of the proposed approach of integrating isogeometric T-Splines discretization of Kirchhoff-Love shells with stochastic multiscale analysis of composite materials. The second benchmark example illustrates the performance of the proposed methodology applied to a cylindrical shell composed of a metamaterial, while the third example deals with the analysis of a real-scale car bumper made of polymer reinforced with stochastically distributed Carbon Nanotube (CNT) inclusions.

All modules of isogeometric analysis, multiscale analysis and stochastic analysis were developed and integrated in the open-source computational mechanics software platform MSolve [MSo] with HPC capcabilities.

6.3.1 Verification of homogenization procedure for plane stress conditions

This benchmark test is used as a proof of concept of the proposed computational procedure described in Section 6.2.2 for extracting the plane stress constitutitve law from the analysis of a three-dimensional microstructure. For a Young's modulus E=4GPa and Poisson's ratio $\nu = 0.4$, the theoretic isotropic plane stress elasticity matrix is calculated as:

$$\mathbf{C} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{bmatrix} = \begin{bmatrix} 4.762 & 1.905 & 0 \\ 1.905 & 4.762 & 0 \\ 0 & 0 & 1.429 \end{bmatrix} \tag{6.33}$$

The same material properties are used for the computation of the constitutive matrix $\mathbf{C}'$ via the homogenization procedure implementing the plane stress extraction described in Section 6.2.2. To this purpose, a cubic RVE of 100nm edge is considered, and discretized with a mesh 10x10x10 hexahedral finite elements. The difference $\mathbf{\Delta C}$ between this calculation and the theoretic matrix of eq. 6.33 is computed at:

$$\mathbf{\Delta C} = \mathbf{C} - \mathbf{C}' = \begin{bmatrix} 0.0 & -9.992E-15 & -9.60E-16 \\ -9.992E-15 & 0.0 & -2.20E-16 \\ -7.08E-16 & -3.37E-16 & 0.0 \end{bmatrix} \tag{6.34}$$

The comparison of the constitutive matrix components in eq. 6.34 illustrates the accuracy of the proposed plane stress constraint procedure. It can be seen that the non-zero values of both matrices coincide, while the zero terms of the isotropic plane stress elastiity matrix are computed close to zero within the computer accuracy. This indicates that the extraction of the constitutive matrix from the homogenization procedure described in Section 6.2.2 provides accurate results compared to the analytical formula of plane stress costitutive matrix.

To test the performance of the proposed procedure for an anisotropic material, two composite RVE cases are examined with a matrix of the same material as before reinforced with CNTs of different volume fractions. The CNTs are modelled as embedded beam elements assuming fully bonded interfacial conditions [137], with properties listed in Table 6.1. The finite element model is created using the random RVE generator described in Section 6.2.4 for 200 and 800 emdedded CNTs respectively. A unidirectional orientation

EBE proporties for CNT(8,8)	
CNT diameter	1.06 nm
CNT thickness	0.34 nm
Bending inertia X,Y	100.18 nm^4
Torsional constant	68.77 nm^4
Young's Modulus	4 GPa
Poisson's ratio	0.4

Table 6.1: CNT Properties.

of CNTs is considered with the principal direction being the axis X, as illustrated in Figure 6.4

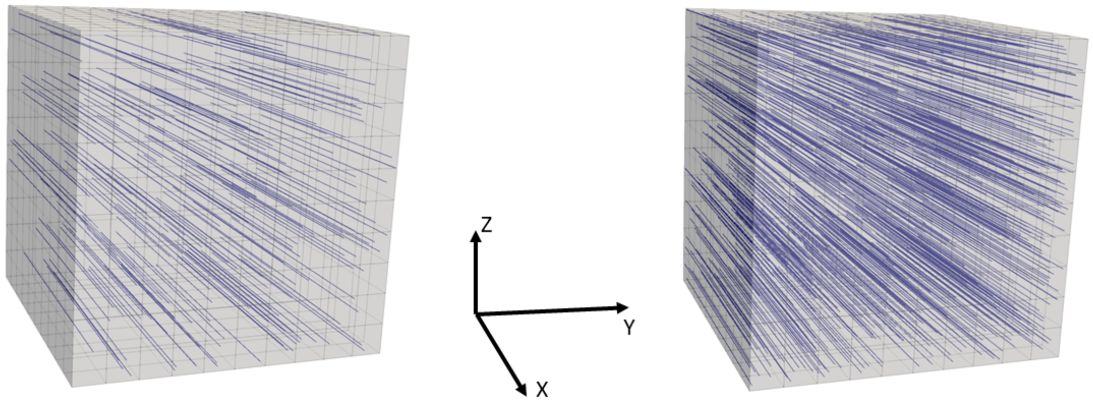


Figure 6.4: Representative volume elements with embedded CNTs a) 2.5% embedded CNTs volume fraction b)10% embedded CNTs volume fraction.

The length of the CNTs is considered to be 100 nm while its position in the YZ plane is given by two uncorrelated uniform variables fro coordinates Y_A and Z_A, as described in Section 6.2.4.

$$\mathbf{C}_{RVE_a} = \begin{bmatrix} 18.328 & 1.905 & -9.09E-18 \\ 1.905 & 4.762 & 5.80E-16 \\ 4.57E-16 & -5.01E-16 & 1.429 \end{bmatrix} \tag{6.35a}$$

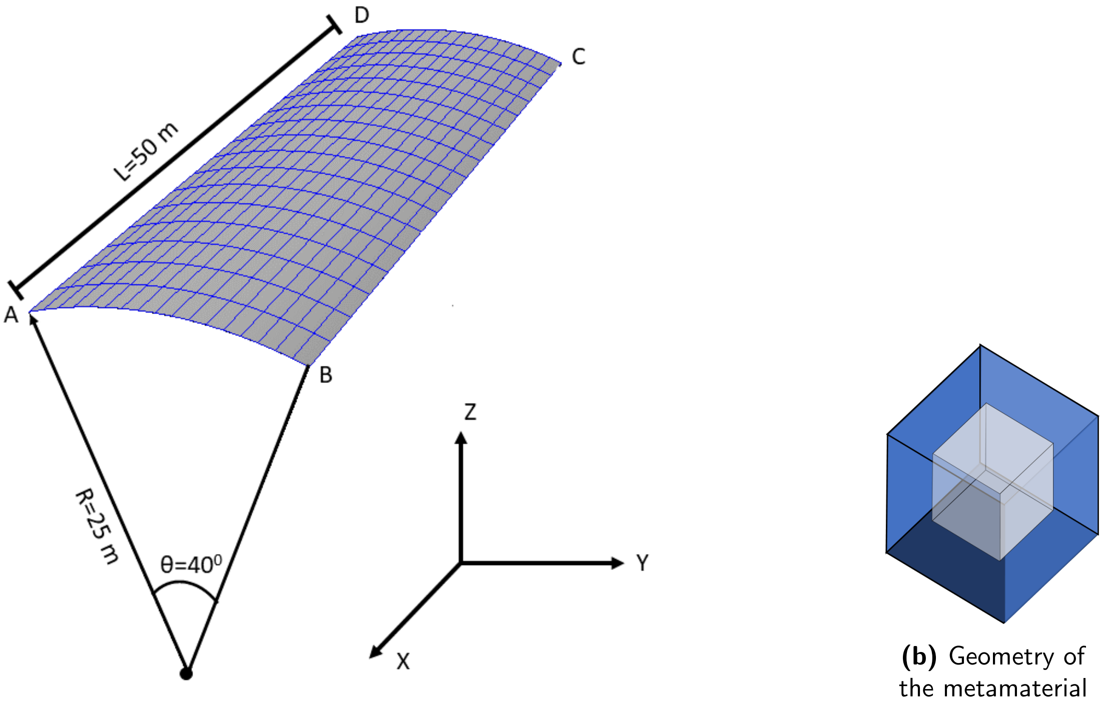
$$\mathbf{C}_{RVE_b} = \begin{bmatrix} 58.554 & 1.905 & 6.47E-16 \\ 1.905 & 4.762 & -3.18E-17 \\ 9.68E-17 & 1.15E-15 & 1.429 \end{bmatrix} \tag{6.35b}$$

The computed constitutive matrices for the two RVE case are preseneted in eq. 6.35. In comparison with the constitutive matrix of eq. 6.33, the reinforced RVE matrices provide a notable increase of stiffness in the X direction. Specifically, the principal component of axis X for the RVE reinforced with 2.5% VF of CNTs, is increased by 385% compared to the isotropic case, while for the 10% VF case an increase of 1230% is observed, thus making apparent the contribution of the reinforcing material in its mechanical performance. In both case, the zero terms of the isotropic elasticity matrix are close to zero within computer accuracy and thus the extracted constitutive matrices are in agreement with the plane stress matrix.

6.3.2 Cylindrical shell

The second example is the benchmark example of the cylindrical shell shown in Figure 6.5a. Its cross-section is a circular arc of 25 meters with its angle being 40^o. The thickness of the shell is 0.25m and the total length of the structure is 50 m. The shell is subjected to a gravitational load of magnitude $f_0 = 90N/m^2$ per unit surface [119, 164]. The boundary conditions of the structure are the following: translational degrees of freedom per axis X and Z are constrained in edge AB and CD.

The material considered in this example is the metamaterial of Figure 6.5b with periodic void. A unit cell of 100x100x100 nm with a void of 50x50x50nm is used to construct the 3D RVE for this case. The modulus of elasticity of the RVE material is considered to be a random variable assumed to follow the lognormal distribution with mean value E=4 GPa and standard deviation 0.5 GPa. Two spatial correlation cases were considered. In the first case, a through thickness variation of the modulus of elasticity was considered in addition to the surface variability. The modulus of elasticity



(a) Geometry of the shell example.

(b) Geometry of the metamaterial used as an RVE

Figure 6.5: Geometries of the cylindrical shell example.

in this case is described as a white noise random field, according to eq. 6.26.

20.000 Monte Carlo simulations of the stochastic cylindrical shell are performed for each spatial variability case. Figure 6.6a, displays the histogram of the vertical displacements of monitor node in the middle of the edge BC for the first spatial variability case. As can be seen, the results follow a near-to normal distribution, with mean value of -0.038762m and a standard deviation of 1.3428e-04m. In the white noise spatial variability case the histogram of displacements versus probability is displayed in Figure 6.6b, where the mean value of the results is -0.0791m and their standard deviation 0.0233m.

Inspection of these results immediately indicates the importance of the multiscale modelling proposed in this study. Considering a constant material for the macroscale model, gives a standard deviation of the displacements equivalent to 30%, i.e. COV=0.3 of the resulting mean value, while in the white noise case the COV reduces to 0.18. In addition, the shapes of the distributions differ significantly with the first (Figure 6.6a) being near to Gaussian and the second being near to lognormal, thus in alignment to

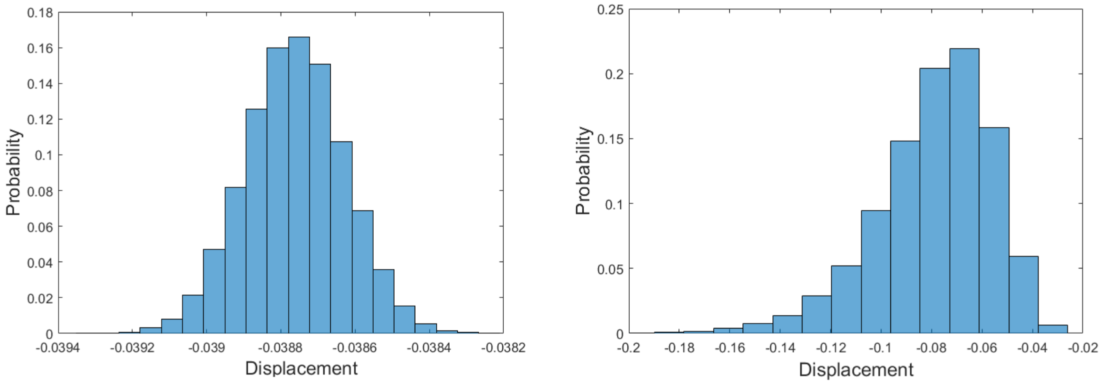


Figure 6.6: Histograms for the vertical displacements of monitor Control Point C.

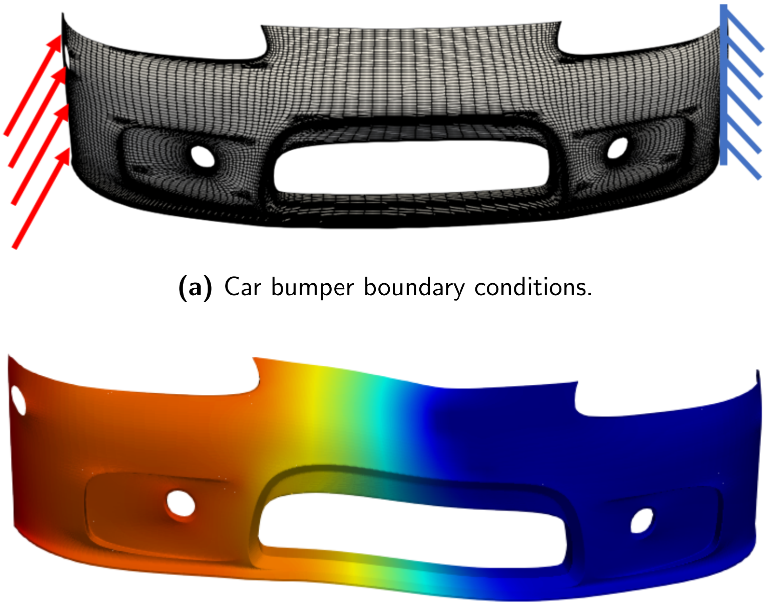
the input random variables.

6.3.3 Car composite bumper example

The next example is a real-scale model of a car bumper illustrated in Figure 6.7. The model is retrieved from [Tur]. Utilizing T-Spline plugin for Rhino [165],the CAD surface is transformed to T-Splines surface. By extracting a Bezier mesh, an analysis of the bumper is performed in MSolve platform [MSo]. Details of the bumper boundary conditions and 6.7a and Table 6.2, respectively. A front view of the bumper is given in Figure 6.7. The right edge of the bumper is considered clamped, while loading of 100 KN is applied at the Control Points of the left edge. The deformed bumper configuration is given in Figure 6.7b for an isotropic material with E=4 GPa modulus of elasticity and Poisson's ratio $\nu = 0.4$.

Assuming that the bumper is made of a polymer matrix reinforced with CNTs, the multiscale analysis described in the previous section is applied. In this context, a cubic RVE with dimensions of 100x100x100 nm is constructed using the RVE generator of Section 6.2.4. The RVE is comprised of a poly-ether-ether-ketone (PEEK) [109] matrix with Young's modulus E=4 GPa and Poisson's ratio $\nu = 0.4$. The matrix is reinforced with (8,8) armchair CNTs that are modelled as beam elements [137](Figure 6.8) embedded in the surrounding matrix assuming fully bonded interfacial conditions, as described in Section 6.2.4.

Both matrix and CNTs are considered linear elastic. The matrix is discretized with 1000 hexahedral elements, while the CNTs are modelled utilizing Equivalent



(a) Car bumper boundary conditions.

(b) Car bumper deformed configuration for unreinforced polymer material.

Figure 6.7: Histograms for the vertical displacements of monitor Control Point C.

Beam Elements (EBE). The EBEs are extracted from detailed Molecular Structural Mechanics (MSM) models of the CNTs [137]. As shown in Figure 6.8, a CNT portion is modelled as a space frame lattice according to the MSM formulation and then projected to an equivalent surrogate beam model with much lower degrees of freedom. The full CNT is then modeleed as a series of EBE elements. Table 6.1 presents the mechanical properties of the EBEs that are extracted with the MSM approach for a CNT (8,8). A detailed description of this surrogate modelling procedure can be found in [137, 157].

Since nanotubes are in theory indefinately long cylinders organized in hexagonal lattice form, the notation (m,n) serves to define the atoms positioning on the circumference of the nanotube. In literature, the term (m,m) refers to armchair type nanotubes, whose paths are defined by two consecutive 60^o left turns, followed by two 60^o right turns repeated every four steps. Figure 6.9 provides an example of a graphene sheet that will serve as the cylindrical development of a CNT. The vectors c_1, c_2 depicted are considered the unit vectors based on a unit cell hexagon of the lattice. By combining **m** unit cells along c_1 direction and **n** along c_2 direction a (m,n) type nanotube is generated.

Model	
Control Points	23312
Bezier Elements	24664
Degrees of freedom	69936

Table 6.2: Details of the T-Spline model discretization.

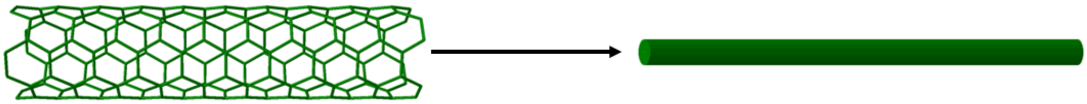


Figure 6.8: Schematic CNT representation a) MSM model of the CNT b) Beam element with equivalent propoerties to the detailed CNT model

A sensitivity analysis is initially performed to determine the sensitivity of the bumpers response to the CNTs VF. To this purpose, 10 RVEs are constructed with varying VFs in the range between 2.5-10%. The RVE matrix geometry is considered constant throughout the structure, while the CNTs are placed unidirectionally parallel to the X axis of the mesoscale model (see Figure 6.4) in the first subcase and randomly oriented in the second subcase (see Figure 6.10). The longitudinal direction coincides with a_1 local element axis as shown in Figure 6.1. Their position in the YZ plane is determined by the random RVE generator as described in section 6.2.4, with the coordinates Y_A and Z_A modeled with two uniformly distributed and uncorrelated random variables as

$$Y_A \sim U(0, 100nm) \tag{6.36a}$$

$$Z_A \sim U(0, 100nm) \tag{6.36b}$$

Figure 6.11, presents the results of the aforementioned sensititvity analysis. The horizontal axis represents the VF of the inclusions, while the vertical provides a measure of the stiffness increase of the structure expressed as a normalized displacement. This displacement is defined as the ratio of the maximum bumper displacement of the CNT reinforced bumper to the maximum displacement for the unreinforced bumper, at the same monitoring degree of freedom:

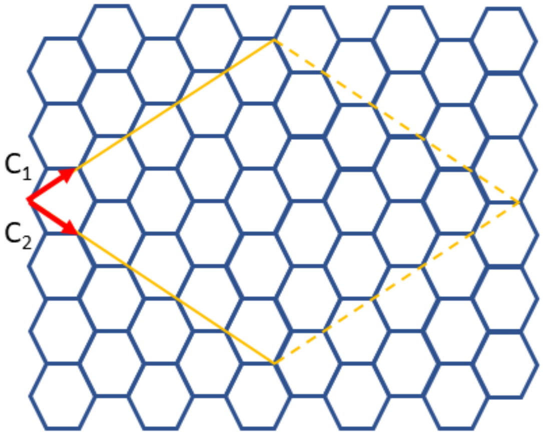
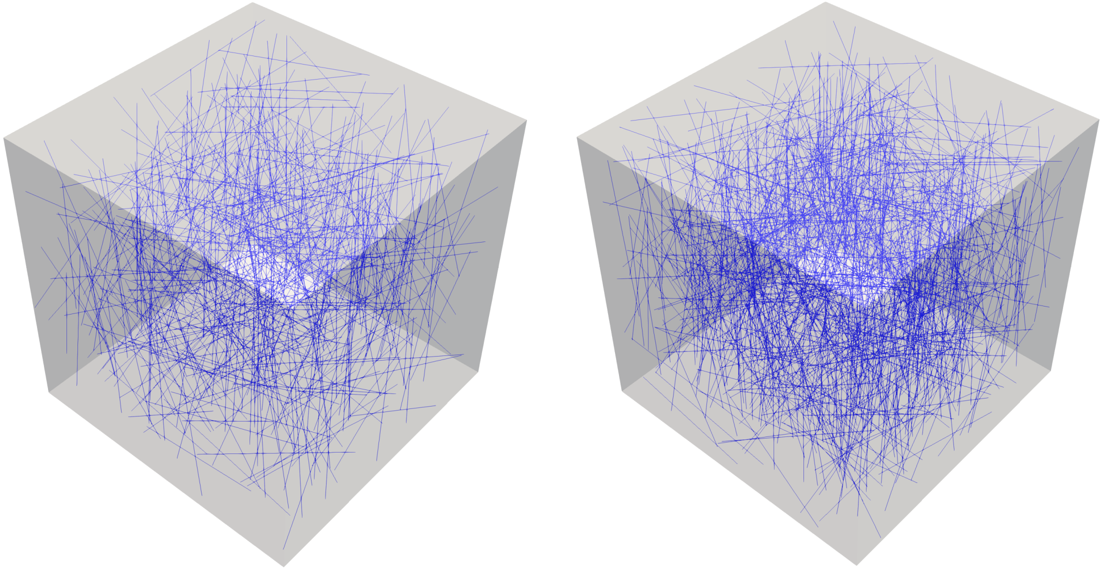


Figure 6.9: Graphene sheet honeycomb structure.

$$U(\%) = \frac{U_{CNT_{max}}}{U_{Matrix_{max}}} \tag{6.37}$$

As observed in Figure 6.37, the minimum volume fraction of 2.5% results in a maximum displacement U 0.55, that is approximately 55% of the corresponding displacement of the bumper with neat PEEK material. This reduction can reach a maximum U=30% in the case of 10% CNT volume fraction randomly dispersed in the matrix, revealing a70% stiffness increase of the bumper. In addition, it can be observed that the variation of U as a function of the weight fraction is nonlinear reaching towards a plateau beyond which the increase of stiffness for weight fractions larger than 10% becomes negligible. Inspection of the curves for longitudinal and randomly oriented CNTs presented in Figure 6.11, immediately emphasizes the importance of the proposed material microstructure modelling in revealing the sensitivity of the macroscale response to microstructural parameters. A significant stiffness increase is observed for the case of longitudinal CNTs. For example, for a stiffness increase of 50% with respect to the unreinforced material illustrated with a dashed line in Figure 6.11, a 3% VF of longitudinal CNTs is required, while the same increase is achieved with a 7% VF of randomly oriented CNTs This indicates that the CNT inclusions of a polymer matrix can be reduced up to 57% while meeting the desired stiffness requirements. In a similar fashion, for a given CNT VF of 4.3%, a 58% stiffness increase is observed for the case of longitudinal CNTs



(a) Volume fraction 3.5%

(b) Volume fraction 6%

Figure 6.10: Randomly distributed CNTs.

compared to 40% in case of the random oriented CNTs, as indicated by the dotted lines of Figure 6.11. This translates to a 31% deviation between the two cases, showcasing the importance of the detailed microstructure modelling. As a result, it is observed that the proposed nested IGA-FEM scheme, can quantitatively assess the influence of the material micrstructure to the final structural response, while at the same time providing crucial insight for the design of advanced composite materials.

The random RVE generator is utilized next, to introduce spatial and through thickness variability to the RVEs. Four cases for the reinforcement of the bumper are examined. In the first two cases, unidirectional CNTs are assummed parallel to the axis X of the mesoscale model(Figure 6.4), uniformly distributed inside the matrix according to eq. 6.36. For the latter two cases, the embedded CNTs have random orientations (Figure 6.10) given by eq. 6.26. In both cases, a Gaussian band limited white noise is assumed for the generation of a 3D random field, with two VF subcases of 3.5% and 6% respectively and a coefficient of variation of 1%. Table6.3, contains the results generated by 2.500 Monte Carlo simulations for each of the cases described above, while Figure 6.12 presents their respective histograms.Comparing unidirectional and randomly oriented

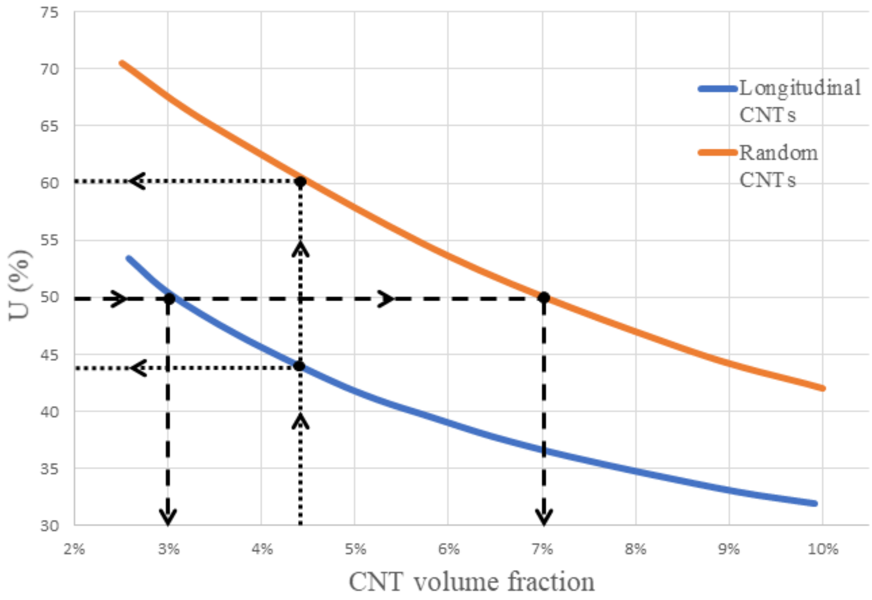


Figure 6.11: Impact of CNT volume fraction to the maximum deflection of the car bumper

CNTs for their two equivalent volume fraction cases, it can be observed that the random orientation of the CNTs has more impact on the performance of the bumper as indicated by the maximum displacement ratio which is significantly lower for the case of randomly oriented CNTs. Specifically, for the 3.5% volume fraction the mean ratio of 50% stiffness augmentation for the aligned CNTs is reduced to 35% for the randomly oriented and similarly for the 6% volume fraction case, the 60% stiffness increase is reduced to 46%. In both cases the impact of random orientation gives a compelling 25-30% reduced stiffness compared to the corresponding unidirectional positioning of the inclusions. From Figure 6.11 it can be seen that the stiffness of the nanocomposites exhibits a significant increase for CNT volume fractions ranging between 2.5-9%. In addition, the rate of the stiffness increase is slowing down for higher CNT volume fraction exhibiting a plateau behavior in which the stiffness increase becomes insignificant.

As in the previous example, inspection of the results immediately indicates the importance of the detailed multiscale modelling proposed in this study. As indicated in the examples considered, the modelling of the micromechanics can severely affect the constitutive response of the material, therefore the accurate description of the materials microstructure, as addressed by the proposed approach, becomes crucial. The inclusion's

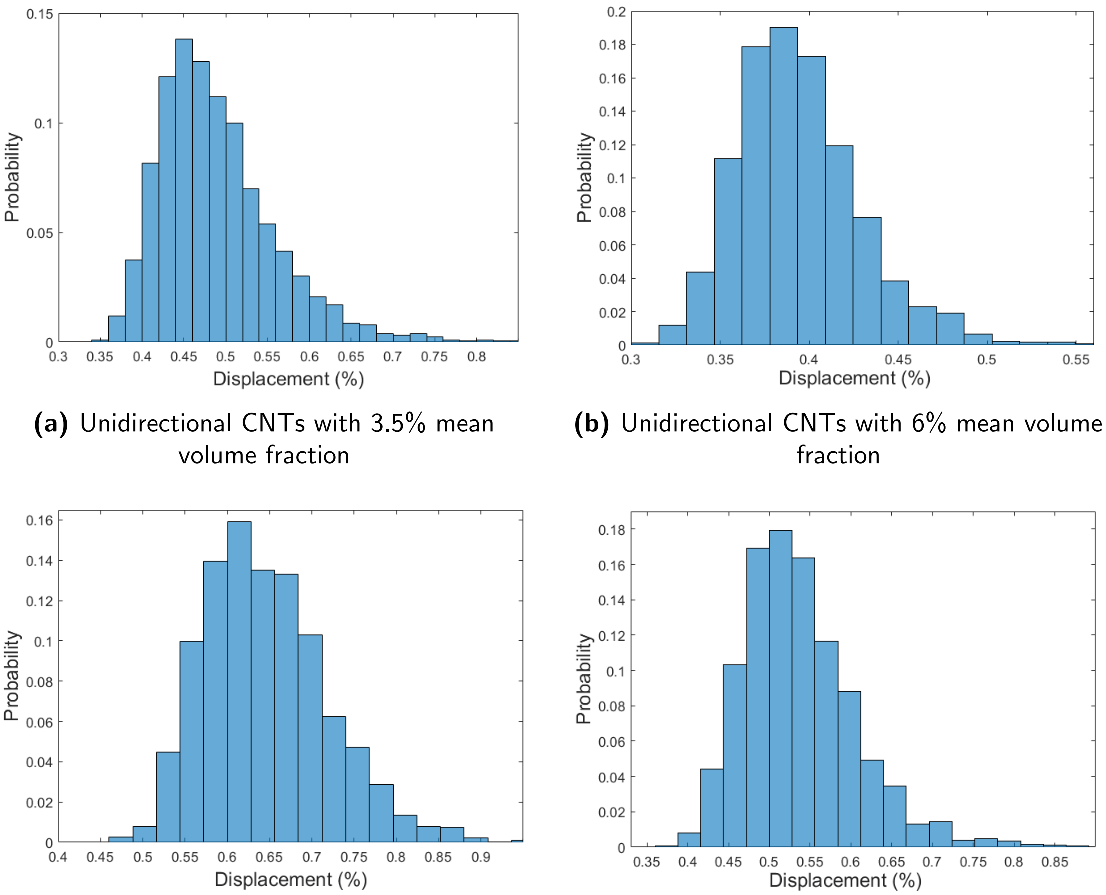
% Reduction	Lognitudinal	Lognitudinal	Random orientation	Random orientation
	CNTs 3.5% VF	CNTs 6% VF	CNTs 3.5% VF	CNTs 6% VF
Mean value	50.98%	60.53%	35.16%	46.44%
Standard Deviation	7.36%	3.55%	7.58%	7.04%

Table 6.3: Results for two volume fraction cases, for longitudinal and randomly oriented CNTs

volume fraction effect on the final material performance follows a non-linear law (Figure 6.11), while the randomness of the inclusion directionality can reduce the overall material performance up to 30%. The sensitivity of material and structural performance to the parameters assumed for the micromechanics modelling emphasizes the necessity of utilizing approaches like the one proposed in this work, for the accurate modeling of shell composites.

6.3.4 Parallel computer implementation

Since both the nested IGA-FEM semi-concurrent multiscale analysis and the Monte Carlo simulations require extensive computational workload, the developed code is parallelized with the aid the High performance computing (HPC) oriented platform MSolve[MSo] which implements the Task Parallel Library (TPL)[110] and MPI.NET [Mpi] ,which are the equivalents to the OpenMP[52] and MPI[White] frameworks for the C# programming language. The parallelization procedure is illustrated schematically in Figure 6.13 and is performed in two phases. Phase 1 includes the generation of the random RVE geometries, of multitude n for each macroscale model and the off-line computation of their equivalent properties. In this phase each computing node of the MPI environment is assigned to a number of RVE geometries. This number is equal to the independent integration points of the macroscale model. Then, finite element analyses are performed independently for each RVE utilizing the cores of the computing node in a shared memory computing environment. As a result, a collection of constitutive responses, linked to a specific RVE is created in each MPI node.



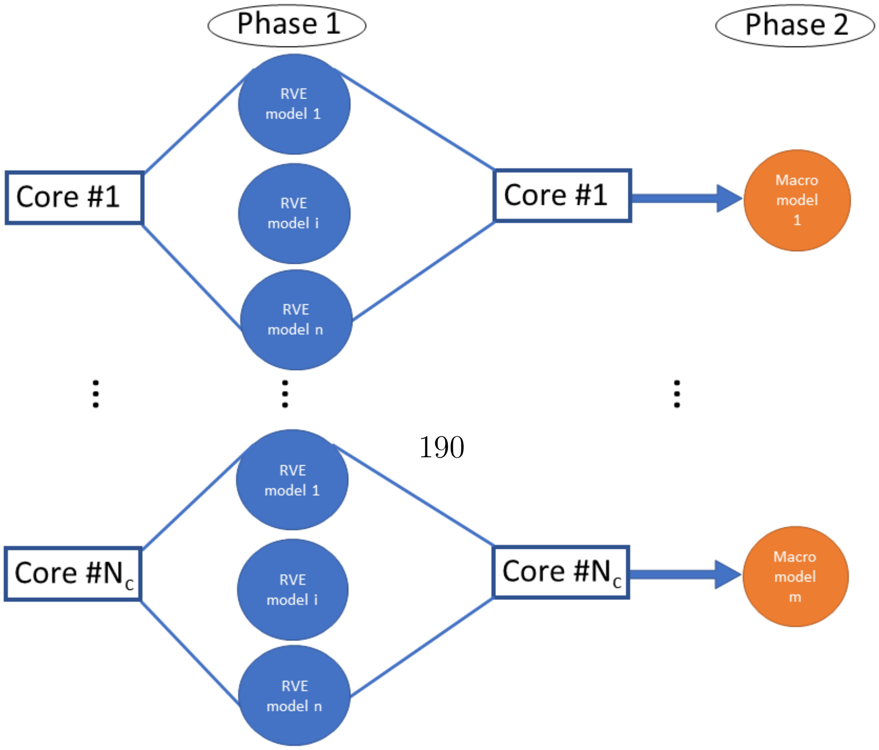
(a) Unidirectional CNTs with 3.5% mean volume fraction

(b) Unidirectional CNTs with 6% mean volume fraction

(c) Random oriented CNTs with 3.5% mean volume fraction

(d) Random oriented CNTs with 6% mean volume fraction

Figure 6.12: Histogram for the displacement reduction vs its probability of occurence.



Following the generation of the required multitude of RVEs, a second parallelization phase (Phase 2) is implemented for the Monte Carlo simulations of m macroscale models. In this Phase a different mesoscale RVE model is assigned to each of the thickness integration points for each macroscale model generated by MCS and the corresponding constitutive matrix is retrieved from the formerly generated collection without additional calculation. The FEM analyses of the macroscale models are also independent and are performed in an embarrassingly parallel way as well. Specifically, each MPI computing node is assigned to macroscale models in which their corresponding RVEs are already computed in the node. As a result, there is no need for communication between the nodes to retrieve RVE data, thus maximizing the parallelization performance.

Analysis time	Degrees of freedom	Monte Carlo Simulations	Sequential time	Parallel (6 cores) time	Parallel (36 cores) time
Cylindrical shell	1.083	20.000	363min	64min	11min
Bumper	69.936	2.500	6.325min	1.351min	226min

Table 6.4: Comparison of sequential and parallel stochastic analysis times.

Table 6.4 shows the discretized degrees of freedom of the macroscale level shell models, the number of Monte Carlo simulations and the corresponding computing times required for examples 2 and 3 in both sequential and parallel computing implementations. The CPU used in in the single CPU case is an i7-980x with 6 cores. Specifically, the sequential time of 363 minutes is reduced to 96 minutes resulting in a speedup of 5.67 times, in the case of 20.000 Monte Carlo simulations of the cylindrical shell. Similarly, for 2.500 simulations of the CNT reinforced bumper, the sequential time is 6.325 minutes is reduced in the parallel implementation to 2.027 minutes resulting in a 4.68 times speedup. It is observed that for the bumper model the speedup is reduced compared to the linear speed up of the cylindrical shell example. This is attributed to the increased memory requirements of the bumper model, that do not allow for the full utilization of the CPU cores. In order to study the scalability of the proposed parallel implementation, both examples are computed on a cluster consisting of 6 computing nodes, containing a 6 core i7-980x processor each. For the cylindrical shell test case, the sequential time of 363 minutes is further reduced to 11 minutes thus achieving a 33 times faster execution.

Similarly, for the car bumper case, the sequential time of 6.325 minutes is reduced to 226 minutes achieving a 28 times faster execution of the MCS simulations. It is observed that in both the single node and multi-node cases the parallelization scheme yields a close to linear speedup and thus significantly reduces the required execution time to tractable computational timeframes.

7

Summary - Innovation of thesis

This thesis presented a numerical framework that addresses the efficient implementation of isogeometric method in real-scale mechanical applications. With the main focus being on the introduction of efficient solution schemes, existing state-of-the-art solution schemes were investigated and efficient alternatives were introduced that alleviate the computational cost for the solution of the resulting linear systems in both symmetric and non-symmetric cases.

Specifically, a family of solution algorithms was proposed for the purposes of isogeometric Galerkin methods that exploit the advantages of an iterative solution scheme coupled with domain decomposition methods. Due to the enhanced continuity introduced by CAD shape functions, a considerable increase in the bandwidth and population of the resulting matrices is observed. This introduced a significant computational overhead to existing DDM. Especially, in the case of isogeometric tearing and interconnecting method (IETI), its main deficit is that the subdivisioning is geometry dependent, thus producing patches of irregular shape and size. Hence, IETI is rendered inefficient in terms of performance, as load balancing is impaired due to the irregularities of the patches. To address these shortcomings, two IETI variants are examined in this work, namely multi-patch IETI, called IETI-P and overlapping IETI, called IETI-O and a non-overlapping version PCG-IETI-N, where in all cases subdivisioning is independent of the patch structure, thus enabling large-scale computations. IETI-P version is basically the IETI method, with the introduction of an appropriate number of patches that ensure proper load balancing of the resulting subdomains. IETI-O on the other hand proposes a way to subdivide the model, without requiring the subdomains to match the patches, at the expense of up 40% larger interface problems. This has a negative effect on the solution performance, yet exhibits improved performance when compared to IETI-P. Finally, the PCG-IETI-N method, exploits a non-overlapping decomposition of the domain in order to introduce a strong and scalable preconditioner for the PCG iterative solution scheme. This is attained by introducing a proper non-overlapping decomposition of the domain in the form of truncated shape functions. The process of shape function truncation introduces a non-overlapping version of the model, that has identical geometry and features yet properly induced discontinuities. For the solution of the interface problem among the non-overlapping subdomains, the IETI method is utilized. Numerical results indicate improvements in terms of both CPU time as well as iterations, rendering PCG-IETI-N method up to 2,4 times faster compared to its IETI counterpart. These improvements are expected to be more pronounced as the model gets larger.

In a similar fashion, the solution of linear systems stemming from isogeometric collocation problems is addressed. Existing solution schemes are limited to overlapping implementations of DDM that automatically rendered the solution inefficient due to the increased interface problem. At the same time, dual domain decomposition schemes such as IETI method cannot be applied to isogeometric collocations methods, as the duality of the unknown fields is no longer valid due to the non-symmetric nature of the resulting matrices. In order to alleviate the aforementioned deficiencies, a non-overlapping domain decomposition method was introduced as an efficient and scalable preconditioner to the GMRES solver. This was achieved by a partitioning of the initial stiffness matrix into multiple subdomains. Compared to its overlapping competitors and namely the overlapping Additive Schwarz method, the proposed solution scheme indicated superior performance in terms of both iteration count and CPU time with an up to 5x times faster execution being attained.

Finally, the efficient implementation of isogeometric shells structures is addressed under the prism of semi-concurrent multiscale material modelling. Existing isogeometric thin shell formulations, employing Kirchhoff-Love shell theory, are extended to incorporate detailed modelling of advanced composites. A nested IGA-FEM multiscale analysis scheme is proposed, in which IGA is used for the discretization of the macroscopic level and FEM for the discretization of the corresponding RVE. As a result, the plane stress material required for the integration of the shell stiffness is derived via a computational homogenization procedure, thus taking into account the microstructural material topology. Numerical results, showcase the effect of detailed material modeling to the overall mechanical performance of the shell structure, as well as the applicability of the proposed formulation to real life applications

References

[MSo] Mgroup.

[Mpi] Mpi.net: High-performance c# library for message passing.

[Tur] Turbosquid front bumper.

[4] (2009a). *Linear Elasticity*, chapter 4, (pp. 109–147). John Wiley & Sons, Ltd.

[5] (2009b). *NURBS as a Basis for Analysis: Linear Problems*, chapter 3, (pp. 69–107). John Wiley & Sons, Ltd.

[6] Adam, C., Hughes, T., Bouabdallah, S., Zarroug, M., & Maitournam, H. (2015). Selective and reduced numerical integrations for nurbs-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284, 732 – 761. Isogeometric Analysis Special Issue.

[7] Ambati, M., Kiendl, J., & De Lorenzis, L. (2018). Isogeometric kirchhoff–love shell formulation for elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering*, 340, 320 – 339.

[8] Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., & Koster, J. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1), 15–41.

[9] Apostolatos, A., Schmidt, R., Wüchner, R., & Bletzinger, K.-U. (2014). A nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 97(7), 473–504.

[10] Arash, B., Park, H. S., & Rabczuk, T. (2015). Tensile fracture behavior of short carbon nanotube reinforced polymer composites: A coarse-grained model. *Composite Structures*, 134, 981 – 988.

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

[12] Auricchio, F., Beirão da Veiga, L., Hughes, T., Reali, A., & Sangalli, G. (2012b). Isogeometric collocation for elastostatics and explicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 249-252, 2 – 14. Higher Order Finite Element and Isogeometric Methods.

[13] Auricchio, F., Beirão da Veiga, L., Hughes, T., Reali, A., & Sangalli, G. (2012c). Isogeometric collocation for elastostatics and explicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 249-252, 2 – 14. Higher Order Finite Element and Isogeometric Methods.

[14] Auricchio, F., Beirão da Veiga, L., Kiendl, J., Lovadina, C., & Reali, A. (2013). Locking-free isogeometric collocation methods for spatial timoshenko rods. *Computer Methods in Applied Mechanics and Engineering*, 263, 113–126. cited By 93.

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

[16] Auricchio, F., Calabrò, F., Hughes, T., Reali, A., & Sangalli, G. (2012e). A simple algorithm for obtaining nearly optimal quadrature rules for nurbs-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249-252, 15 – 27. Higher Order Finite Element and Isogeometric Methods.

[17] Auricchio, F., Da Veiga, L., Kiendl, J., Lovadina, C., & Reali, A. (2016). Isogeometric collocation mixed methods for rods. *Discrete and Continuous Dynamical Systems - Series S*, 9(1), 33–42. cited By 3.

[18] AURICCHIO, F., DA VEIGA, L. B., HUGHES, T. J. R., REALI, A., & SANGALLI, G. (2010a). Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, 20(11), 2075–2107.

[19] AURICCHIO, F., DA VEIGA, L. B., HUGHES, T. J. R., REALI, A., & SANGALLI, G. (2010b). Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, 20(11), 2075–2107.

[20] Balduzzi, G., Morganti, S., Auricchio, F., & Reali, A. (2017). Non-prismatic timoshenko-like beam model: Numerical solution via isogeometric collocation. *Computers and Mathematics with Applications*, 74(7), 1531–1541. cited By 19.

[21] Bazilevs, Y. & Akkerman, I. (2010). Large eddy simulation of turbulent taylor–couette flow using isogeometric analysis and the residual-based variational multiscale method. *Journal of Computational Physics*, 229(9), 3402 – 3414.

[22] Bazilevs, Y., Calo, V. M., Hughes, T. J. R., & Zhang, Y. (2008). Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43(1), 3–37.

[23] Bazilevs, Y., Gohean, J., Hughes, T., Moser, R., & Zhang, Y. (2009). Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198(45), 3534 – 3550. Models and Methods in Computational Vascular and Cardiovascular Mechanics.

[24] Beirão da Veiga, L., Cho, D., Pavarino, L., & Scacchi, S. (2013). Isogeometric schwarz preconditioners for linear elasticity systems. *Computer Methods in Applied Mechanics and Engineering*, 253, 439 – 454.

[25] Beirão da Veiga, L., Cho, D., Pavarino, L., & Scacchi, S. (2014a). Overlapping schwarz preconditioners for isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 278, 239 – 253.

[26] Beirão da Veiga, L., Cho, D., Pavarino, L., & Scacchi, S. (2014b). Overlapping schwarz preconditioners for isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 278, 239 – 253.

[27] BEIRÃO DA VEIGA, L., CHO, D., PAVARINO, L. F., & SCACCHI, S. (2013). Bddc preconditioners for isogeometric analysis. *Mathematical Models and Methods in Applied Sciences*, 23(06), 1099–1142.

[28] Beirão da Veiga, L., Lovadina, C., & Reali, A. (2012). Avoiding shear locking for the timoshenko beam problem via isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 241-244, 38 – 51.

[29] Beirão da Veiga, L., Lovadina, C., & Reali, A. (2012). Avoiding shear locking for the timoshenko beam problem via isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 241-244, 38–51. cited By 113.

[30] Belgaid, H. & Bouazzouni, A. (2018). Vibration analysis of mechanical structures with a new formulation of the isogeometric collocation method. *European Journal of Mechanics, A/Solids*, 68, 88–103. cited By 3.

[31] Bellis, M. & Artioli, E. (2016). Nurbs-based collocation methods for the structural analysis of shells of revolution. *Metals*, 6(3). cited By 0.

[32] Bitoulas, N. & Papadrakakis, M. (1994). An optimized computer implementation of incomplete cholesky factorization. *Computing Systems in Engineering*, 5(3), 265 – 274.

[33] Bitzarakis, S., Papadrakakis, M., & Kotsopulos, A. (1997). Parallel solution techniques in computational structural mechanics. *Computer Methods in Applied Mechanics and Engineering*, 148(1), 75 – 104.

[34] Borden, M. J., Scott, M. A., Evans, J. A., & Hughes, T. J. R. (2011). Isogeometric finite element data structures based on bézier extraction of nurbs. *International Journal for Numerical Methods in Engineering*, 87(1-5), 15–47.

[35] Brivadis, E., Buffa, A., Wohlmuth, B., & Wunderlich, L. (2015a). Isogeometric mortar methods. *Computer Methods in Applied Mechanics and Engineering*, 284, 292 – 319. Isogeometric Analysis Special Issue.

[36] Brivadis, E., Buffa, A., Wohlmuth, B., & Wunderlich, L. (2015b). Isogeometric mortar methods. *Computer Methods in Applied Mechanics and Engineering*, 284, 292 – 319. Isogeometric Analysis Special Issue.

[37] Bézier, P. (1986). 1 - curves. In P. Bézier (Ed.), *The Mathematical Basis of the UNIURF CAD System* (pp. 3 – 22). Butterworth-Heinemann.

[38] Calabrò, F., Sangalli, G., & Tani, M. (2017). Fast formation of isogeometric galerkin matrices by weighted quadrature. *Computer Methods in Applied Mechanics and Engineering*, 316, 606 – 622. Special Issue on Isogeometric Analysis: Progress and Challenges.

[39] Casquero, H., Liu, L., Zhang, Y., Reali, A., Kiendl, J., & Gomez, H. (2017). Arbitrary-degree t-splines for isogeometric analysis of fully nonlinear kirchhoff–love shells. *Computer-Aided Design*, 82, 140 – 153. Isogeometric Design and Analysis.

[40] Casquero, H., Wei, X., Toshniwal, D., Li, A., Hughes, T. J., Kiendl, J., & Zhang, Y. J. (2020). Seamless integration of design and kirchhoff–love shell analysis using analysis-suitable unstructured t-splines. *Computer Methods in Applied Mechanics and Engineering*, 360, 112765.

[41] Chan, T. F. & Wan, W. L. (1997). Analysis of projection methods for solving linear systems with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 18(6), 1698–1721.

[42] Chang, G. & Wu, J. (1981). Mathematical foundations of bézier's technique. *Computer-Aided Design*, 13(3), 133 – 136.

[43] Charawi, L. A. (2017). Isogeometric overlapping schwarz preconditioners for the bidomain reaction–diffusion system. *Computer Methods in Applied Mechanics and Engineering*, 319, 472 – 490.

[44] Chen, T., Mo, R., & Wan, N. (2012). Imposing dirichlet boundary conditions with point collocation method in isogeometric analysis. *Jixie Gongcheng Xuebao/Journal of Mechanical Engineering*, 48(5), 157–164. cited By 2.

[45] Chen, X., Toh, K.-C., & Phoon, K.-K. (2006a). A modified ssor preconditioner for sparse symmetric indefinite linear systems of equations. *International Journal for Numerical Methods in Engineering*, 65, 785 – 807.

[46] Chen, X., Toh, K.-C., & Phoon, K.-K. (2006b). A modified ssor preconditioner for sparse symmetric indefinite linear systems of equations. *International Journal for Numerical Methods in Engineering*, 65, 785 – 807.

[47] Cho, D. (2020a). Optimal multilevel preconditioners for isogeometric collocation methods. *Mathematics and Computers in Simulation*, 168, 76 – 89.

[48] Cho, D. (2020b). Optimal multilevel preconditioners for isogeometric collocation methods. *Mathematics and Computers in Simulation*, 168, 76 – 89.

[49] Collier, N., Dalcin, L., Pardo, D., & Calo, V. M. (2012a). The cost of continuity: performance of iterative solvers on isogeometric finite elements.

[50] Collier, N., Dalcin, L., Pardo, D., & Calo, V. M. (2013). The cost of continuity: Performance of iterative solvers on isogeometric finite elements. *SIAM Journal on Scientific Computing*, 35(2), A767–A784.

[51] Collier, N., Pardo, D., Dalcin, L., Paszynski, M., & Calo, V. (2012b). The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers. *Computer Methods in Applied Mechanics and Engineering*, 213-216, 353 – 361.

[52] Costa, J., Cortes, T., Martorell, X., Ayguade, E., & Labarta, J. (2006). Running openmp applications efficiently on an everything-shared sdsm. *Journal of Parallel and Distributed Computing*, 66(5), 647 – 658. IPDPS '04 Special Issue.

[53] Cottrell, J., Reali, A., Bazilevs, Y., & Hughes, T. (2006). Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41), 5257 – 5296. John H. Argyris Memorial Issue. Part II.

[54] COX, M. G. (1972). The Numerical Evaluation of B-Splines*. *IMA Journal of Applied Mathematics*, 10(2), 134–149.

[55] da Veiga, L. B., Cho, D., Pavarino, L. F., & Scacchi, S. (2012a). Overlapping schwarz methods for isogeometric analysis. *SIAM Journal on Numerical Analysis*, 50(3), 1394–1416.

[56] da Veiga, L. B., Cho, D., Pavarino, L. F., & Scacchi, S. (2012b). Overlapping schwarz methods for isogeometric analysis. *SIAM Journal on Numerical Analysis*, 50(3), 1394–1416.

[57] da Veiga, L. B., Pavarino, L. F., Scacchi, S., Widlund, O. B., & Zampini, S. (2014). Isogeometric bddc preconditioners with deluxe scaling. *SIAM Journal on Scientific Computing*, 36(3), A1118–A1139.

[58] de Boor, C. (1972). On calculating with b-splines. *Journal of Approximation Theory*, 6(1), 50 – 62.

[59] De Boor, C. (2001). *A practical guide to splines; rev. ed.* Applied mathematical sciences. Berlin: Springer.

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

[61] Deng, X., Korobenko, A., Yan, J., & Bazilevs, Y. (2015). Isogeometric analysis of continuum damage in rotation-free composite shells. *Computer Methods in Applied Mechanics and Engineering*, 284, 349 – 372. Isogeometric Analysis Special Issue.

[62] Donatelli, M., Garoni, C., Manni, C., Serra-Capizzano, S., & Speleers, H. (2016). Spectral analysis and spectral symbol of matrices in isogeometric collocation methods. *Mathematics of Computation*, 85(300), 1639–1680. cited By 29.

[63] Dostál, Z., Horák, D., & Kučera, R. (2006a). Total feti—an easier implementable variant of the feti method for numerical solution of elliptic pde. *Communications in Numerical Methods in Engineering*, 22(12), 1155–1162.

[64] Dostál, Z., Horák, D., & Kučera, R. (2006b). Total feti—an easier implementable variant of the feti method for numerical solution of elliptic pde. *Communications in Numerical Methods in Engineering*, 22(12), 1155–1162.

[65] Evans, J., Hiemstra, R., Hughes, T., & Reali, A. (2018). Explicit higher-order accurate isogeometric collocation methods for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 338, 208–240. cited By 25.

[66] Fahrendorf, F., De Lorenzis, L., & Gomez, H. (2018a). Reduced integration at superconvergent points in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 328, 390–410. cited By 23.

[67] Fahrendorf, F., De Lorenzis, L., & Gomez, H. (2018b). Reduced integration at superconvergent points in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 328, 390 – 410.

[68] Farhat, C., Lesoinne, M., Letallec, P., Pierson, K., & Rixen, D. (2001a). Feti-dp: A dual-primal unified feti method part i: A faster alternative to the two-level feti method. *International Journal for Numerical Methods in Engineering*, 50(7), 1523–1544. cited By 415.

[69] Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., & Rixen, D. (2001b). Feti-dp: a dual–primal unified feti method—part i: A faster alternative to the two-level feti method. *International Journal for Numerical Methods in Engineering*, 50(7), 1523–1544.

[70] Farhat, C., Mandel, J., & Roux, F. X. (1994). Optimal convergence properties of the feti domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 115(3), 365 – 385.

[71] Farhat, C. & Roux, F.-X. (1991). A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6), 1205–1227.

[72] Farin, G. (1993). Chapter 1 - p. bézier: How a simple system was born. In G. Farin (Ed.), *Curves and Surfaces for Computer-Aided Geometric Design (Third Edition)* (pp. 1 – 12). Boston: Academic Press, third edition edition.

[73] Feng, Y., Owen, D., & Perić, D. (1995). A block conjugate gradient method applied to linear systems with multiple right-hand sides. *Computer Methods in Applied Mechanics and Engineering*, 127(1), 203 – 215.

[74] Ferguson, J. (1964). Multivariable curve interpolation. *J. ACM*, 11(2), 221–228.

[75] Forrest, A. (1990). Interactive interpolation and approximation by bézier polynomials. *Computer-Aided Design*, 22(9), 527 – 537.

[76] Fragakis, Y. & Papadrakakis, M. (2003). The mosaic of high performance domain decomposition methods for structural mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods. *Computer Methods in Applied Mechanics and Engineering*, 192(35), 3799 – 3830.

[77] Fragakis, Y. & Papadrakakis, M. (2004). The mosaic of high-performance domain decomposition methods for structural mechanics—part ii: Formulation enhance-

ments, multiple right-hand sides and implicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 193(42), 4611 – 4662.

[78] Ghanem, R. (2003). *Spanos, P.: Stochastic Finite Element: a Spectral Approach. Springer, New York*, volume 224.

[79] Gomez, H., Hughes, T. J., Nogueira, X., & Calo, V. M. (2010). Isogeometric analysis of the isothermal navier–stokes–korteweg equations. *Computer Methods in Applied Mechanics and Engineering*, 199(25), 1828 – 1840.

[80] Gomez, H., Reali, A., & Sangalli, G. (2014). Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal of Computational Physics*, 262, 153–171. cited By 63.

[81] Gordon, W. J. & Riesenfeld, R. F. (1974). Bernstein-bézier methods for the computer-aided design of free-form curves and surfaces. *J. ACM*, 21(2), 293–310.

[82] Herrema, A. J., Johnson, E. L., Proserpio, D., Wu, M. C., Kiendl, J., & Hsu, M.-C. (2019). Penalty coupling of non-matching isogeometric kirchhoff–love shell patches with application to composite wind turbine blades. *Computer Methods in Applied Mechanics and Engineering*, 346, 810 – 840.

[83] Hesch, C. & Betsch, P. (2012). Isogeometric analysis and domain decomposition methods. *Computer Methods in Applied Mechanics and Engineering*, 213-216, 104 – 112.

[84] Hill, R. (1972). On constitutive macro-variables for heterogeneous solids at finite strain. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 326(1565), 131–147.

[85] Hirschler, T., Bouclier, R., Duval, A., Elguedj, T., & Morlier, J. (2019). The embedded isogeometric kirchhoff–love shell: From design to shape optimization of non-conforming stiffened multipatch structures. *Computer Methods in Applied Mechanics and Engineering*, 349, 774 – 797.

[86] Hofer, C. & Langer, U. (2017a). Dual-primal isogeometric tearing and interconnecting solvers for multipatch dg-iga equations. *Computer Methods in Applied Mechanics and Engineering*, 316, 2 – 21. Special Issue on Isogeometric Analysis: Progress and Challenges.

[87] Hofer, C. & Langer, U. (2017b). Dual-primal isogeometric tearing and interconnecting solvers for multipatch dg-iga equations. *Computer Methods in Applied Mechanics and Engineering*, 316, 2 – 21. Special Issue on Isogeometric Analysis: Progress and Challenges.

[88] Hoschek, J., Lasser, D., & Schumaker, L. L. (1993). *Fundamentals of Computer Aided Geometric Design*. USA: A. K. Peters, Ltd.

[89] Hughes, T., Cottrell, J., & Bazilevs, Y. (2005). Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39), 4135 – 4195.

[90] Hughes, T., Reali, A., & Sangalli, G. (2008a). Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method nurbs. *Computer Methods in Applied Mechanics and Engineering*, 197(49), 4104 – 4124.

[91] Hughes, T., Reali, A., & Sangalli, G. (2008b). Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method nurbs. *Computer Methods in Applied Mechanics and Engineering*, 197(49), 4104 – 4124.

[92] Hughes, T., Reali, A., & Sangalli, G. (2010a). Efficient quadrature for nurbs-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5), 301 – 313. Computational Geometry and Analysis.

[93] Hughes, T., Reali, A., & Sangalli, G. (2010b). Efficient quadrature for nurbs-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5), 301 – 313. Computational Geometry and Analysis.

[94] Johannessen, K. A. (2017). Optimal quadrature for univariate and tensor product splines. *Computer Methods in Applied Mechanics and Engineering*, 316, 84 – 99. Special Issue on Isogeometric Analysis: Progress and Challenges.

[95] Kershaw, D. S. (1978a). The incomplete cholesky—conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics*, 26(1), 43 – 65.

[96] Kershaw, D. S. (1978b). The incomplete cholesky—conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics*, 26(1), 43 – 65.

[97] Kiendl, J., Auricchio, F., Beirão da Veiga, L., Lovadina, C., & Reali, A. (2015a). Isogeometric collocation methods for the reissner-mindlin plate problem. *Computer Methods in Applied Mechanics and Engineering*, 284, 489–507. cited By 58.

[98] Kiendl, J., Auricchio, F., & Reali, A. (2018). A displacement-free formulation for the timoshenko beam problem and a corresponding isogeometric collocation approach. *Meccanica*, 53(6), 1403–1413. cited By 8.

[99] Kiendl, J., Bletzinger, K.-U., Linhard, J., & Wüchner, R. (2009). Isogeometric shell analysis with kirchhoff–love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49), 3902 – 3914.

[100] Kiendl, J., Hsu, M.-C., Wu, M. C., & Reali, A. (2015b). Isogeometric kirchhoff–love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 291, 280 – 303.

[101] Kiendl, J., Marino, E., & Lorenzis, L. D. (2017). Isogeometric collocation for the reissner–mindlin shell problem. *Computer Methods in Applied Mechanics and Engineering*, 325, 645 – 665.

[102] Klawonn, A. & Widlund, O. B. (2006a). Dual-primal feti methods for linear elasticity. *Communications on Pure and Applied Mathematics*, 59(11), 1523–1572.

[103] Klawonn, A. & Widlund, O. B. (2006b). Dual-primal feti methods for linear elasticity. *Communications on Pure and Applied Mathematics*, 59(11), 1523–1572.

[104] Klawonn, A., Widlund, O. B., & Dryja, M. (2002a). Dual-primal feti methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM Journal on Numerical Analysis*, 40(1), 159–179.

[105] Klawonn, A., Widlund, O. B., & Dryja, M. (2002b). Dual-primal feti methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM Journal on Numerical Analysis*, 40(1), 159–179.

[106] Kleiss, S. K., Pechstein, C., Jüttler, B., & Tomar, S. (2012a). Ieti – isogeometric tearing and interconnecting. *Computer Methods in Applied Mechanics and Engineering*, 247-248, 201 – 215.

[107] Kleiss, S. K., Pechstein, C., Jüttler, B., & Tomar, S. (2012b). Ieti – isogeometric tearing and interconnecting. *Computer Methods in Applied Mechanics and Engineering*, 247-248, 201 – 215.

[108] Kouznetsova, V., Brekelmans, W., & Baaijens, F. (2001). Approach to micro-macro modeling of heterogeneous materials. *Computational Mechanics*, 27, 37–48.

[109] Kurtz, S. M. (2012). Chapter 1 - an overview of peek biomaterials. In S. M. Kurtz (Ed.), *PEEK Biomaterials Handbook*, Plastics Design Library (pp. 1 – 7). Oxford: William Andrew Publishing.

[110] Leijen, D., Schulte, W., & Burckhardt, S. (2009). The design of a task parallel library. In *Proceedings of the 24th ACM SIGPLAN Conference on Object Oriented Programming Systems Languages and Applications*, OOPSLA '09 (pp. 227–242). New York, NY, USA: Association for Computing Machinery.

[111] Leslie, W. H. P. (1973). Numerical control: Mathematics and applications, p. bézier, wiley, london, 1972. no. of pages: 240. price: £4.95. *International Journal for Numerical Methods in Engineering*, 6(3), 456–456.

[L'Hermite] L'Hermite, T. *Oeuvres complètes [Texte imprimé] / Tristan L'Hermite ; volume publié sous la direction de Roger Guichemerre avec la collaboration de Daniela Dalla Valle et Anne Tournon. V. Théâtre (suite), Plaidoyers historiques*. Sources classiques. Paris: H. Champion.

[113] Lin, H., Hu, Q., & Xiong, Y. (2013). Consistency and convergence properties of the isogeometric collocation method. *Computer Methods in Applied Mechanics and Engineering*, 267, 471–486. cited By 13.

[114] Lipski, P. & Paszynski, M. (2015a). Multi-frontal parallel direct solver for one dimensional isogeometric collocation method. *Computer Methods in Materials Science*, 15, 192–197.

[115] Lipski, P. & Paszynski, M. (2015b). Multi-frontal parallel direct solver for one dimensional isogeometric collocation method. *Computer Methods in Materials Science*, 15, 192–197.

[116] Lipski, P. & Paszyński, M. (2015). Multi-frontal parallel direct solver for one dimensional isogeometric collocation method. *Computer Methods in Materials Science*, Vol. 15, No.1, 192–197.

[117] Liu, Z., Cheng, J., Yang, M., Yuan, P., Qiu, C., Gao, W., & Tan, J. (2019). Isogeometric analysis of large thin shell structures based on weak coupling of substructures with unstructured t-splines patches. *Advances in Engineering Software*, 135, 102692.

[M] M, S. Nurbs toolbox.

[119] Macneal, R. H. & Harder, R. L. (1985). A proposed standard set of problems to test finite element accuracy. *Finite Elements in Analysis and Design*, 1(1), 3 – 20.

[120] Mandel, J. & Tezaur, R. (2000). On the convergence of a dual-primal substructuring method. *Numerische Mathematik*, 88.

[121] Mandel, J. & Tezaur, R. (2001). On the convergence of a dual-primal substructuring method. *Numerische Mathematik*, 88(3), 543–558. cited By 105.

[122] Manh, N. D., Evgrafov, A., Gersborg, A. R., & Gravesen, J. (2011). Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering*, 200(13), 1343 – 1353.

[123] Mantzaflaris, A. & Jüttler, B. (2015). Integration by interpolation and look-up for galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284, 373 – 400. Isogeometric Analysis Special Issue.

[124] Marino, E. (2016). Isogeometric collocation for three-dimensional geometrically exact shear-deformable beams. *Computer Methods in Applied Mechanics and Engineering*, 307, 383–410. cited By 28.

[125] Marino, E. (2017). Isogeometric collocation method for geometrically exact timoshenko beams. volume 2 (pp. 1647–1656). cited By 0.

[126] Marino, E., Kiendl, J., & De Lorenzis, L. (2019). Isogeometric collocation for implicit dynamics of three-dimensional beams undergoing finite motions. *Computer Methods in Applied Mechanics and Engineering*, 356, 548–570. cited By 6.

[127] Markou, G. & Papadrakakis, M. (2012). An efficient generation method of embedded reinforcement in hexahedral elements for reinforced concrete simulations. *Advances in Engineering Software*, 45(1), 175 – 187.

[128] Maurin, F., Greco, F., Coox, L., Vandepitte, D., & Desmet, W. (2018a). Isogeometric collocation for kirchhoff–love plates and shells. *Computer Methods in Applied Mechanics and Engineering*, 329, 396–420. cited By 17.

[129] Maurin, F., Greco, F., Dedoncker, S., & Desmet, W. (2018b). Isogeometric analysis for nonlinear planar kirchhoff rods: Weighted residual formulation and collocation of the strong form. *Computer Methods in Applied Mechanics and Engineering*, 340, 1023–1043. cited By 13.

[130] Metsis, P. & Papadrakakis, M. (2012). Overlapping and non-overlapping domain decomposition methods for large-scale meshless efg simulations. *Computer Methods in Applied Mechanics and Engineering*, 229-232, 128 – 141.

[131] Montardini, M., Sangalli, G., & Tamellini, L. (2017). Optimal-order isogeometric collocation at galerkin superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 316, 741–757. cited By 30.

[132] Nguyen, L. H. & Schillinger, D. (2017). A collocated isogeometric finite element method based on gauss–lobatto lagrange extraction of splines. *Computer Methods in Applied Mechanics and Engineering*, 316, 720 – 740. Special Issue on Isogeometric Analysis: Progress and Challenges.

[133] Nielsen, P. N., Gersborg, A. R., Gravesen, J., & Pedersen, N. L. (2011). Discretizations in isogeometric analysis of navier–stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 200(45), 3242 – 3253.

[134] Nitti, A., Kiendl, J., Reali, A., & de Tullio, M. D. (2020). An immersed-boundary/isogeometric method for fluid–structure interaction involving thin shells. *Computer Methods in Applied Mechanics and Engineering*, 364, 112977.

[135] Nordanger, K., Rasheed, A., Okstad, K. M., Kvarving, A. M., Holdahl, R., & Kvamsdal, T. (2016). Numerical benchmarking of fluid–structure interaction: An isogeometric finite element approach. *Ocean Engineering*, 124, 324 – 339.

[136] Oliveira, H. L., de Castro e Andrade, H., & Leonel, E. D. (2020). An isogeometric boundary element approach for topology optimization using the level set method. *Applied Mathematical Modelling*, 84, 536 – 553.

[137] Papadopoulos, V., Soimiris, G., Giovanis, D., & Papadrakakis, M. (2018). A neural network-based surrogate model for carbon nanotubes with geometric nonlinearities. *Computer Methods in Applied Mechanics and Engineering*, 328, 411 – 430.

[138] Papadrakakis, M. (1993). Solving large-scale linear problems in solid and structural mechanics.

[139] Pavan, G. & Nanjunda Rao, K. (2017). Bending analysis of laminated composite plates using isogeometric collocation method. *Composite Structures*, 176, 715–728. cited By 13.

[140] Phoon, K., Toh, K., Chan, S., & Lee, F. (2002a). An efficient diagonal preconditioner for finite element solution of biot's consolidation equations. *International Journal for Numerical Methods in Engineering*, 55(4), 377–400. cited By 57.

[141] Phoon, K. K., Toh, K. C., Chan, S. H., & Lee, F. H. (2002b). An efficient diagonal preconditioner for finite element solution of biot's consolidation equations. *International Journal for Numerical Methods in Engineering*, 55(4), 377–400.

[142] Piegl, L. & Tiller, W. (1987). Curve and surface constructions using rational b-splines. *Computer-Aided Design*, 19(9), 485 – 498.

[143] Piegl, L. & Tiller, W. (1995). *The NURBS Book*. Berlin, Heidelberg: Springer-Verlag.

[144] Pigazzini, M., Kamensky, D., van Iersel, D., Alaydin, M., Remmers, J., & Bazilevs, Y. (2019). Gradient-enhanced damage modeling in kirchhoff–love shells: Application to isogeometric analysis of composite laminates. *Computer Methods in Applied Mechanics and Engineering*, 346, 152 – 179.

[145] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1993). *Numerical Recipes in FORTRAN; The Art of Scientific Computing.* USA: Cambridge University Press, 2nd edition.

[146] PRZEMIENIECKI, J. S. (1963). Matrix structural analysis of substructures. *AIAA Journal*, 1(1), 138–147.

[147] Qian, X. (2010). Full analytical sensitivities in nurbs based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199(29), 2059 – 2071.

[148] Qian, X. & Sigmund, O. (2011). Isogeometric shape optimization of photonic crystals via coons patches. *Computer Methods in Applied Mechanics and Engineering*, 200(25), 2237 – 2255.

[149] Ramshaw, L. (1987). Blossoming: a connect-the- dots approach to splines.

[150] Reali, A. & Gomez, H. (2015). An isogeometric collocation approach for bernoulli-euler beams and kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering*, 284, 623–636. cited By 80.

[151] Reali, A. & Hughes, T. J. R. (2015). *An Introduction to Isogeometric Collocation Methods*, (pp. 173–204). Springer Vienna: Vienna.

[152] Rixen, D. J. & Farhat, C. (1999). A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *International Journal for Numerical Methods in Engineering*, 44(4), 489–516.

[153] Ruess, M., Schillinger, D., Özcan, A. I., & Rank, E. (2014). Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269, 46 – 71.

[154] Saad, Y. (2003a). *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, second edition.

[155] Saad, Y. (2003b). *Iterative Methods for Sparse Linear Systems.* USA: Society for Industrial and Applied Mathematics, 2nd edition.

[156] Saad, Y. & Schultz, M. H. (1986). Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3), 856–869.

[157] Savvas, D., Papadopoulos, V., & Papadrakakis, M. (2012). The effect of interfacial shear strength on damping behavior of carbon nanotube reinforced composites. *International Journal of Solids and Structures*, 49(26), 3823 – 3837.

[158] Schenk, O., Wächter, A., & Hagemann, M. (2007). Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. *Computational Optimization and Applications*, 36(2), 321–341.

[159] Schillinger, D., Borden, M., & Stolarski, H. (2015). Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284, 583–610. cited By 50.

[160] Schillinger, D., Evans, J., Reali, A., Scott, M., & Hughes, T. (2013a). Isogeometric collocation: Cost comparison with galerkin methods and extension to adaptive hierarchical nurbs discretizations. *Computer Methods in Applied Mechanics and Engineering*, 267, 170–232. cited By 192.

[161] Schillinger, D., Evans, J. A., Reali, A., Scott, M. A., & Hughes, T. J. (2013b). Isogeometric collocation: Cost comparison with galerkin methods and extension to adaptive hierarchical nurbs discretizations. *Computer Methods in Applied Mechanics and Engineering*, 267, 170 – 232.

[162] Schoenberg, I. J. (1988). *Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions*, (pp. 3–57). Birkhäuser Boston: Boston, MA.

[163] Schuß, S., Dittmann, M., Wohlmuth, B., Klinkel, S., & Hesch, C. (2019). Multi-patch isogeometric analysis for kirchhoff–love shell elements. *Computer Methods in Applied Mechanics and Engineering*, 349, 91 – 116.

[164] Scordelis, A. C. & Lo, K. (1964). Computer analysis in cylindrical shells.

[165] Scott, M., Hughes, T. J. R., Sederberg, T. W., & Sederberg, M. T. (2014). An integrated approach to engineering design and analysis using the autodesk t-spline plugin for rhino3d.

[166] Scott, M. A. (2011). T-splines as a design-through-analysis technology.

[167] Scott, M. A., Borden, M. J., Verhoosel, C. V., Sederberg, T. W., & Hughes, T. J. R. (2011). Isogeometric finite element data structures based on bézier extraction of t-splines. *International Journal for Numerical Methods in Engineering*, 88(2), 126–156.

[168] Sederberg, T. W., Cardon, D. L., Finnigan, G. T., North, N. S., Zheng, J., & Lyche, T. (2004). T-spline simplification and local refinement. *ACM Trans. Graph.*, 23(3), 276–283.

[169] Sederberg, T. W., Finnigan, G. T., Li, X., Lin, H., & Ipson, H. (2008). Watertight trimmed nurbs. *ACM Trans. Graph.*, 27(3), 1–8.

[170] Sederberg, T. W., Zheng, J., Bakenov, A., & Nasri, A. (2003). T-splines and t-nurccs. *ACM Trans. Graph.*, 22(3), 477–484.

[171] Seo, Y.-D., Kim, H.-J., & Youn, S.-K. (2010). Isogeometric topology optimization using trimmed spline surfaces. *Computer Methods in Applied Mechanics and Engineering*, 199(49), 3270 – 3296.

[172] Stavroulakis, G., Giovanis, D. G., Papadrakakis, M., & Papadopoulos, V. (2014). A new perspective on the solution of uncertainty quantification and reliability analysis of large-scale problems. *Computer Methods in Applied Mechanics and Engineering*, 276, 627 – 658.

[173] Stavroulakis, G., Tsapetis, D., & Papadrakakis, M. (2018). Non-overlapping domain decomposition solution schemes for structural mechanics isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 341, 695 – 717.

[174] Tepole, A. B., Kabaria, H., Bletzinger, K.-U., & Kuhl, E. (2015). Isogeometric kirchhoff–love shell formulations for biological membranes. *Computer Methods in Applied Mechanics and Engineering*, 293, 328 – 347.

[175] Tiller, W. (1983). Rational b-splines for curve and surface representation. *IEEE Computer Graphics and Applications*, 3(6), 61–69.

[176] Trypakis, E. (2015). Hierarchical formulation of the isogeometric analysis method for structures.

[177] Tsapetis, D., Sotiropoulos, G., Stavroulakis, G., Papadopoulos, V., & Papadrakakis, M. A stochastic multiscale formulation for isogeometric composite kirchhoff-love shells. *Computer Methods in Applied Mechanics and Engineering.*

[178] Tsapetis, D., Stavroulakis, G., & Papadrakakis, M. Non-overlapping domain decomposition solution schemes for isogeometric collocation. *Computer Methods in Applied Mechanics and Engineering.*

[179] Veiga, L., Cho, D., Pavarino, L., & Scacchi, S. (2013). Isogeometric schwarz preconditioners for linear elasticity systems. *Computer Methods in Applied Mechanics and Engineering*, 253, 439 – 454.

[180] Versprille, K. J. (1975). Computer-aided design applications of the rational b-spline approximation form.

[181] Wall, W. A., Frenzel, M. A., & Cyron, C. (2008). Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33), 2976 – 2988.

[182] Weeger, O., Narayanan, B., De Lorenzis, L., Kiendl, J., & Dunn, M. (2017). An isogeometric collocation method for frictionless contact of cosserat rods. *Computer Methods in Applied Mechanics and Engineering*, 321, 361–382. cited By 15.

[183] Weeger, O., Narayanan, B., & Dunn, M. (2018). Isogeometric collocation for nonlinear dynamic analysis of cosserat rods with frictional contact. *Nonlinear Dynamics*, 91(2), 1213–1227. cited By 8.

[White] White, R. Message passing interface.

[185] Widlund, O. & Toselli, A. (2004). *Domain decomposition methods - algorithms and theory*, volume 34. Springer.

[186] Zareh, M. & Qian, X. (2019). Kirchhoff–love shell formulation based on triangular isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 347, 853 – 873.