



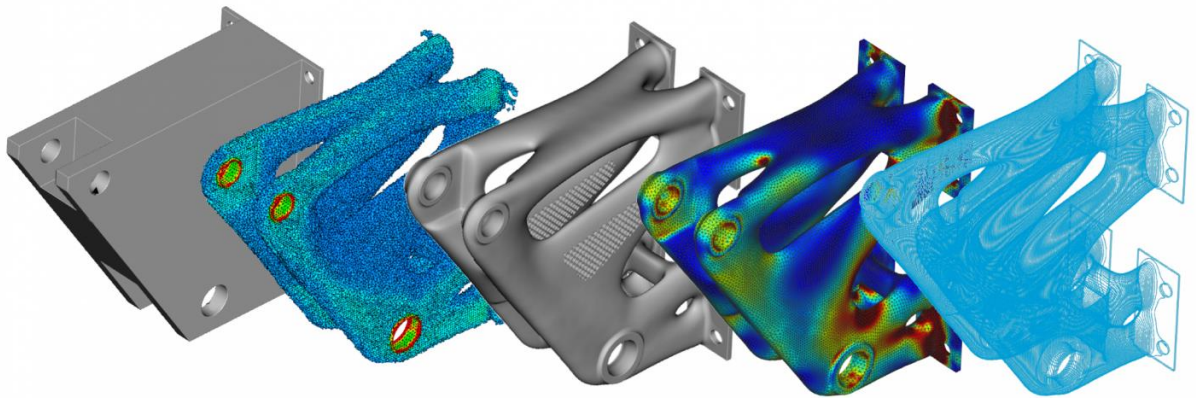
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Πολιτικών Μηχανικών

Τομέας Δομοστατικής

Εργαστήριο Στατικής & Αντισεισμικών Ερευνών

## Τοπολογική Βελτιστοποίηση με Ισογεωμετρικά Πεπερασμένα Στοιχεία



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δήμητρα Χ. Μουστόγιαννη

Επιβλέπων: **Βησσαρίων Παπαδόπουλος**, Αναπλ. Καθηγητής ΕΜΠ

Αθήνα, Ιούνιος 2020

Μουστόγιαννη Δ. Χ. (2020).  
Τοπολογική Βελτιστοποίηση με Ισογεωμετρικά Πεπερασμένα Στοιχεία  
Διπλωματική Εργασία  
Εργαστήριο Στατικής & Αντισεισμικών Ερευνών, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα.

Moustogianni D. C. (2020).  
Topology Optimization with Isogeometric Finite Elements  
Diploma Thesis  
Institute of Static Analysis & Antiseismic Research, National Technical University of  
Athens, Greece.

# Πίνακας περιεχομένων

Περίληψη .....	III
Abstract .....	IV
Ευχαριστίες .....	V
1 Εισαγωγή .....	1
1.1 Αντικείμενο Διπλωματικής Εργασίας .....	1
1.2 Δομή .....	1
2 Μέθοδος Πεπερασμένων Στοιχείων – Finite Elements Analysis (FEA).....	3
2.1 Εισαγωγή .....	3
2.2 Ιστορική Αναδρομή.....	4
2.3 Μαθηματικό Υπόβαθρο .....	5
2.3.1 Τάσεις και παραμορφώσεις.....	5
2.3.2 Αρχή Δυνατών Έργων.....	5
2.3.3 Σχέσεις ανηγμένων παραμορφώσεων – μετατοπίσεων.....	6
2.3.4 Σχέσεις τάσεων – ανηγμένων παραμορφώσεων .....	7
2.3.5 Συναρτήσεις σχήματος πεπερασμένου στοιχείου και πολυώνυμα παρεμβολής .....	8
2.4 Διαστάσεις πεπερασμένων στοιχείων φορέα .....	11
2.4.1 1-Dimensional Analysis .....	11
2.4.2 2-Dimensional Analysis .....	11
2.4.3 3-Dimensional Analysis .....	13
2.5 Ισοπαραμετρικά Πεπερασμένα Στοιχεία.....	13
2.6 Διαδικασία Υπολογισμού Μητρώου Στιβαρότητας.....	15
3 Ισογεωμετρική Ανάλυση – Isogeometric Analysis (IGA).....	21
3.1 Εισαγωγή.....	21
3.2 Ιστορική Εξέλιξη.....	23
3.3 Καμπύλες B-SPLines .....	25
3.3.1 Index, Parameter and Physical Space.....	26
3.3.2 Διάνυσμα κόμβων (Knot Value Vector) .....	29
3.3.3 Σημεία Ελέγχου (Control Points).....	29
3.3.4 Βασικές συναρτήσεις B-SPLines .....	30
3.3.5 Ιδιότητες βασικών συναρτήσεων B-SPLines.....	31
3.3.6 Ιδιότητες καμπύλων B-SPLines.....	32
3.3.7 Παράγωγοι των βασικών συναρτήσεων ( B-SPLine Basis Function Derivatives) .....	37
3.4 Μη ομοιόμορφες ρητές καμπύλες B-SPLines (Non Uniform Rational B-SPLines - NURBS).....	37
3.4.1 Συναρτήσεις σχήματος NURBS.....	38
3.4.2 Ιδιότητες καμπύλων NURBS .....	39
3.4.3 Ιδιότητες συναρτήσεων NURBS.....	39
3.4.4 Παράγωγοι των συναρτήσεων σχήματος NURBS.....	40
4 Μητρώο Στιβαρότητας .....	43
4.1 Βήματα για την μόρφωση του Μητρώου Στιβαρότητας .....	43
4.1.1 Συναρτήσεις Σχήματος.....	43
4.1.2 Control Points.....	43
4.1.3 Elements.....	44
4.1.4 Gauss Points .....	44
4.2 Μόρφωση Μητρώου Στιβαρότητας .....	45
4.2.1 Μητρώο Στιβαρότητας 1D.....	45

4.2.2	Μητρώο Στιβαρότητας 2D .....	48
5	Τοπολογική Βελτιστοποίηση –Topology Optimization .....	53
5.1	Εισαγωγή .....	53
5.2	Βέλτιστος Σχεδιασμός Κατασκευών .....	53
5.2.1	Κατηγορίες Αλγορίθμων Βελτιστοποίησης .....	53
5.2.2	Κατηγορίες Βελτιστοποίησης Κατασκευών.....	54
5.3	Μαθηματικό Μοντέλο Βελτιστοποίησης .....	56
5.3.1	Αντικειμενική Συνάρτηση.....	57
5.3.2	Μεταβλητές Σχεδιασμού.....	57
5.3.3	Συναρτήσεις Περιορισμού.....	57
5.4	Επίλυση μαθηματικού μοντέλου με τη μέθοδο Optimality Criteria (OC) .....	58
5.5	Εφαρμογή βελτιστοποίησης τοπολογίας σε έργα Πολιτικού Μηχανικού .....	58
6	Επίλυση προβλημάτων με τη Μέθοδο SIMP σε κώδικα MATLAB .....	61
6.1	Εισαγωγή .....	61
6.2	Μέθοδος SIMP (Solid Isotropic Material with Penalization).....	61
6.3	Προβλήματα της μεθόδου .....	62
6.4	Κώδικες Βελτιστοποίησης γραμμένοι σε MATLAB .....	62
6.4.1	Κώδικας SIMP Μοντελοποίησης στοιχείων σε δύο διαστάσεις .....	62
6.5	Βελτιστοποίηση Τοπολογίας με Ισογεωμετρική Ανάλυση σε κώδικα MATLAB – IGA for Topology Optimization .....	65
6.5.1	Εισαγωγή.....	65
6.5.2	Αλγόριθμος Ισογεωμετρικής ανάλυσης (IGA) .....	66
6.6	Απλές εφαρμογές Τοπολογικής Βελτιστοποίησης .....	68
7	Συμπεράσματα.....	72
8	Βιβλιογραφία .....	74
	Παράρτημα Α. Αποτελέσματα Αναλύσεων .....	76
	Παράρτημα Β. Κώδικες αλγορίθμων για επίλυση προβλημάτων βελτιστοποίησης με FEM και IGA .....	90

## **Τοπολογική Βελτιστοποίηση με Ισογεωμετρικά Πεπερασμένα Στοιχεία**

Μουστόγιαννη Δ. Χ. (Επιβλέπων: Παπαδόπουλος Β.)

### **Περίληψη**

Η εργασία με τίτλο «Τοπολογική Βελτιστοποίηση με Ισογεωμετρικά Πεπερασμένα Στοιχεία», εκπονήθηκε στα πλαίσια προπτυχιακού επιπέδου σπουδών με σκοπό την εξέταση ενός εργαλείου βέλτιστου σχεδιασμού προβλημάτων μηχανικής, τη Τοπολογική Βελτιστοποίηση, που συνδυάζει την ανάλυση με πεπερασμένα στοιχεία και τον μαθηματικό προγραμματισμό. Παράλληλα εξετάστηκε η υλοποίησή της με δύο μεθόδους ανάλυσης, την ανάλυση με Πεπερασμένα Στοιχεία και την ανάλυση με Ισογεωμετρικά Πεπερασμένα Στοιχεία.

Πρώτος στόχος της Τοπολογικής Βελτιστοποίησης είναι ο σχεδιασμός μίας κατασκευής με τη βέλτιστη κατανομή υλικού σε δεδομένο χώρο, ενώ συγχρόνως θα πληρούνται τα κριτήρια σχεδιασμού και θα ικανοποιούνται οι περιορισμοί. Η επίλυση των προβλημάτων βελτιστοποίησης γίνεται με τη μέθοδο “SIMP” (Solid Isotropic Microstructure/Material with Penalization for intermediate densities), ώστε η βέλτιστη λύση που θα προκύψει να μην αποτελείται από στοιχεία ενδιάμεσων πυκνοτήτων.

Η μέθοδος των Πεπερασμένων Στοιχείων όπως και η Ισογεωμετρική Ανάλυση είναι αριθμητικοί μέθοδοι μοντελοποίησης φυσικών προβλημάτων και βρίσκουν ευρεία εφαρμογή στην επιστήμη του Πολιτικού Μηχανικού. Οι μέθοδοι αυτοί θα χρησιμοποιηθούν για την προσομοίωση προβλημάτων βελτιστοποίησης τοπολογίας.

Η Ισογεωμετρική Ανάλυση αποτελεί μία καινοτόμο μέθοδο και βασική της ιδέα είναι η χρήση κοινών συναρτήσεων κατά την σχεδίαση και την ανάλυση φορέων, αποφεύγοντας την προσέγγιση της γεωμετρίας λόγω διακριτοποίησης.

Για την διερεύνηση της Τοπολογικής Βελτιστοποίησης με τις δύο παραπάνω μεθόδους ανάλυσης, αναπτύχθηκαν οι αντίστοιχοι αλγόριθμοι για την επίλυση προβλημάτων βελτιστοποίησης τοπολογίας. Οι κώδικες προγραμματίστηκαν σε περιβάλλον “MATLAB”, καθώς διαθέτει ισχυρό μαθηματικό υπόβαθρο, διαδραστικό περιβάλλον αλλά και ευκολία στη χρήση.

Οι δύο μέθοδοι μελετήθηκαν χωριστά και τα αποτελέσματά τους αξιολογήθηκαν ως προς την ακρίβεια λύσης.

NATIONAL TECHNICAL UNIVERSITY OF ATHENS  
FACULTY OF CIVIL ENGINEERING  
INSTITUTE OF STATIC ANALYSIS & ANTISEISMIC RESEARCH

DIPLOMA THESIS

## **Topology Optimization with Isogeometric Finite Elements**

Moustogianni D. C. (supervised by Papadopoulos V.)

### **Abstract**

The diploma thesis entitled “Topology Optimization” with Isogeometric Finite Elements”, is carried out for the completion of the undergraduate studies and its objective is to investigate a design optimization tool used in engineering applications, called “Topology Optimization”, which combines Finite Elements Analysis with Mathematical Programming. Both cases of Topology Optimization with Finite Element and with Isogeometric Elements are examined.

The primary goal of Topology Optimization is the design of a structure with optimized material distribution in a reference domain, while design variables and constraints are satisfied. The solution of optimization problems is performed using “SIMP” (Solid Isotropic Microstructure/Material with Penalization for intermediate densities), so that the resulting solution is not comprised of elements with intermediate densities.

The Finite Element Method and the Isogeometric Analysis are numerical methods for the simulation of structures and have a wide range of civil engineering applications. These methods will be used for the simulation of structures within the framework of topology optimization

Isogeometric analysis is an innovative methodology and its main idea is to use the same functions for both the design and analysis of a structure, thus eliminating the geometrical approximation introduced by meshing.

For the investigation of Topology Optimization utilizing the two methods of analysis, the corresponding algorithms for topology optimization applications were developed. The algorithms were developed with “MATLAB” since it provides a strong basis mathematics, an interactive environment and easiness of use.

Both of analysis methods were studied separately and their results were assessed for the accuracy of the solution.

## Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω πραγματικά τους ανθρώπους που συνέβαλαν καθ' όλη τη διάρκεια της φοιτητικής μου πορείας.

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή τον κ. Βησσαρίωνα Παπαδόπουλο που μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο θέμα και με έκανε να αναθεωρήσω σχετικά με την επιστήμη του Πολιτικού Μηχανικού. Η συνεργασία μας ήταν εξαιρετική σε όλη της τη διάρκεια και ήταν πάντα πρόθυμος να ασχοληθεί με οποιοδήποτε προβληματισμό μου.

Θα ήθελα επίσης να εκφράσω ένα μεγάλο ευχαριστώ στον υποψήφιο Διδάκτορα κ. Δημήτρη Τσαπέτη, για την πολύτιμη βοήθειά του αλλά και τον χρόνο που αφιέρωσε τόσο για να κατανοήσω τα θεωρητικά στοιχεία όσο και να τα υλοποιήσω στη συνέχεια.

Τέλος, το μεγαλύτερο ευχαριστώ θέλω να το εκφράσω στην οικογένεια μου που με στηρίζει σε κάθε μου απόφαση και που δείχνει εμπιστοσύνη σε ό,τι επιλέγω να κάνω. Χάρη στη βοήθεια και τη συνεχή στήριξη των δικών μου ανθρώπων, μπορώ να θέτω στόχους και να προσπαθώ να τους φέρνω εις πέρας.





# 1 Εισαγωγή

## 1.1 Αντικείμενο Διπλωματικής Εργασίας

Κατά τον σχεδιασμό ενός έργου μηχανικού ο κύριος στόχος είναι η εύρεση της όσο το δυνατόν καλύτερης λύσης. Αναζητείται δηλαδή, ο βέλτιστος σχεδιασμός του έργου με τον οποίο αφού εξασφαλιστεί ότι ικανοποιούνται όλες οι λειτουργικές απαιτήσεις και οι περιορισμοί, θα ελαχιστοποιούνται παράλληλα κάποια κριτήρια.

Η βασική ιδέα της παρούσας διπλωματικής εργασίας λοιπόν, είναι η διερεύνηση της εφαρμογής της βελτιστοποίησης στα έργα του Πολιτικού Μηχανικού, η οποία ονομάζεται «Βελτιστοποίηση Κατασκευών», και συγκεκριμένα εξετάστηκε η περίπτωση της «Τοπολογικής Βελτιστοποίησης». Στόχος της Τοπολογικής Βελτιστοποίησης είναι η βέλτιστη κατανομή του υλικού σε συγκεκριμένο πεδίο σχεδιασμού, υπό δεδομένες συνθήκες φόρτισης και στήριξης, με την προϋπόθεση ότι ικανοποιούνται οι απαιτήσεις σχεδιασμού και ελαχιστοποιείται παράλληλα κάποιο επιθυμητό κριτήριο. Για την κατανόηση της εφαρμογής της Τοπολογικής Βελτιστοποίησης, αναλύεται το θεωρητικό της υπόβαθρο, η μέθοδος επίλυσης προβλημάτων βελτιστοποίησης SIMP και η τεχνική βελτιστοποίησης Optimality Criteria (OC), που επιλέχθηκαν για τη συγκεκριμένη εργασία.

Δεύτερος στόχος της διπλωματικής εργασίας, είναι διερεύνηση δύο μεθόδων για την ανάλυση των φορέων. Αρχικά επιλέχθηκε η Μέθοδος Πεπερασμένων Στοιχείων (Finite Element Analysis, FEA). Πρόκειται για μία αριθμητική μέθοδο η οποία μπορεί να επιλύει σύνθετα προβλήματα του μηχανικού και συγχρόνως είναι εφικτό να προγραμματιστεί σε ηλεκτρονικό υπολογιστή. Η επίλυση των διαφορικών εξισώσεων γίνεται με τη χρήση μερικών παραγώγων. Αν και η μέθοδος των Πεπερασμένων Στοιχείων παρέχει τη δυνατότητα διαχείρισης και επίλυσης σύνθετων προβλημάτων, μειονεκτεί στο ότι δεν συνδέεται με σχεδιαστικά προγράμματα CAD (Computer-Aided Design).

Αντίθετα η ανάλυση με Ισογεωμετρικά Πεπερασμένα Στοιχεία, που είναι μία πρόσφατα αναπτυσσόμενη μεθοδολογία, συνδέει την κλασική ανάλυση με Πεπερασμένα Στοιχεία με το επιστημονικό πεδίο των σχεδιαστικών προγραμμάτων CAD. Η μέθοδος της Ισογεωμετρικής Ανάλυσης (Isogeometric Analysis, IGA) χρησιμοποιεί τις ίδιες συναρτήσεις και στο σχεδιασμό και στη ανάλυση δίνοντας το πλεονέκτημα της διατήρησης της συνέχειας της λύσης και της γεωμετρίας, εξασφαλίζοντας μεγαλύτερη ακρίβεια.

Για την σύγκριση των δύο μεθόδων ανάλυσης κατά την επίλυση προβλημάτων Τοπολογικής Βελτιστοποίησης υλοποιήθηκαν οι κατάλληλοι αλγόριθμοι. Για την εξαγωγή και επαλήθευση συμπερασμάτων των δύο μεθόδων, υλοποιήθηκαν τυπικές εφαρμογές 2D προβλημάτων.

## 1.2 Δομή

Για καλύτερη κατανόηση του περιεχομένου της εργασίας χωρίστηκε και αναπτύχθηκε σε επιμέρους ενότητες. Αποτελείται συνολικά από οχτώ ενότητες, των οποίων το περιεχόμενο αναφέρεται περιληπτικά παρακάτω.

Στη **2<sup>η</sup> Ενότητα** αναφέρεται το απαιτούμενο μαθηματικό υπόβαθρο που απαιτείται για την κατανόηση της μεθόδου Πεπερασμένων Στοιχείων. Αρχικά γίνεται η ιστορική αναδρομή της και στη συνέχεια η μαθηματική της διατύπωση για να είναι εφικτός έπειτα ο προγραμματισμός της. Δίνεται έμφαση στον τρόπο υπολογισμού των ισοπαραμετρικών στοιχείων, καθώς αυτά θα χρησιμοποιηθούν στις παρακάτω αναλύσεις.

Στην **3<sup>η</sup> Ενότητα**, παρουσιάζεται η Ισογεωμετρική Ανάλυση. Αφού γίνει η ιστορική της αναδρομή, αναφέρονται όλες οι έννοιες που αφορούν την Ισογεωμετρική Ανάλυση, όπως το Knot Value Vector και τα Control Points, δίνοντας έμφαση στις συναρτήσεις που χρησιμοποιεί. Συγκεκριμένα, αναφέρονται οι B-SPLines καμπύλες και συναρτήσεις, οι ιδιότητές τους καθώς και μία ειδική περίπτωση των B-SPLines συναρτήσεων, οι NURBS (Non Uniform Rational B-SPLines). Για την εκπόνηση της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκαν οι συναρτήσεις NURBS.

Στην **4<sup>η</sup> Ενότητα** περιγράφεται ο τρόπος μόρφωσης του τοπικού και του καθολικού μητρώου στιβαρότητας. Επιπλέον, αναφέρεται ο τρόπος προσδιορισμού των συντοκικών και των φορτίων, καθώς και ο υπολογισμός των τάσεων και παραμορφώσεων του φορέα.

Στην **5<sup>η</sup> Ενότητα** παρουσιάζεται η έννοια της Βελτιστοποίησης Κατασκευών και η μαθηματική διατύπωση της Τοπολογικής Βελτιστοποίησης.

Στην **6<sup>η</sup> Ενότητα** ακολουθεί η ανάλυση της μεθόδου SIMP που επιλέχθηκε για την επίλυση των προβλημάτων. Επιπλέον, παρουσιάζονται και εξηγούνται οι κώδικες της μεθόδου SIMP σε περιβάλλον MATLAB. Οι κώδικες της Βελτιστοποίησης Τοπολογίας αφορούν αναλύσεις με διδιάστατα ισοπαραμετρικά πεπερασμένα στοιχεία και αναλύσεις με διδιάστατα ισογεωμετρικά πεπερασμένα στοιχεία.

Στην **7<sup>η</sup> Ενότητα** παρουσιάζονται διάφορα παραδείγματα Τοπολογικής Βελτιστοποίησης με τις δύο μεθόδους ανάλυσης. Τέλος καταγράφονται τα συμπεράσματα που προέκυψαν από τη σύγκριση των αποτελεσμάτων της ανάλυσης με Πεπερασμένα Στοιχεία και της Ισογεωμετρικής.

Στην **8<sup>η</sup> Ενότητα** αναφέρονται όλες οι βιβλιογραφικές πηγές που χρησιμοποιήθηκαν προκειμένου να υλοποιηθεί η διπλωματική εργασία.

## 2 Μέθοδος Πεπερασμένων Στοιχείων – Finite Elements Analysis (FEA)

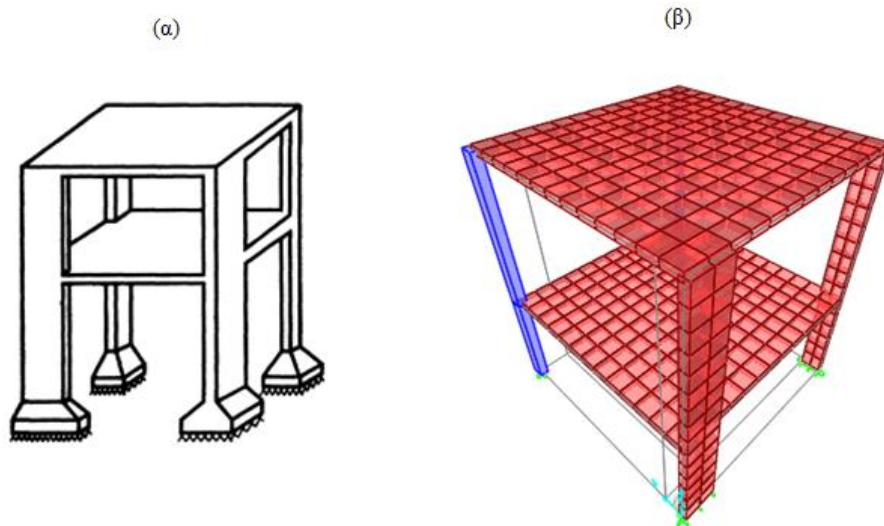
### 2.1 Εισαγωγή

Για την αριθμητική ανάλυση των σύνθετων προβλημάτων του μηχανικού, με τη χρήση σύγχρονων υπολογιστικών μέσων, χρησιμοποιούνται διάφορες αριθμητικές μέθοδοι. Αν και η διατύπωση των περισσότερων προβλημάτων είναι πλέον δυνατή, η εύρεση της ακριβούς λύσεως, απαιτεί ιδιαίτερο υπολογιστικό χρόνο. Γι' αυτό λοιπόν, αναπτύχθηκαν οι προσεγγιστικές μέθοδοι. Η μέθοδος των Πεπερασμένων Στοιχείων, πρόκειται για μία αριθμητική μέθοδο επίλυσης προβλημάτων αυθαίρετης γεωμετρίας, οριακών συνθηκών και φορτίων, βασισμένη στις μεθόδους Rayleigh-Ritz [1] και σταθμικών υπολοίπων.

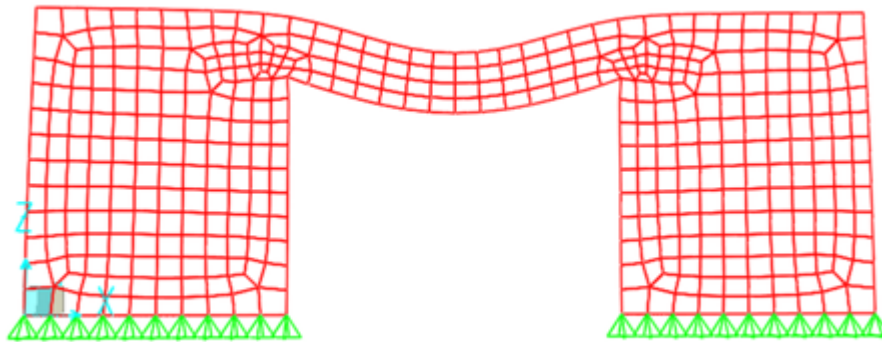
Ο στόχος της μεθόδου Πεπερασμένων Στοιχείων είναι η προσομοίωση της πραγματικής κατασκευής με μικρότερα και απλούστερα στοιχεία τα οποία συνδέονται σε έναν πεπερασμένο αριθμό κόμβων. Τα στοιχεία αυτά μπορεί να είναι διδιάστατα ή τριδιάστατα στοιχεία. Για να γίνει χρήση μητρικών μεθόδων απαιτείται να προσομοιωθεί η κατασκευή με ένα πεπερασμένο αριθμό συνεχών μεταβλητών. Οι μεταβλητές αυτές είναι οι μετατοπίσεις κόμβων και οι παράγωγοί τους.

Καθώς οι μετατοπίσεις στο εσωτερικό των στοιχείων εξαρτώνται πλήρως από τις μετατοπίσεις των κόμβων, οι πλέον άγνωστοι του προβλήματος είναι οι επικόμβιες μετατοπίσεις, και έτσι το πρόβλημα από συνεχές μετατρέπεται σε διακριτό. Αφού λοιπόν ληφθούν υπόψη οι οριακές συνθήκες και η φόρτιση της κατασκευής, καταλήγουμε στις εξισώσεις ισορροπίας, που θα επιλυθούν με τη χρήση αριθμητικών μεθόδων. Επομένως, γνωρίζοντας τις κομβικές μετατοπίσεις, είναι δυνατός ο υπολογισμός των τάσεων ( $\sigma$ ) και των παραμορφώσεων ( $\epsilon$ ) σε κάθε σημείο της κατασκευής, όπως και των αντιδράσεων ( $R$ ) στις στηρίξεις.

Η μέθοδος Πεπερασμένων Στοιχείων, αν και αναπτύχθηκε για την επίλυση προβλημάτων μηχανικής στερεών σωμάτων, εφαρμόζεται σε κάθε πρόβλημα μηχανικής που περιγράφεται από διαφορικές εξισώσεις με μερικές παραγώγους.



(γ)



Σχήμα 2.1 Διακριτοποίηση κατασκευών στο λογισμικό SAP (α) μονώροφο κτίριο - (β) προσομοίωση μονώροφου κτιρίου - (γ)προσομοίωση τοιχωματικού πλαισίου με στοιχεία επίπεδης έντασης

## 2.2 Ιστορική Αναδρομή

Οι αρχές της μεθόδου Πεπερασμένων Στοιχείων, αναπτύχθηκαν το 1909 από τον Γερμανό μαθηματικό Ritz ενώ το 1915 μελετήθηκαν σε βάθος από τον Ρώσο μαθηματικό Galerkin. Ωστόσο, λόγω της απουσίας ηλεκτρονικού υπολογιστή, η μέθοδος δεν αναπτύχθηκε περαιτέρω αλλά ούτε και διαδόθηκε. Η ιδέα ανάπτυξης της μεθόδου γεννήθηκε στην αεροναυπηγική, κατά τη διάρκεια του Β' Παγκόσμιου Πολέμου, για την αντιμετώπιση προβλημάτων στην κατασκευή αεροσκαφών.

Τα πρώτα ουσιαστικά βήματα έγιναν το 1941 όταν ο Hrenikoff [2] επινόησε μία μέθοδο για την επίλυση προβλημάτων ελαστικότητας δύο διαστάσεων, προσεγγίζοντας το πεδίο με ισοδύναμες ράβδους και δοκούς. Ακολούθησε το 1943, η θεωρία του Γερμανού μαθηματικού Courant [3] αλλά δεν εφαρμόστηκε καθώς δεν υπήρχε ακόμα ο ηλεκτρονικός υπολογιστής. Ακολούθησαν αρκετές δημοσιεύσεις, την περίοδο 1954-1960, όπως αυτές του I. Αργύρη και των συνεργατών του, καθώς και των Αμερικανών Turner, Clough, Martin και Top [4].

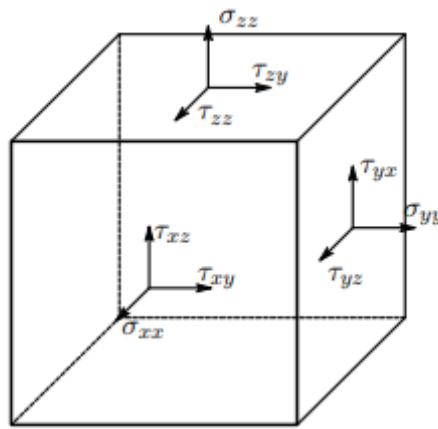
Η πρώτη επίσημη παρουσίαση της μεθόδου έγινε το 1960, από τους I. Αργύρη και Kelsey [5]. Η ονομασία «Μέθοδος Πεπερασμένων Στοιχείων» (Finite Element Method – F.E.M) χρησιμοποιήθηκε πρώτη φορά το 1960, από τον καθηγητή του πανεπιστημίου “University of California, Berkeley”, και από τότε καθιερώθηκε. Το πρώτο βιβλίο με θέμα τη μέθοδο Πεπερασμένων Στοιχείων γράφτηκε το 1967 από τους Zienkiewicz και Cheung [6]. Τα πρώτα πεπερασμένα που χρησιμοποιήθηκαν, ήταν τα στοιχεία επίπεδης έντασης ή επίπεδης παραμόρφωσης, ενώ στην πορεία χρησιμοποιήθηκαν τριδιάστατα στοιχεία, στοιχεία κελυφών και γενικά στοιχεία για διάφορες μορφές κατασκευών. Τέλος, ενώ αρχικά χρησιμοποιήθηκαν σε προβλήματα στην γραμμική ελαστική περιοχή, πλέον είναι εφαρμόσιμα σε διαφόρων ειδών προβλήματα, όπως δυναμικής συμπεριφοράς, λυγισμού ή και σε προβλήματα με μη-γραμμική απόκριση και συμπεριφορά υλικού.

## 2.3 Μαθηματικό Υπόβαθρο

Η μαθηματική προσέγγιση της Μεθόδου Πεπερασμένων Στοιχείων για την παρούσα διπλωματική εργασία, βασίστηκε στο βιβλίο «Ανάλυση Φορέων με τη Μέθοδο των Πεπερασμένων Στοιχείων» [7].

### 2.3.1 Τάσεις και παραμορφώσεις

Για την περιγραφή της εντατικής κατάστασης κάθε σημείου ενός φορέα, αρκεί να είναι γνωστές οι έξι συνιστώσες της τάσης  $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}$  και  $\tau_{zx}$ , καθώς ισχύουν οι σχέσεις  $\tau_{xy}=\tau_{yx}, \tau_{xz}=\tau_{zx}$  και  $\tau_{yz}=\tau_{zy}$ . Αντίστοιχα, για την περιγραφή της παραμορφωσιακής κατάστασης κάθε σημείου ενός φορέα, προσδιορίζονται οι αντίστοιχες έξι συνιστώσες της παραμόρφωσης  $\epsilon_x, \epsilon_y, \epsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}$ .



Σχήμα 2. 2 Τριαξονική Εντατική Κατάσταση Πεπερασμένου Στοιχείου [7]

### 2.3.2 Αρχή Δυνατών Έργων

Για τον προσδιορισμό της εξίσωσης ισορροπίας του φορέα που φορτίζεται με τα εξωτερικά φορτία  $\{f_s\}, \{f^V\}, \{R^C\}$ , διατυπώνεται η αρχή των δυνατών έργων, σύμφωνα με την οποία «Όταν ένας φορέας φορτίζεται με εξωτερικά φορτία και ισορροπεί, τότε για οποιαδήποτε “μικρή” δυνατή παραμόρφωση του φορέα, συμβιβαστή με τις συνθήκες στηρίζεώς του, το δυνατό έργο των εσωτερικών δυνάμεων ισούται με το δυνατό έργο των εξωτερικών δυνάμεων».

Το συνολικό δυνατό έργο των δυνάμεων, ορθών και διατμητικών, που δρουν σε ένα στοιχειώδες παραλληλεπίπεδο ισούται με

$$dW_{\text{int}} = (\bar{\epsilon}_x \sigma_x + \bar{\epsilon}_y \sigma_y + \bar{\epsilon}_z \sigma_z + \bar{\gamma}_{xy} \tau_{xy} + \bar{\gamma}_{yz} \tau_{yz} + \bar{\gamma}_{zx} \tau_{zx}) dV, \quad (2.1)$$

και σε όλο τον φορέα θα ισούται με

$$W_{\text{int}} = \int_V \{\bar{\epsilon}\}^T \{\sigma\} dV, \quad (2.2)$$

ενώ το δυνατό έργο των εξωτερικών δυνάμεων θα ισούται με το άθροισμα του έργου των επιφανειακών δυνάμεων, του έργου των μαζικών δυνάμεων, και του έργου των επικόμβιων δυνάμεων. Έτσι,

$$W_{ext} = \int_V \{\bar{U}\}^T \{f^V\} dV + \int_S \{\bar{U}_S\}^T \{f_S\} dS + \{\bar{D}\}^T \{R_C\}, \quad (2.3)$$

όπου  $\{\bar{U}\}$ , είναι το διάνυσμα των δυνατών μετατοπίσεων ενός τυχαίου σημείου του φορέα με συντεταγμένες  $(X,Y,Z)$ ,  $\{U^S\}$  είναι το διάνυσμα των δυνατών μετατοπίσεων ενός τυχαίου σημείου της φορτιζόμενης επιφάνειας  $S$ , και  $\{\bar{D}\}, \{R_C\}$  είναι τα διανύσματα των δυνατών επικόμβιων μετατοπίσεων και επικόμβιων δράσεων του φορέα, αντίστοιχα.

Επομένως, από την αρχή των δυνατών έργων, κατά την οποία  $W_{int} = W_{ext}$ , προσδιορίζονται οι δυνατές ανηγμένες παραμορφώσεις  $\{\bar{\epsilon}\}$ .

### 2.3.3 Σχέσεις ανηγμένων παραμορφώσεων – μετατοπίσεων

Για τον προσδιορισμό της σχέσης μεταξύ των ανηγμένων παραμορφώσεων και των μετατοπίσεων, γίνεται η παραδοχή των μικρών παραμορφώσεων. Έτσι, για ένα στοιχειώδες ορθογώνιο ενός επίπεδου σώματος, με μετατοπίσεις  $U,V$  εκφρασμένες συναρτήσει των συντεταγμένων  $X,Y$ , υπολογίζονται οι ανηγμένες παραμορφώσεις από τις σχέσεις

$$\epsilon_X = \frac{\left[ dX + \left( U + \frac{\partial U}{\partial X} dX \right) - U \right] - dX}{dX} = \frac{\partial U}{\partial X}, \quad (2.4\alpha)$$

$$\epsilon_Y = \frac{\left[ dY + \left( V + \frac{\partial V}{\partial Y} dY \right) - V \right] - dY}{dY} = \frac{\partial V}{\partial Y}, \quad (2.4\beta)$$

$$\epsilon_Z = \frac{\left[ dZ + \left( W + \frac{\partial W}{\partial Z} dZ \right) - W \right] - dZ}{dZ} = \frac{\partial W}{\partial Z}, \quad (2.4\gamma)$$

Η διατμητική ανηγμένη παραμόρφωση του στοιχειώδους παραλληλογράμμου  $dX-dY$ , υπολογίζεται επίσης με την παραδοχή των μικρών παραμορφώσεων και δίνεται από τη σχέση

$$\gamma_{XY} = \frac{\left[ \left( U + \frac{\partial U}{\partial Y} dY \right) - U \right]}{dY} + \frac{\left[ \left( V + \frac{\partial V}{\partial X} dX \right) - V \right]}{dX} = \frac{\partial U}{\partial Y} + \frac{\partial V}{\partial X}, \quad (2.5)$$

Κατ' αντιστοιχία, υπολογίζονται το σύνολο των σχέσεων ανηγμένων παραμορφώσεων-μετατοπίσεων στις τρεις διαστάσεις και δίνονται από τις σχέσεις

$$\begin{aligned}
 \varepsilon_x &= \frac{\partial U}{\partial X} & \gamma_{xy} &= \frac{\partial U}{\partial Y} + \frac{\partial V}{\partial X} \\
 \varepsilon_y &= \frac{\partial V}{\partial Y} & \gamma_{yz} &= \frac{\partial V}{\partial Z} + \frac{\partial W}{\partial Y} \\
 \varepsilon_z &= \frac{\partial W}{\partial Z} & \gamma_{zx} &= \frac{\partial W}{\partial X} + \frac{\partial U}{\partial Z}
 \end{aligned}
 \tag{2.6}$$

### 2.3.4 Σχέσεις τάσεων – ανηγμένων παραμορφώσεων

Ο υπολογισμός των τάσεων γίνεται σύμφωνα με τον καταστατικό νόμο του *Hooke*, κατά τον οποίο οι τάσεις είναι γραμμικές συναρτήσεις των ανηγμένων παραμορφώσεων. Επομένως, οι μητρικές σχέσεις μεταξύ τάσεων – ανηγμένων παραμορφώσεων δίνονται από τις σχέσεις

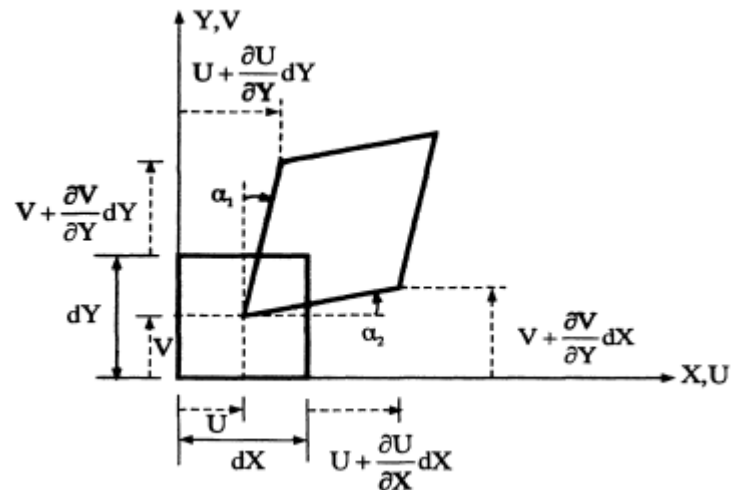
$$\{\varepsilon\} = [C]\{\sigma\}, \tag{2.7a}$$

$$\{\sigma\} = [E]\{\varepsilon\}, \tag{2.7β}$$

όπου  $[C]$  είναι το μητρώο που εκφράζει την παραμόρφωση του υλικού στις ασκούμενες τάσεις και  $[E]$  είναι το μητρώο στιβαρότητας του υλικού στις επιβαλλόμενες παραμορφώσεις. Για τα ισότροπα υλικά οι παραπάνω ιδιότητες των υλικών, εκφράζονται συναρτήσει του μέτρου ελαστικότητας  $E$  και του συντελεστή *Poisson*  $\nu$ , και δίνονται από τις σχέσεις

$$[C] = \frac{1}{E} \cdot \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix}, \tag{2.8}$$

$$[E] = \frac{E}{(1+\nu) \cdot (1-2\nu)} \cdot \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}, \tag{2.9}$$



Σχήμα 2.3 Παραμόρφωση στοιχειώδους ορθογωνίου  $dX-dY$  [7].

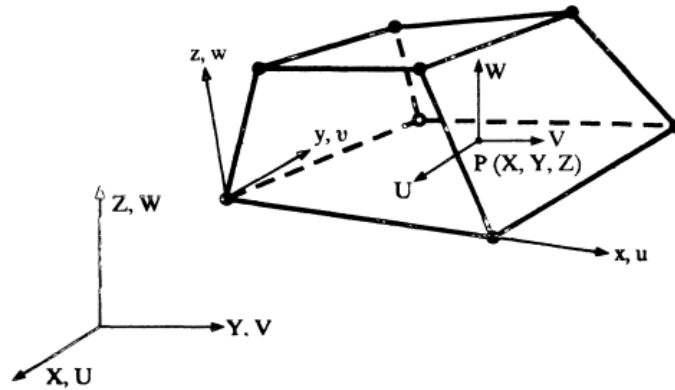
### 2.3.5 Συναρτήσεις σχήματος πεπερασμένου στοιχείου και πολώνυμα παρεμβολής

#### Συναρτήσεις σχήματος

Η βασική ιδέα της μεθόδου των Πεπερασμένων Στοιχείων είναι η διακριτοποίηση του φορέα σε επιμέρους στοιχεία. Η συμπεριφορά των στοιχείων καθορίζεται από συναρτήσεις οι οποίες ορίζουν τον τρόπο μεταβολής των μετατοπίσεων στο εσωτερικό τους. Για κάθε στοιχείο δηλαδή, ορίζεται ένα μητρώο συναρτήσεων σχήματος  $[N(X,Y,Z)]$  το οποίο εκφράζει τον τρόπο με τον οποίο υπολογίζεται το πεδίο μετατοπίσεων συναρτήσει των επικόμβιων μετατοπίσεών του  $\{d\} = [U_1 \ V_1 \ W_1 \ U_2 \ V_2 \ W_2 \ \dots]^T$ , όπου  $U, V, W$  οι συνιστώσες της μετατόπισης στο καθολικό σύστημα του φορέα κάθε σημείου  $(X, Y, Z)$ . Έτσι, για ένα στοιχείο  $m$  με  $n$  κόμβους, η μετατόπισή του δίνεται από τη σχέση

$$\begin{bmatrix} U(X,Y,Z) \\ V(X,Y,Z) \\ W(X,Y,Z) \end{bmatrix} = [N(X,Y,Z)] \cdot \begin{bmatrix} U_1 \\ V_1 \\ W_1 \\ \vdots \\ U_n \\ V_n \\ W_n \end{bmatrix}, \quad (2.10)$$





Σχήμα 2.4 Εξαεδρικό στοιχείο οκτώ κόμβων με το τοπικό και το καθολικό σύστημα αξόνων [7]

### Πολυώνυμα Παρεμβολής

Για τον υπολογισμό των συναρτήσεων σχήματος, χρησιμοποιούνται είτε οι συναρτήσεις παρεμβολής *Lagrange* είτε οι συναρτήσεις παρεμβολής *Hermite*. Προτιμάται, η χρήση των πολυωνύμων παρεμβολής *Lagrange*, καθώς στην περίπτωση αυτή αποφεύγεται η αντιστροφή του μητρώου [A] που συνδέει τις επικόμβιες μετατοπίσεις με τις γενικευμένες συντεταγμένες.

Συγκεκριμένα, σύμφωνα με τις συναρτήσεις παρεμβολής *Lagrange*, μια συνάρτηση  $\phi = \phi(x)$ ,  $(n-1)$  βαθμού και  $n$  τα σημεία ορισμού της, προσεγγίζεται ως εξής

$$\phi = \sum_{i=1}^n L_i \phi_i, \quad (2.11)$$

για  $\phi_i = \phi(x_i)$ , με  $x_i$  τις συντεταγμένες των σημείων της συνάρτησης και  $L_i$  τους συντελεστές *Lagrange*.

$$L_n = \frac{(x_1 - x)(x_2 - x) \dots (x_{n-1} - x)}{(x_1 - x_n)(x_2 - x_n) \dots (x_{n-1} - x_n)}, \quad (2.12)$$

Επομένως, για ένα εξαεδρικό ορθογωνικό στοιχείο οκτώ κόμβων, που είναι η συνήθης περίπτωση, οι συναρτήσεις σχήματος δίνονται από το πολυώνυμο

$$\phi = N_1\phi_1 + N_2\phi_2 + N_3\phi_3 + N_4\phi_4 + N_5\phi_5 + N_6\phi_6 + N_7\phi_7 + N_8\phi_8, \quad (2.13)$$

και προκύπτουν οι σχέσεις

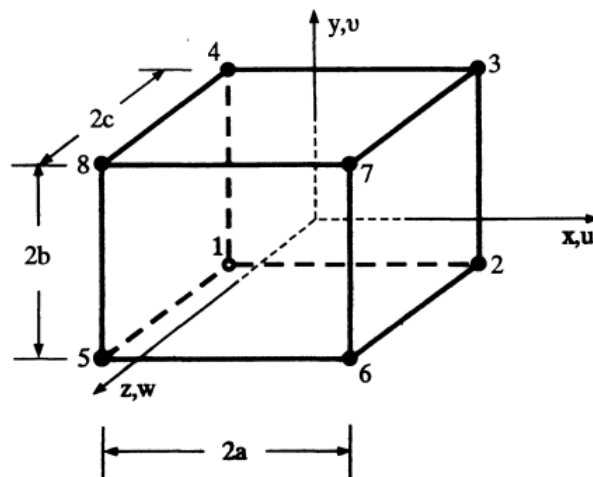
$$\begin{aligned}
 N_1(x, y, z) &= L_\alpha(x)L_\alpha(y)L_\alpha(z) \\
 N_2(x, y, z) &= L_\beta(x)L_\alpha(y)L_\alpha(z) \\
 N_3(x, y, z) &= L_\beta(x)L_\beta(y)L_\alpha(z) \\
 N_4(x, y, z) &= L_\alpha(x)L_\beta(y)L_\alpha(z) \\
 N_5(x, y, z) &= L_\alpha(x)L_\alpha(y)L_\beta(z) \\
 N_6(x, y, z) &= L_\beta(x)L_\alpha(y)L_\beta(z) \\
 N_7(x, y, z) &= L_\beta(x)L_\beta(y)L_\beta(z) \\
 N_8(x, y, z) &= L_\alpha(x)L_\beta(y)L_\beta(z)
 \end{aligned} \tag{2.14}$$

Κατά αυτόν τον τρόπο, υπολογίζεται το πεδίο μετατοπίσεων το οποίο δίνεται από τη σχέση

$$\{u\} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & \dots & N_8 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & \dots & 0 & N_8 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & \dots & 0 & 0 & N_8 \end{bmatrix} \cdot \{d\} \tag{2.15}$$

όπου

$$\{d\} = [u_1 \ v_1 \ w_1 \ u_2 \ v_2 \ w_2 \ u_3 \ v_3 \ w_3 \ u_4 \ v_4 \ w_4 \ u_5 \ v_5 \ w_5 \ u_6 \ v_6 \ w_6 \ u_7 \ v_7 \ w_7 \ u_8 \ v_8 \ w_8]^T, \tag{2.16}$$



Σχήμα 2.5 Εξαεδρικό ορθογωνικό στοιχείο οκτώ κόμβων [7]

## 2.4 Διαστάσεις πεπερασμένων στοιχείων φορέα

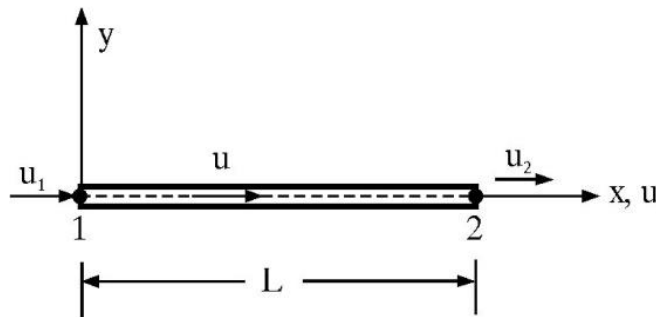
Για την ανάλυση ενός φορέα με τη μέθοδο Πεπερασμένων Στοιχείων, πρέπει να οριστούν αρχικά οι διαστάσεις των επιμέρους στοιχείων προσομοίωσης, του ώστε να σχηματιστούν κατάλληλα τα μητρώα για την εκτέλεση της ανάλυσης. Οι τρεις δυνατές περιπτώσεις είναι οι εξής

### 2.4.1 1-Dimensional Analysis

Στην περίπτωση αυτή, υπάρχει μόνο ένας μετακινησιακός βαθμός ελευθερίας σε κάθε κόμβο, όπως συμβαίνει δηλαδή στην ανάλυση στοιχείων δικτυώματος. Χρησιμοποιείται σε φορείς στους οποίους επιβάλλονται φορτία και μετακινήσεις μόνο κατά μήκος τους και των οποίων το ύψος και το πάχος είναι αρκετά μικρά συγκριτικά με το μήκος τους.

Το μητρώο ελαστικότητας στην περίπτωση μονοδιάστατης ανάλυσης είναι ίσο με το μέτρο ελαστικότητας του στοιχείου

$$[E]_{(1 \times 1)} = E, \quad (2.17)$$



Σχήμα 2.6 Στοιχείο δικτυώματος με δύο κόμβους [7]

### 2.4.2 2-Dimensional Analysis

Για την απλούστευση ενός χωρικού προβλήματος, χωρίς όμως να μειώνεται η ακρίβεια των αποτελεσμάτων, είναι δυνατό να γίνουν κάποιες απλοποιητικές παραδοχές κατά την επίλυση του φορέα. Οι παραδοχές απλοποίησης είναι αποδεκτό να γίνονται όταν το επιτρέπουν τα γεωμετρικά χαρακτηριστικά του φορέα και οι συνθήκες φόρτισής του. Μία τέτοια περίπτωση είναι η θεώρηση συνθηκών επίπεδης έντασης και επίπεδης παραμόρφωσης, όπου κάθε κόμβος θα έχει δύο μετακινησιακούς βαθμούς ελευθερίας.

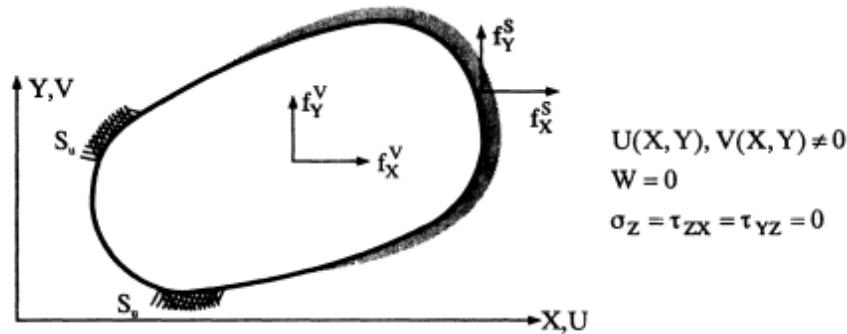
#### Επίπεδη Ένταση

Ένας ολόσωμος φορέας έχει διαστάσεις και στις τρεις διευθύνσεις. Όταν στην διεύθυνση Z η διάστασή του είναι πολύ μικρή συγκριτικά με τις διαστάσεις του στις δύο άλλες διευθύνσεις X, Y, και συγχρόνως δεν ασκούνται φορτία κατά τη διεύθυνση Z στις άνω και κάτω επιφάνειες του φορέα, τις παράλληλες δηλαδή με το επίπεδο XY, τότε γίνεται η θεώρηση της επίπεδης έντασης του φορέα.

Κατά την παραδοχή επίπεδης έντασης του φορέα, μηδενίζονται οι τάσεις  $\sigma_z$ ,  $\tau_{zx}$ ,  $\tau_{yz}$ , ενώ οι συνιστώσες της μετατόπισης U, V, οι ανηγμένες παραμορφώσεις  $\epsilon_x$ ,  $\epsilon_y$ ,  $\gamma_{xy}$ , οι τάσεις  $\sigma_x$ ,  $\sigma_y$ ,  $\tau_{xy}$ , καθώς και τα φορτία, είναι συναρτήσεις μόνο των X και Y.

Από τις σχέσεις ανηγμένων παραμορφώσεων – τάσεων και της τριδιάστατης ελαστικότητας, αντικαθιστώντας τις παραδοχές  $\sigma_z = \tau_{zx} = \tau_{yz} = 0$ , προσδιορίζεται το μητρώο ελαστικότητας  $[E]$  για την περίπτωση επίπεδης έντασης.

$$[E]_{(3 \times 3)} = \frac{E}{(1-\nu)^2} \cdot \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \quad (2.18)$$



Σχήμα 2.7 Ολόσωμος φορέας με συνθήκες επίπεδης έντασης [7]

### Επίπεδη παραμόρφωση

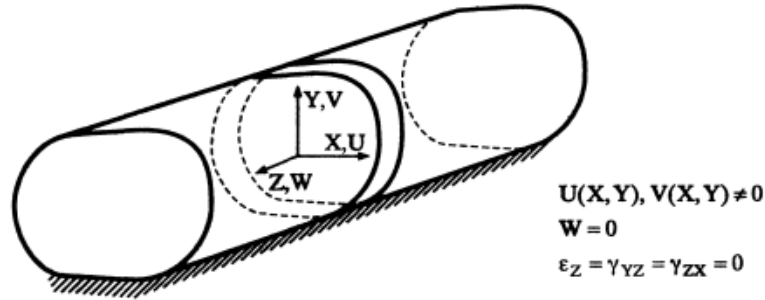
Η θεώρηση συνθηκών επίπεδης παραμόρφωσης, γίνεται όταν η διάσταση κατά μήκος του άξονα Z είναι μεγάλη συγκριτικά με τις άλλες δύο διευθύνσεις και τα φορτία ασκούνται επί του επιπέδου XY.

Κατά την παραδοχή επίπεδης παραμόρφωσης του φορέα, οι ανηγμένες παραμορφώσεις  $\epsilon_z, \gamma_{yz}, \gamma_{zx}$ , είναι μηδενικές, ενώ οι συνιστώσες της μετατόπισης U, V, οι ανηγμένες παραμορφώσεις  $\epsilon_x, \epsilon_y, \gamma_{xy}$ , και οι τάσεις  $\sigma_x, \sigma_y, \tau_{xy}$  είναι συναρτήσεις μόνο των X και Y.

Από τις σχέσεις ανηγμένων παραμορφώσεων – τάσεων και της τριδιάστατης ελαστικότητας, αντικαθιστώντας τις παραδοχές  $\epsilon_z = \gamma_{yz} = \gamma_{zx} = 0$ , προσδιορίζεται το μητρώο ελαστικότητας  $[E]$  για την περίπτωση επίπεδης παραμόρφωσης.

$$[E]_{(3 \times 3)} = \frac{E^*}{(1-\nu^{*2})} \cdot \begin{bmatrix} 1 & \nu^* & 0 \\ \nu^* & 1 & 0 \\ 0 & 0 & \frac{1-\nu^*}{2} \end{bmatrix}, \quad (2.19)$$

όπου  $E^* = \frac{E}{(1-\nu^2)}$  και  $\nu^* = \frac{\nu}{(1-\nu)}$ , (2.20)



Σχήμα 2.8 Ολόσωμος φορέας με συνθήκες επίπεδης παραμόρφωσης [7]

### 2.4.3 3-Dimensional Analysis

Πρόκειται για την προσομοίωση ενός φορέα στις πραγματικές διαστάσεις, όπου κάθε κόμβος έχει τρεις μετακινησιακούς βαθμούς ελευθερίας και δεν γίνεται καμία παραδοχή απλοποίησης του προβλήματος.

Το μητρώο τριδιάστατης ελαστικότητας αναφέρθηκε και παραπάνω και είναι το εξής

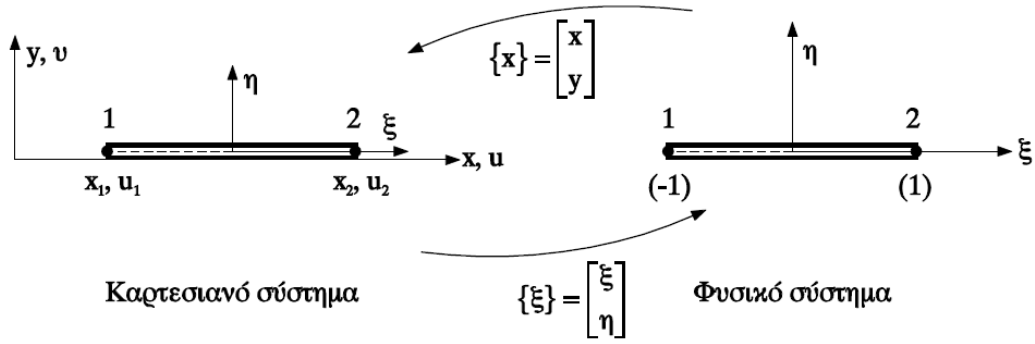
$$[E] = \frac{E}{(1+\nu) \cdot (1-2\nu)} \cdot \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix},$$

## 2.5 Ισοπαραμετρικά Πεπερασμένα Στοιχεία

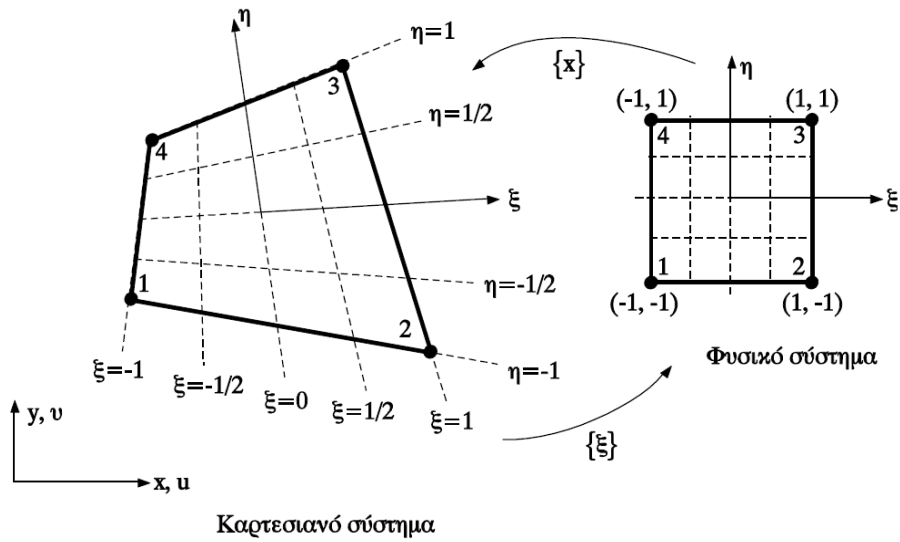
Στόχος της μεθόδου των Πεπερασμένων Στοιχείων είναι η προσομοίωση κάθε πραγματικής κατασκευής διακριτοποιώντας την σε μικρότερα επιμέρους στοιχεία, για την απλούστερη ανάλυσή της. Ωστόσο με τα συνήθη κυβικά στοιχεία (quad elements), δεν είναι δυνατόν να προσομοιωθούν όλα τα είδη σχήματος ή και σχήματα με καμπύλα σύνορα. Επιπλέον, τα αποτελέσματα που προκύπτουν για την προσομοίωση πιο πολύπλοκων φορέων δεν είναι ικανοποιητικής ακρίβειας.

Την λύση έδωσαν οι Taig και Irons, οι οποίοι εμπλούτισαν την μέθοδο των Πεπερασμένων Στοιχείων με τα λεγόμενα ισοπαραμετρικά στοιχεία (isoparametric elements). Η ισοπαραμετρική θεώρηση βασίζεται στη θεώρηση ενός δευτέρου συστήματος συντεταγμένων, το οποίο ορίζεται στο Καρτεσιανό σύστημα (x, y, z) μέσω μίας απεικόνισης και ονομάζεται φυσικό σύστημα (ξ, η, ζ). Για την σύνδεση των συναρτήσεων μεταξύ των δύο συστημάτων, ορίζεται το Ιακωβιανό μητρώο, το οποίο εκφράζει την Ιακωβιανή της απεικόνισης [J].

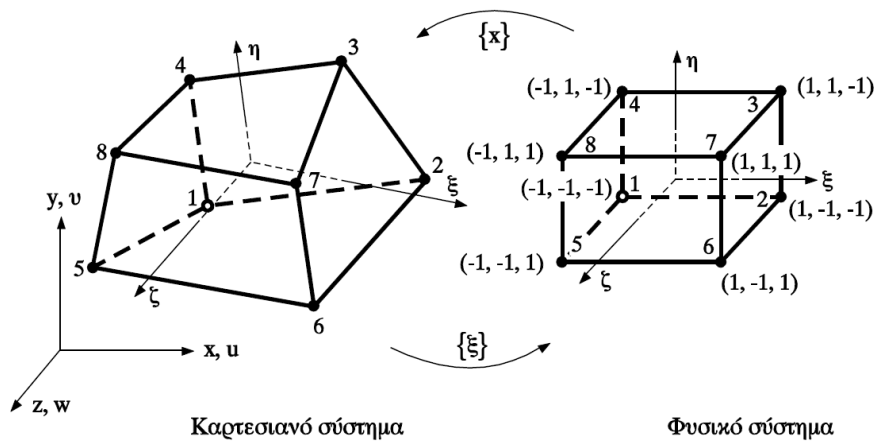
Τοπολογική Βελτιστοποίηση με Ισογεωμετρικά Πεπερασμένα Στοιχεία



Σχήμα 2.9 Απεικόνιση Ραβδωτού ισοπαραμετρικού στοιχείου δύο κόμβων [7]



Σχήμα 2.10 Απεικόνιση τετραπλευρικού ισοπαραμετρικού στοιχείου επίπεδης έντασης και επίπεδης παραμόρφωσης τεσσάρων κόμβων [7]



Σχήμα 2.11 Απεικόνιση Εξαεδρικού ισοπαραμετρικού στοιχείου οκτώ κόμβων τριδιάστατης ελαστικότητας [7]

## 2.6 Διαδικασία Υπολογισμού Μητρώου Στιβαρότητας

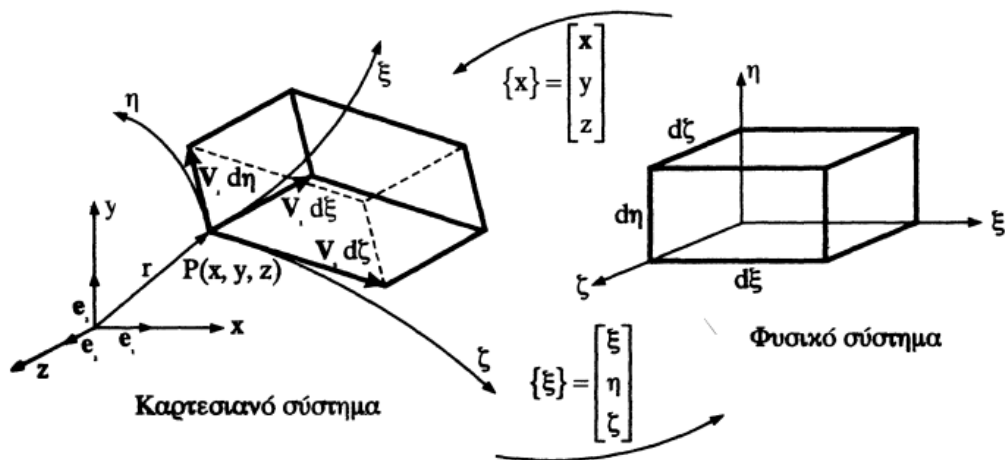
Ακολουθεί ο αναλυτικός υπολογισμός του μητρώου στιβαρότητας για κυβικά ισοπαραμετρικά στοιχεία οκτώ κόμβων καθώς αποτελεί την γενική περίπτωση υπολογισμού, και από αυτό προκύπτουν τα μητρώα κάθε είδους ισοπαραμετρικού στοιχείου.

### Απεικόνιση του Καρτεσιανού συστήματος στο φυσικό σύστημα

Η απεικόνιση του Καρτεσιανού συστήματος στο φυσικό σύστημα πρέπει να είναι αμφιμονοσήμαντη έτσι ώστε κάθε σημείο με Καρτεσιανές συντεταγμένες  $(x, y, z)$  να έχει αντίστοιχες φυσικές  $(\xi, \eta, \zeta)$ , και αντιστρόφως. Η ορθή απεικόνιση των συντεταγμένων  $(x, y, z)$  συναρτήσει των φυσικών συντεταγμένων  $(\xi, \eta, \zeta)$ , και η αντίστροφη απεικόνιση των  $(\xi, \eta, \zeta)$  συναρτήσει των  $(x, y, z)$ , ορίζονται αντιστοίχως ως εξής

$$\begin{aligned} x &= x(\xi, \eta, \zeta) \\ y &= y(\xi, \eta, \zeta), \\ z &= z(\xi, \eta, \zeta) \end{aligned} \tag{2.21\alpha}$$

$$\begin{aligned} \xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z), \\ \zeta &= \zeta(x, y, z) \end{aligned} \tag{2.21\beta}$$



Σχήμα 2.12 Ορθή και αντίστροφη απεικόνιση στο Φυσικό και Καρτεσιανό σύστημα συντεταγμένων [7]

### Συναρτήσεις Σχήματος

Για τον υπολογισμό του μητρώου στιβαρότητας, διατυπώνεται αρχικά η απεικόνιση του φυσικού συστήματος στο Καρτεσιανό σύστημα συντεταγμένων, με όρια απεικόνισης  $-1 \leq \xi \leq 1$  στο φυσικό σύστημα, και  $x_1 \leq x \leq x_2$  στο Καρτεσιανό σύστημα. Η απεικόνιση αυτή ορίζεται ως εξής

$$\begin{aligned} x &= a_1 + a_2\xi + a_3\eta + a_4\zeta + a_5\xi\eta + a_6\eta\zeta + a_7\xi\zeta + a_8\xi\eta\zeta \\ y &= a_9 + a_{10}\xi + a_{11}\eta + a_{12}\zeta + a_{13}\xi\eta + a_{14}\eta\zeta + a_{15}\xi\zeta + a_{16}\xi\eta\zeta, \\ z &= a_{17} + a_{18}\xi + a_{19}\eta + a_{20}\zeta + a_{21}\xi\eta + a_{22}\eta\zeta + a_{23}\xi\zeta + a_{24}\xi\eta\zeta \end{aligned} \quad (2.22)$$

και με τις κατάλληλες αντικαταστάσεις, προκύπτουν οι σχέσεις

$$x = N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 + N_5x_5 + N_6x_6 + N_7x_7 + N_8x_8, \quad (2.23)$$

όπου  $N_i$ , με  $i = 1 \div 8$ , οι συναρτήσεις σχήματος

$$\begin{aligned} N_1 &= \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta) & N_5 &= \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta) \\ N_2 &= \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta) & N_6 &= \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta) \\ N_3 &= \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta) & N_7 &= \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta) \\ N_4 &= \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta) & N_8 &= \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta) \end{aligned} \quad (2.24)$$

### Μητρώο Παραμορφώσεως

Το μητρώο παραμορφώσεως  $[B]$  συνδέει τις ανηγμένες παραμορφώσεις με τις επικόμβιες μετατοπίσεις. Στην περίπτωση των τριδιάστατων ισοπαραμετρικών στοιχείων, το μητρώο παραμορφώσεως  $[B]$  προκύπτει από το γινόμενο δύο επιμέρους μητρώων  $[B_1]$  και  $[B_2]$ .

Για τον προσδιορισμό του μητρώου  $[B_1]$ , υπολογίζεται το Ιακωβιανό μητρώο το οποίο συνδέει τις παραγώγους μίας συνάρτησης  $\phi$  ως προς τα δύο συστήματα συντεταγμένων και εν συνεχεία υπολογίζεται το αντίστροφο του Ιακωβιανού μητρώου  $[J]^{-1}$ , το οποίο έχει στοιχεία  $J_{ij}^*$ . Προκύπτει, λοιπόν το μητρώο  $[B_1]$  ως εξής

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}, \quad (2.25)$$



$$[B_1] = \begin{bmatrix} J_{11}^* & J_{12}^* & J_{13}^* & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & J_{21}^* & J_{22}^* & J_{23}^* & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & J_{31}^* & J_{32}^* & J_{33}^* \\ J_{21}^* & J_{22}^* & J_{23}^* & J_{11}^* & J_{12}^* & J_{13}^* & 0 & 0 & 0 \\ 0 & 0 & 0 & J_{31}^* & J_{32}^* & J_{33}^* & J_{21}^* & J_{22}^* & J_{23}^* \\ J_{31}^* & J_{32}^* & J_{33}^* & 0 & 0 & 0 & J_{11}^* & J_{12}^* & J_{13}^* \end{bmatrix}, \quad (2.26)$$

έτσι ώστε να υπολογιστεί η σχέση

$$\left\{ \varepsilon \right\}_{(6 \times 1)} = [B_1] \left\{ u, \xi \right\}_{(9 \times 1)}, \quad (2.27)$$

Εν συνεχεία, υπολογίζονται οι μερικές παράγωγοι των  $u$ ,  $v$ ,  $w$  ως προς  $\xi$ ,  $\eta$ ,  $\zeta$  και για τον προσδιορισμό της σχέσης που τις συνδέει με τις επικόμβιες μετατοπίσεις, διατυπώνεται το μητρώο  $[B_2]$  ως εξής

$$[B_2] = \begin{bmatrix} N_{1,\xi} & 0 & 0 & N_{2,\xi} & 0 & 0 & \dots & N_{8,\xi} & 0 & 0 \\ N_{1,\eta} & 0 & 0 & N_{2,\eta} & \dots & \dots & \dots & N_{8,\eta} & 0 & 0 \\ N_{1,\zeta} & 0 & 0 & N_{2,\zeta} & \dots & \dots & \dots & N_{8,\zeta} & 0 & 0 \\ 0 & N_{1,\xi} & 0 & 0 & N_{2,\xi} & 0 & \dots & 0 & N_{8,\xi} & 0 \\ 0 & N_{1,\eta} & 0 & 0 & N_{2,\eta} & \dots & \dots & 0 & N_{8,\eta} & 0 \\ 0 & N_{1,\zeta} & 0 & 0 & N_{2,\zeta} & \dots & \dots & 0 & N_{8,\zeta} & 0 \\ \dots & \dots & N_{1,\xi} & 0 & 0 & N_{2,\xi} & \dots & 0 & 0 & N_{8,\xi} \\ \dots & \dots & N_{1,\eta} & \dots & \dots & N_{2,\eta} & \dots & 0 & 0 & N_{8,\eta} \\ \dots & \dots & N_{1,\zeta} & \dots & \dots & N_{2,\zeta} & \dots & 0 & 0 & N_{8,\zeta} \end{bmatrix}, \quad (2.28)$$

$$\left\{ u, \xi \right\}_{(9 \times 1)} = [B_2] \left\{ d \right\}_{(9 \times 24)(24 \times 1)}, \quad (2.29)$$

Επομένως, μπορεί να γίνει ο υπολογισμός των ανηγμένων παραμορφώσεων συναρτήσει των επικόμβιων μετατοπίσεων μέσω της σχέσης

$$\left\{ \varepsilon \right\}_{(6 \times 1)} = [B_1] [B_2] \left\{ d \right\}_{(6 \times 9)(9 \times 24)(24 \times 1)}, \quad (2.30)$$

$$\text{ή} \quad [B] = [B_1] [B_2], \quad (2.31)$$

### Μητρώο στιβαρότητας στοιχείου

Το μητρώο στιβαρότητας ενός εξαεδρικού ισοπαραμετρικού στοιχείου οκτώ κόμβων υπολογίζεται από την εξής σχέση

$$[k]_{(24 \times 24)} = \int_{V_e} [B]_{(24 \times 6)}^T [E]_{(6 \times 6)} [B]_{(6 \times 24)} dV_e, \quad (2.32)$$

ή

$$[k] = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 [B(\xi, \eta, \zeta)]^T [E][B(\xi, \eta, \zeta)] \det[J] d\xi d\eta d\zeta, \quad (2.33\alpha)$$

$$[k] = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(\xi, \eta, \zeta) d\xi d\eta d\zeta, \quad (2.33\beta)$$

### Αριθμητική Ολοκλήρωση Gauss

Για την ολοκλήρωση της διαδικασίας επίλυσης, απαιτείται η αριθμητική ολοκλήρωση του μητρώου στιβαρότητας, καθώς η αναλυτική δεν είναι πάντα δυνατή. Υπάρχουν διάφορες μέθοδοι αριθμητικής ολοκλήρωσης, όπως η ολοκλήρωση *Newton-Cotes* και η ολοκλήρωση *Gauss*. Στην παρούσα διπλωματική εργασία επιλέχθηκε η μέθοδος ολοκλήρωσης *Gauss*.

Πλεονέκτημα της μεθόδου είναι ότι η επιλογή των σημείων υπολογισμού της συνάρτησης  $f$  είναι τυχαία και έτσι είναι δυνατή η βελτίωση της ακρίβειας αλλάζοντας τα σημεία αυτά. Αφού ακολουθήθηκε η κατάλληλη διαδικασία, προσδιορίστηκαν οι θέσεις  $\xi_i$  με τους αντίστοιχους συντελεστές βάρους  $a_i$ , για αριθμό σημείων  $n = 1 \div 6$ . Οι τιμές που υπολογίστηκαν καταγράφονται σε πίνακα.

Για την ολοκλήρωση συνάρτησης τριών μεταβλητών, επαναλαμβάνεται η αριθμητική ολοκλήρωση διαδοχικά σε κάθε διεύθυνση. Ο αριθμός των σημείων σε μία διεύθυνση είναι ανεξάρτητος από τον αριθμό σημείων στις άλλες διευθύνσεις. Η σχέση υπολογισμού του ολοκληρώματος είναι η εξής

$$I = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(\xi, \eta, \zeta) d\xi d\eta d\zeta, \quad (2.34\alpha)$$

και

$$I = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n a_i a_j a_k f(\xi_i, \eta_j, \zeta_k), \quad (2.34\beta)$$

**Πίνακας 2. 1** Συντεταγμένες σημείων και συντελεστές βάρους αριθμητικής ολοκλήρωσης Gauss στο διάστημα (-1,1)

n	$\xi_i, \eta_i, \zeta_i$	$\alpha_i$
1	0.0	2.0
2	$\pm 0.57735$	1.0
3	$\pm 0.77459$	0.55555
	0.0	0.88888
4	$\pm 0.86113$	0.34785
	$\pm 0.33998$	0.65214
5	$\pm 0.90617$	0.23692
	0.53846	0.47862
	0.0	0.56888
6	$\pm 0.93246$	0.17132
	$\pm 0.66120$	0.36076
	$\pm 0.23861$	0.46791



## 3 Ισογεωμετρική Ανάλυση – Isogeometric Analysis (IGA)

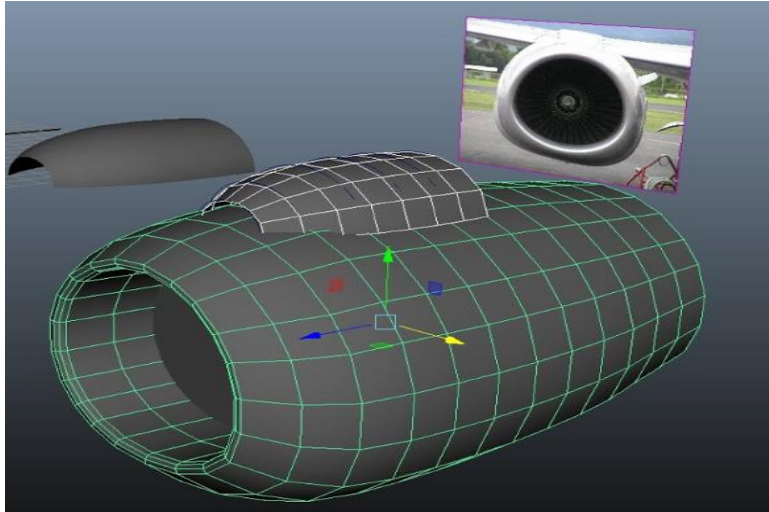
### 3.1 Εισαγωγή

Παρά τη συνεχή εξέλιξή της, η μέθοδος των Πεπερασμένων στοιχείων δεν έχει καταφέρει να δώσει λύση σε όλα τα προβλήματα του μηχανικού αλλά ούτε και την απαραίτητη ακρίβεια στις λύσεις των πιο περίπλοκων. Συγκεκριμένα, ένα από τα βασικά μειονεκτήματα της μεθόδου είναι ότι μετά την πρώτη διακριτοποίηση του φορέα, η αρχική του γεωμετρία παύει να έχει κάποιο ρόλο στη διαδικασία των αναλύσεων. Για μείωση του σφάλματος λοιπόν, χρειάζεται υψηλή ακρίβεια κατά την προσομοίωση του φορέα το οποίο όμως είναι αρκετά χρονοβόρο. Ωστόσο και πάλι δεν εξασφαλίζεται μεγάλη αξιοπιστία στο αποτέλεσμα καθώς λόγω των πολλών επαναλήψεων που απαιτούνται μέχρι την εύρεση της πιο κατάλληλης διακριτοποίησης, είναι αυξημένη η πιθανότητα λάθους από το μηχανικό. Τα παραπάνω προβλήματα, καλείται να λύσει η μέθοδος της Ισογεωμετρικής Ανάλυσης.

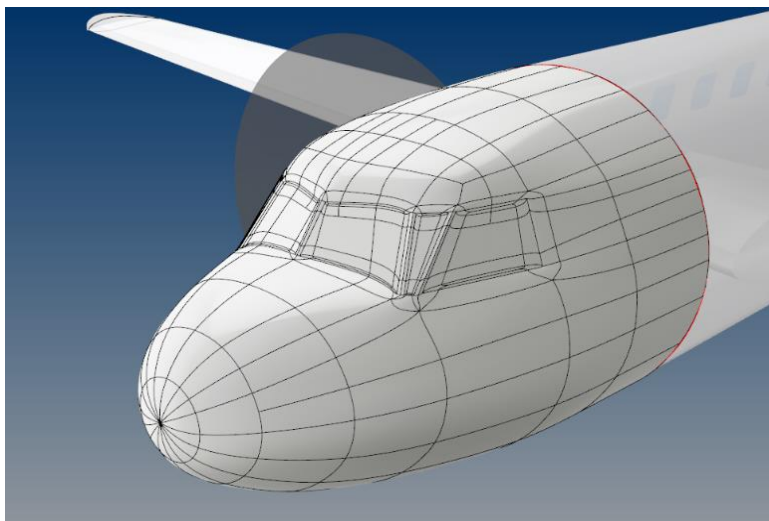
Η λογική της Ισογεωμετρικής Ανάλυσης είναι να χρησιμοποιεί την ίδια γεωμετρία καννάβου κατά το σχεδιασμό και την ανάλυση. Για να πραγματοποιηθεί αυτό χρησιμοποιούνται οι ίδιες συναρτήσεις κατάλληλες για την περιγραφή της γεωμετρίας στο στάδιο του σχεδιασμού και έπειτα της ανάλυσης.

Σημαντικό ρόλο στην εξέλιξη της μεθόδου έχει και η συνεχής εξέλιξη των υπολογιστικών συστημάτων. Με τη χρήση των υπολογιστικών μέσων ο σχεδιασμός μοντέλων γίνεται πιο ακριβής, η διόρθωση και βελτίωση των προσχεδίων είναι αρκετά εύκολη, και η αποθήκευση και μεταφορά δεδομένων πραγματοποιούνται με υψηλές ταχύτητες. Έτσι είναι εφικτή πλέον, η σχεδίαση και πιο σύνθετων μοντέλων, όπως γέφυρες μεγάλων ανοιγμάτων, αυτοκίνητα ή και ακόμα διαστημικά λεωφορεία. Όλα αυτά πραγματοποιούνται σε πολύ λιγότερο χρόνο, δίνοντας τη δυνατότητα στο μηχανικό να ασχολείται περισσότερο χρόνο στα επόμενα στάδια, κατά την ανάλυση του μοντέλου.

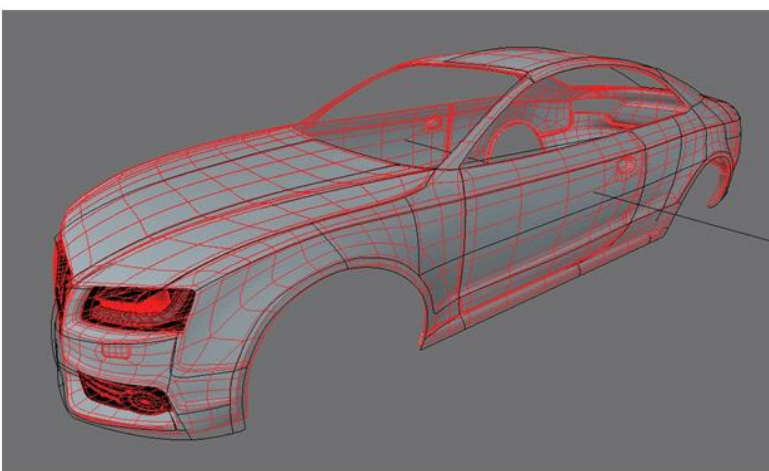
Συνεπώς, η Ισογεωμετρική ανάλυση δίνει τη δυνατότητα συσχέτισης μεταξύ του σχεδιασμού και της ανάλυσης πεπερασμένων στοιχείων, παρέχοντας, υψηλής ακρίβειας απεικόνιση της γεωμετρίας του μοντέλου. Επιπλέον, ο χρόνος που απαιτείται για τον σχεδιασμό μειώνεται αρκετά, καθώς πλέον δεν απαιτείται επαναληπτική διαδικασία για την πιστότερη απεικόνιση της αρχικής γεωμετρίας. Η Ισογεωμετρική ανάλυση βρίσκει εφαρμογή και βελτιώνει συνεχώς προβλήματα στον τομέα της υπολογιστικής μηχανικής, σε στατικές και δυναμικές αναλύσεις αλλά και στην βιομηχανική, και δίνει κίνητρο για περαιτέρω έρευνα και εξέλιξη της μεθόδου.



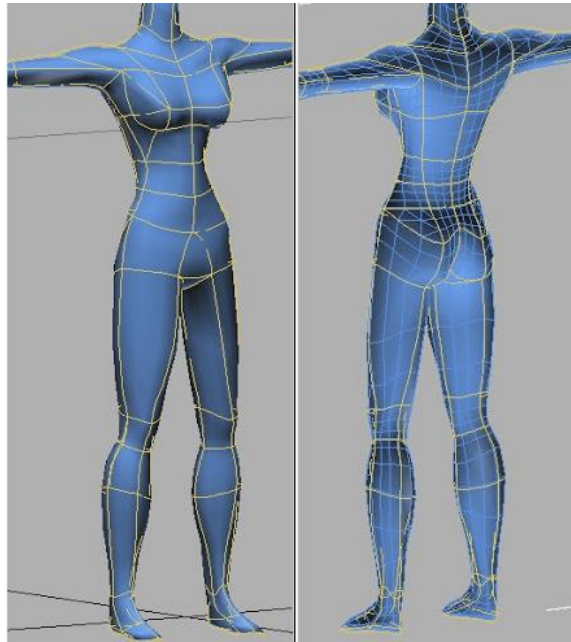
Σχήμα 5.1 Προσομοίωση κινητήρα αεροπλάνου με SPLines



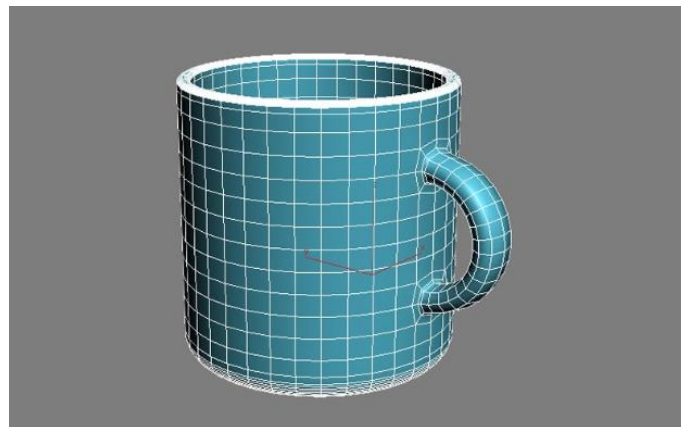
Σχήμα 5.2 Προσομοίωση αεροπλάνου με SPLines



Σχήμα 5.3 Προσομοίωση αυτοκινήτου με SPLines



Σχήμα 5.4 Προσομοίωση ανθρώπινου σώματος με SPLines



Σχήμα 5.5 Προσομοίωση κούπας με SPLines

### 3.2 Ιστορική Εξέλιξη

Η Ισογεωμετρική ανάλυση είναι μία καινοτόμος μέθοδος η οποία συνδυάζει τον ακριβή σχεδιασμό και την ανάλυση ενός φορέα. Η ιδέα προήλθε το 2003 από τον Thomas J.R Hughes, καθηγητή Αεροναυπηγικής και Τεχνικής Μηχανικής του πανεπιστημίου “University of Texas, Austin”. Το πρώτο βιβλίο που αφορά στην ανάπτυξη της μεθόδου, εκδόθηκε το 2009 από τους J. Austin Cottrell, Thomas J.R. Hughes, Yuri Bazilevs [14].

Ιστορικά, πριν την εξέλιξη των υπολογιστικών συστημάτων η σχεδίαση πιο περίπλοκων σχημάτων, όπως ελλείψεις και καμπύλες, ήταν αρκετά δύσκολη. Ιδιαίτερο

πρόβλημα αντιμετώπιζε ο τομέας της ναυπηγικής όπου απαιτούνταν τεράστια και περίπλοκα σχέδια. Τότε, για την σχεδίαση τέτοιων σχημάτων χρησιμοποιούνταν μακριά και εύκαμπτα δοκάρια από πλαστικό, μέταλλο ή και ξύλο, τα οποία ονομάζονταν *Splines*. Τα *Splines* διατηρούσαν τη θέση τους με τη βοήθεια βαριδιών, τα οποία ονομάζονταν *ducks*. Η καμπύλωση προέκυπτε από την θέση των βαριδιών και η τελική καμπύλη ήταν ομαλή.

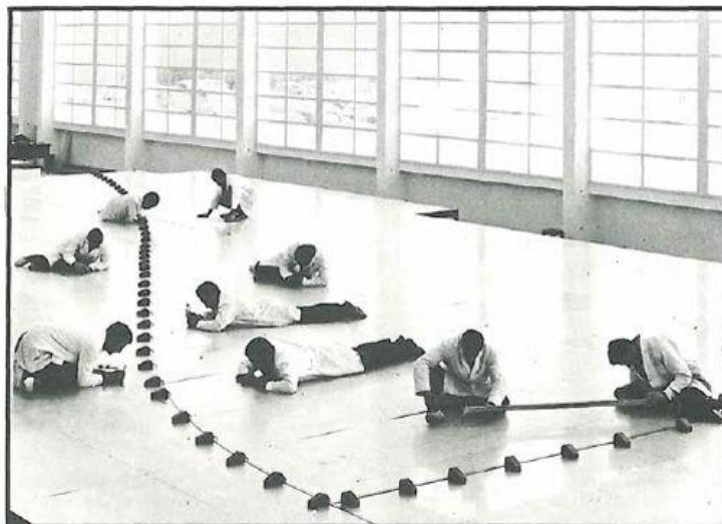
Η πρώτη προσέγγιση για τη σχεδίαση καμπύλων σχημάτων, έγινε το 1960 από τον Fergusson στην Boeing, για τη σχεδίαση τμημάτων επιφανειών αεροσκαφών. Τα τμήματα Fergusson είναι κυβικά πολυωνυμικά τμήματα που προσδιορίζονται από τα διανύσματα θέσης και τα εφαπτόμενα διανύσματα στα άκρα του τμήματος, χρησιμοποιώντας την παρεμβολή *Hermite*. Αυτό το σύστημα σχεδίασης, ονομάστηκε FMILL. Ωστόσο, βασικό μειονέκτημα της μεθόδου ήταν ότι κατά την ένωση διαδοχικών τμημάτων για τη δημιουργία καμπύλης, η συνέχειά της δεν εξαρτιόταν από τα διαδοχικά τμήματα αλλά από όλη την καμπύλη.

Το 1972, δημιουργήθηκε το σύστημα *UNISERF* από τον Γάλλο μαθηματικό Bézier. Ο Bézier εργαζόμενος στη Renault, χρησιμοποίησε τα πολώνυμα *Bernstein* ως συναρτήσεις μίξης και εισήγαγε τη μέθοδο περιγραφής των καμπυλών από το χαρακτηριστικό πολύγωνο, το λεγόμενο *πολύγωνο ελέγχου*. Οι καμπύλες Bézier αποτέλεσαν τη βάση για πολλά συστήματα CAD (Computer-Aided Design). Το μειονέκτημα της μεθόδου ήταν πως για τη σύνδεση τμημάτων, δημιουργούσε πολυωνυμικές καμπύλες μεγάλου βαθμού και το τελικό μοντέλο αποτελούνταν από πολλά επιμέρους τμήματα.

Το 1973 ο Riesenfeld και το 1974 ο Gordon, δημιούργησαν τις καμπύλες *B-SPLines*, οι οποίες αποτελούν μία γενίκευση των καμπυλών Bézier. Επομένως, οι καμπύλες *B-SPLines* έχουν όλες τις ιδιότητες των καμπυλών Bézier, αλλά συγχρόνως έχουν και κάποια επιπλέον προτερήματα, όπως η δυνατότητα τοπικού ελέγχου της καμπύλης, η ανεξαρτησία του βαθμού της καμπύλης από τα σημεία ελέγχου και η δυνατότητα προσθήκης ή αφαίρεσης σημείων ελέγχου χωρίς αντίστοιχη αλλαγή του βαθμού της καμπύλης.

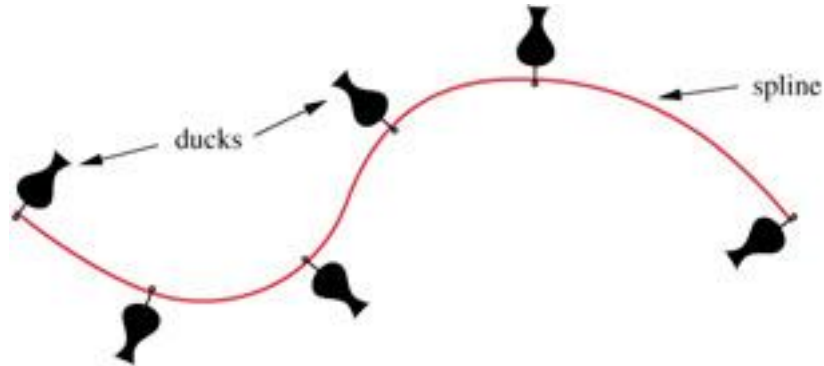
Ωστόσο, οι καμπύλες *B-SPLines* δεν μπορούν να παράγουν ακριβή απεικόνιση κωνικών τομών. Έτσι το 1975 ο Ken Versprille, χρησιμοποίησε τις *Non-Uniform Rational B-Splines (NURBS)*, που αποτελούν εξέλιξη των *B-SPLines* και επιτυγχάνουν μεγαλύτερη ακρίβεια στην απεικόνιση κωνικών τομών. [15].

Για τους σκοπούς της παρούσας διπλωματικής εργασίας, χρησιμοποιήθηκαν καμπύλες *B-SPLines* και *NURBS*, και γι' αυτό ακολουθεί αναλυτική αναφορά σε αυτές.





Σχήμα 5.6 *Loftsmen*: Μηχανικοί κατά την σχεδίαση με τη χρήση βαριδιών ducks [16]



Σχήμα 5.7 Σχέδιο με βαρίδια ducks για τη σχεδίαση καμπύλων Splines [17]



Σχήμα 5.8 Πραγματικά Splines- Ducks [18]

### 3.3 Καμπύλες B-SPLines

Οι καμπύλες B-SPLines, επινοήθηκαν από τον Nikolai Lobachevsky στις αρχές του 19ου αιώνα, στο πανεπιστήμιο της Ρωσίας “Kazan University” ενώ η ονομασία τους προέρχεται από τον Isaac Jacob Schoenberg και πρόκειται για συντομογραφία του Basis Spline (βασική καμπύλη). Πολλοί ερευνητές όπως οι Riesenfeld, Boehm και άλλοι, ασχολήθηκαν με την χρήση των B-SPLines για σχεδίαση με ηλεκτρονικούς υπολογιστές.

Οι καμπύλες B-SPLines αποτελούνται από περισσότερα του ενός τμημάτων, είναι τμηματικά πολυωνυμικές ή ρητές και ορίζονται από μία σειρά σημείων ελέγχου, τα σημεία *de Boor*, τα οποία σχηματίζουν το αντίστοιχο πολύγωνο ελέγχου, το πολύγωνο *de Boor*. Ο αριθμός των τμημάτων  $m$  εξαρτάται από τον αριθμό των σημείων ελέγχου  $(n+1)$  και τον βαθμό της καμπύλης  $p$ , ο οποίος από τον χρήστη. Παρακάτω, αναλύονται οι ορισμοί, οι βασικές ιδιότητες και οι αλγόριθμοι των B-SPLines [14], [19].

### 3.3.1 Index, Parameter and Physical Space

Σε πολλές περιπτώσεις, η εύρεση ακριβούς λύσεως είναι είτε αδύνατη είτε μη απαραίτητη. Η ακριβής γεωμετρική απεικόνιση μίας πραγματικής κατασκευής, σχεδιάζεται στο Καρτεσιανό σύστημα αναφοράς, το οποίο ονομάζεται “*Physical Space*”. Η απεικόνιση πιο πολύπλοκων κατασκευών, επιλέγεται να πραγματοποιηθεί σε έναν ιδεατό χώρο, στον οποίο οποιαδήποτε γεωμετρία μπορεί να αναπαρασταθεί με γραμμές, ορθογώνια και κύβους. Ο ιδεατός αυτός χώρος ονομάζεται “*Parameter Space*”. Υπάρχει ένας ακόμα βοηθητικός χώρος ο οποίος ονομάζεται “*Index Space*”, και είναι σημαντικός σε κάποιου είδους B-SPLines ενώ είναι απαραίτητος για τα NURBS.

#### Index Space

Η απεικόνιση του προσομοιώματος στον Index Space γίνεται βάσει των τιμών όλων των κόμβων του (*Knot Value Vector*). Ο Index Space περιγράφει την συμβολή κάθε κόμβου στο σχηματισμό μίας συνάρτησης B-SPLine.

Στην περίπτωση προσομοίωσης σε μία διάσταση, ο Index Space είναι μία γραμμή, στις δύο διαστάσεις απεικονίζεται με ορθογώνια ενώ στις τρεις διαστάσεις απεικονίζεται με κυβοειδή.

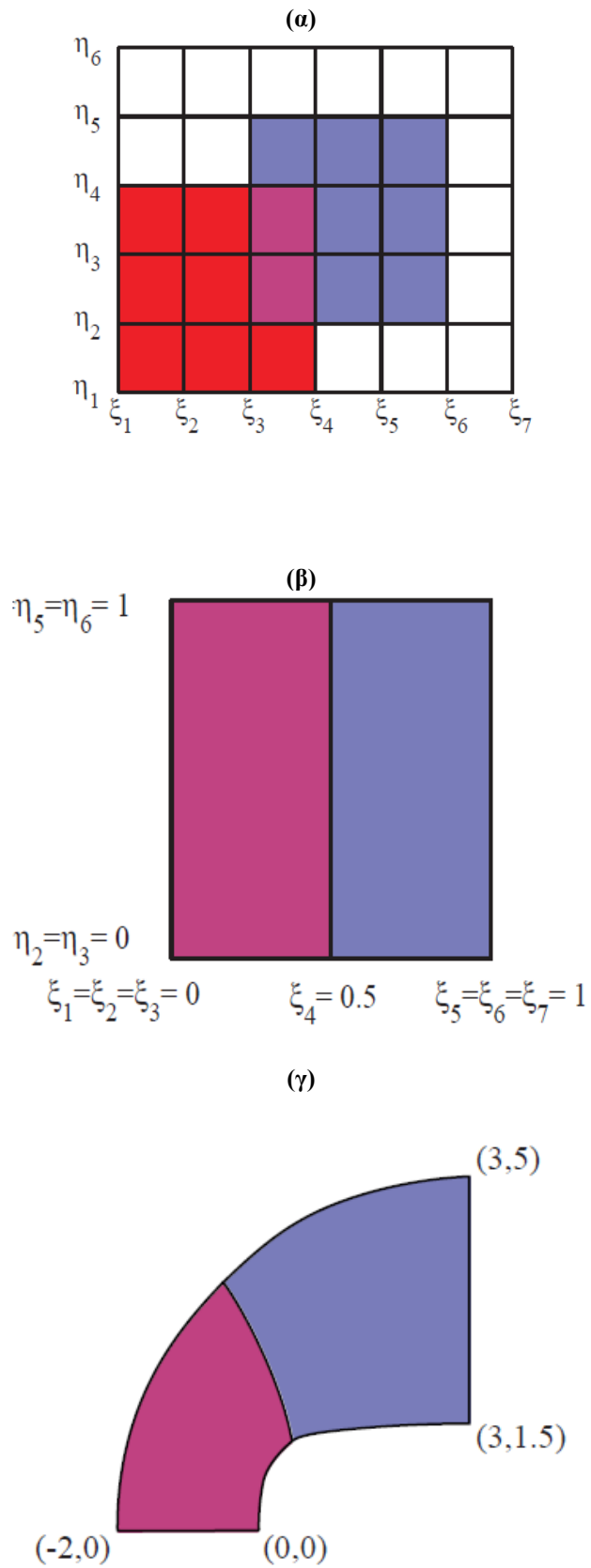
#### Parameter Space

Η απεικόνιση στον Παραμετρικό χώρο γίνεται βάσει των κόμβων (*knots*) του Knot Vector. Τα σημεία ελέγχου απεικονίζονται υπολογίζοντας τις παραμετρικές συντεταγμένες τους. Για το μετασχηματισμό της απεικόνισης από τον Φυσικό χώρο με άξονες (x, y, z) στον Παραμετρικό χώρο με άξονες (ξ, η, ζ), ή και αντίστροφα, απαιτείται η σύνθεση του μητρώου μετασχηματισμού *Jacobian*. Πρόκειται για έναν μετασχηματισμό σχημάτων πολύπλοκης γεωμετρίας σε πιο απλά σχήματα Έτσι, οι καμπύλες, οι επιφάνειες και τα στερεά του Φυσικού χώρου απεικονίζονται ως γραμμικά τμήματα, ορθογώνια και κυβοειδή στον Παραμετρικό χώρο.

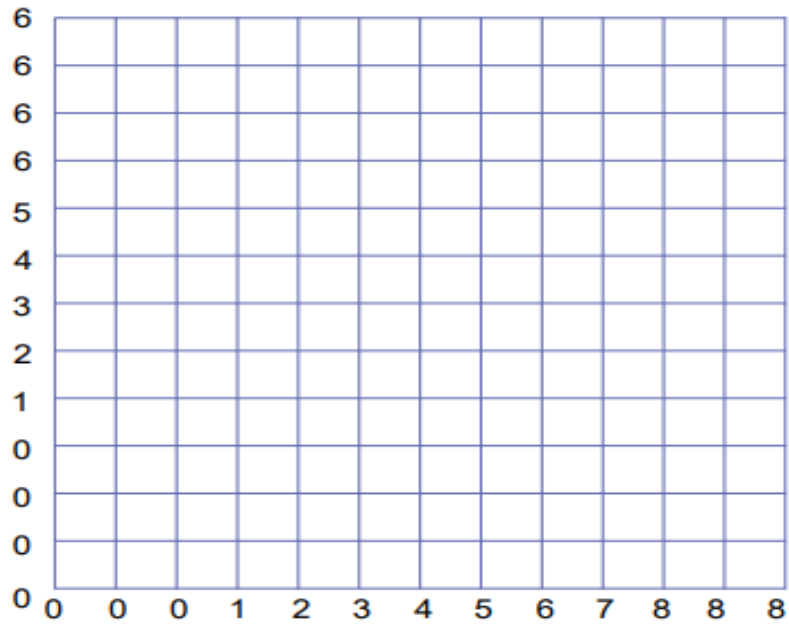
Η ιδέα απεικόνισης στον παραμετρικό χώρο υπάρχει ήδη από τη μέθοδο των Πεπερασμένων Στοιχείων, και συγκεκριμένα στα ισοπαραμετρικά στοιχεία. Ο τρόπος μετασχηματισμού είναι επομένως ήδη γνωστός.

#### Physical Space

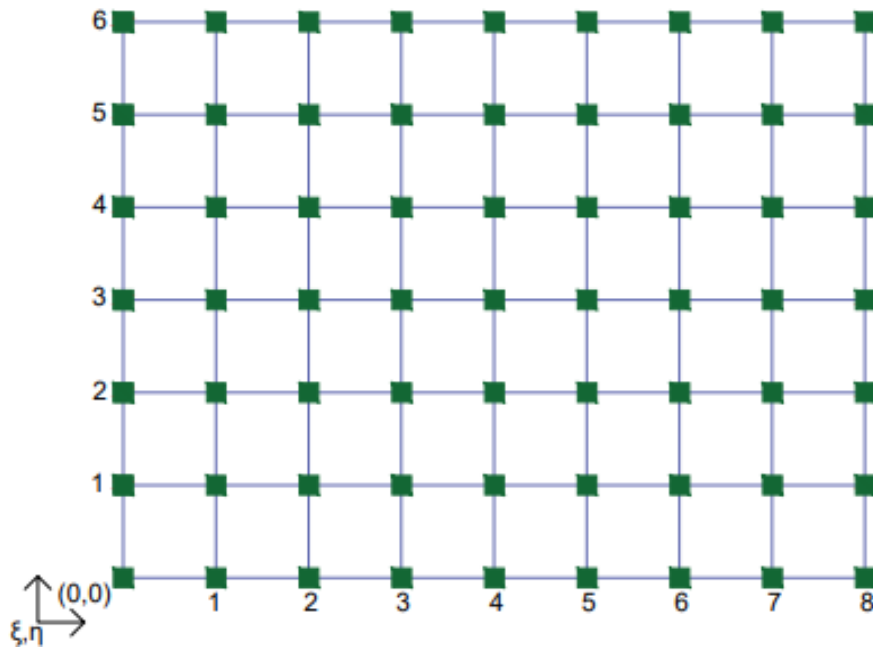
Ο Φυσικός χώρος είναι ο γνωστός Καρτεσιανός χώρος, όπου απεικονίζεται το πραγματικό προσομοίωμα. Στον Φυσικό χώρο απεικονίζονται και τα σημεία ελέγχου βάσει των καρτεσιανών συντεταγμένων τους χωρίς να είναι απαραίτητως σημεία του προσομοιώματος. Η απόσταση των σημείων ελέγχου από την γεωμετρία του προσομοιώματος εξαρτάται από τα βάρη τους, καθώς όσο αυξάνεται το βάρος ενός σημείου, τόσο αυτό πλησιάζει την πραγματική γεωμετρία. Αντιθέτως, στη μέθοδο Πεπερασμένων στοιχείων όλοι οι κόμβοι ανήκουν στον κάρναβο του προσομοιώματος.



Σχήμα 5.9 (α) Index Space - (β) Parameter Space – (γ) Physical Space [14]



(α)



(β)

Σχήμα 5.10 (α)Index Space - (β)Parameter Space  
 για Knot Value Vectors  $\Xi = \{0\ 0\ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 8\ 8\}$  και  $H = \{0\ 0\ 0\ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 6\ 6\ 6\}$   
 όπου  : οι κόμβοι των στοιχείων

### 3.3.2 Διάνυσμα κόμβων (Knot Value Vector)

Ένας από τους παράγοντες που επηρεάζουν τον υπολογισμό των βασικών συναρτήσεων και εν συνεχεία θα καθορίσουν το σχήμα της καμπύλης είναι ένα διάνυσμα τιμών, το οποίο ονομάζεται *Knot Vector*. Στην βιβλιογραφία το *Knot Vector* συμβολίζεται με  $\Xi$ , και οι συντεταγμένες του που ονομάζονται *Knots* (κόμβοι) και είναι οι θέσεις του παραμετρικού χώρου, συμβολίζονται με  $\xi_i$ . Κάθε *knot* είναι μοναδικό σε κάθε *Knot Vector*. Το σύνολο των *knots* είναι σε μία αύξουσα ακολουθία αριθμών ικανοποιώντας τη σχέση  $\xi_i \leq \xi_{i+1}$ . Ωστόσο, στην πραγματικότητα κάθε *Knot* μπορεί να επαναλαμβάνεται κι άλλες φορές, Ο αριθμός εμφάνισης κάθε *Knot* ονομάζεται πολλαπλότητα (multiplicity) ενώ το σύνολο με συντεταγμένες τα *Knots* με την αντίστοιχη πολλαπλότητά τους ονομάζεται *Knot Value Vector*. Επιπλέον, όταν η πρώτη και η τελευταία τιμή του *Knot Value Vector* έχουν πολλαπλότητα ίση με  $p+1$ , το διάνυσμα θεωρείται "Ανοικτό" (Open) καθώς τότε η συνάρτηση θα έχει στα άκρα της συνέχεια  $C^{-1}$ , κι έτσι δημιουργείται μία ανοικτή καμπύλη. Τέλος, οι τιμές  $\sigma'$  ένα διάνυσμα μπορεί να είναι τυχαίες ή να ισαπέχουν μεταξύ τους. Τότε, το *Knot Value Vector* θεωρείται ανομοιόμορφο (non-uniform) ή ομοιόμορφο (uniform) αντιστοίχως. Στην πραγματικότητα, δεν έχει σημασία η αριθμητική τιμή του *Knot*, αλλά η απόσταση που υπάρχει μεταξύ των *Knots*. Στην παρούσα διπλωματική εργασία, χρησιμοποιήθηκαν μόνο Open-Uniform *Knot Value Vectors*, και όσο αναφέρονται παρακάτω αφορούν αυτή την περίπτωση

Για παράδειγμα, καμπύλη πολυωνυμικού βαθμού  $p=2$  και με Open-Uniform *Knot Value Vector* = {0 0 0 1 2 3 4 4 5 5 5} και *Knot Values* τις τιμές  $\xi_i = 0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5$  έχει ως *Knot Vector*  $\Xi = \{0 1 2 3 4 5\}$  και *Knots* τις τιμές  $\xi_i = 0, 1, 2, 3, 4, 5$ .

Για τον υπολογισμό του αριθμού των *Knots* εφαρμόζεται η παρακάτω σχέση

$$\begin{aligned} (p+1) + (n+1) &= m+1 \\ \text{ή} & \\ m &= n + p + 1 \end{aligned} \tag{3.1}$$

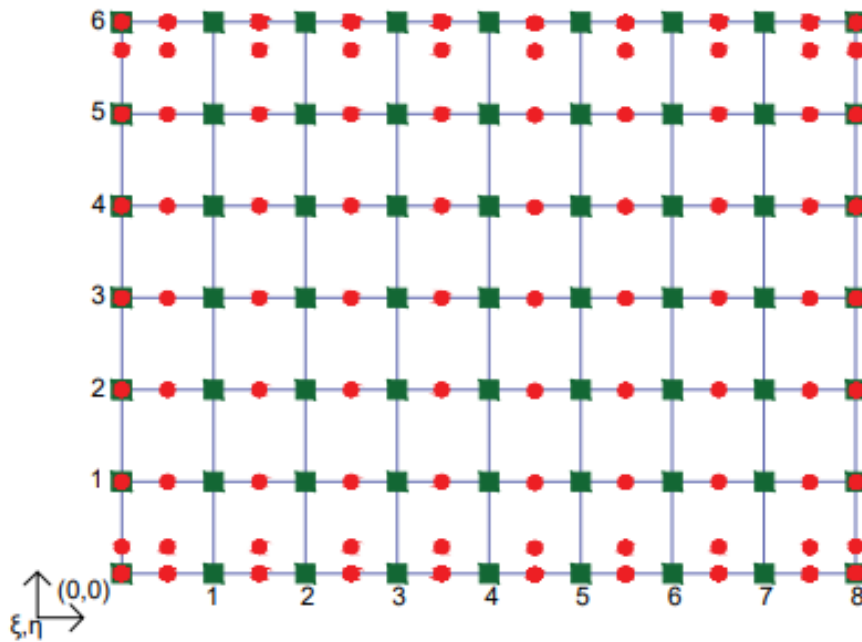
### 3.3.3 Σημεία Ελέγχου (Control Points)

Εκτός από τους κόμβους, για το σχεδιασμό μίας καμπύλης είναι απαραίτητο να προσδιοριστεί η πολυγωνική γραμμή της, η οποία προκύπτει από τον ένωση κάποιων σημείων τα οποία ονομάζονται σημεία ελέγχου (control points). Η ύπαρξη των control points προσφέρουν το πλεονέκτημα του τοπικού ελέγχου της καμπύλης. Δηλαδή, κάθε σημείο της καμπύλης επηρεάζεται από συγκεκριμένα control points, το πλήθος των οποίων καθορίζεται από τον πολυωνυμικό βαθμό. Έτσι, αλλάζοντας κάποια control points, επεξεργάζεται το αντίστοιχο τμήμα της καμπύλης και συγχρόνως χωρίς να αλλάζει η συνέχεια της καμπύλης.

Τα control points υπάρχουν και στους τρεις χώρους (Index, Parameter and Physical Spaces). Ο προσδιορισμός των παραμετρικών συντεταγμένων τους γίνεται στον Index Space, στον οποίο τα control points βρίσκονται στο κέντρο κάθε διαστήματος που επηρεάζει μία βασική συνάρτηση. Κάθε βασική συνάρτηση έχει επιρροή (support) σε  $p+1$  διαστήματα, τα οποία ονομάζονται *Knot Value Spans*.

Η συντεταγμένη ενός Control Point, υπολογίζεται από τη σχέση

$$\xi_{CP,i} = \sum_{k=1}^p \frac{\xi_{i+k}}{p}, \quad (3.2)$$



Σχήμα 5.11 Parameter Space για Knot Value Vectors  $\Xi = \{0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 8 \ 8\}$   
και  $H = \{0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 6 \ 6 \ 6\}$

όπου ■ : οι κόμβοι των στοιχείων  
● : οι θέσεις των Control points

### 3.3.4 Βασικές συναρτήσεις B-SPLines

Η καμπύλη B-SPLine βαθμού  $p$ , που ορίζεται από  $n+1$  σημεία ελέγχου  $P_i$ , με  $0 \leq i \leq n$ , υπολογίζεται σύμφωνα με την εξίσωση

$$C(\xi) = \sum_{i=1}^n P_i N_{i,p}(\xi), \quad 0 \leq \xi \leq \xi_{\max}, \quad 1 \leq p \leq n, \quad (3.3)$$

όπου  $N_{i,p}$  είναι οι βασικές συναρτήσεις B-SPLines της καμπύλης και υπολογίζονται από την αναδρομική σχέση *Cox-de Boor*, σύμφωνα με την οποία :

Για πολυωνυμικό βαθμό  $p = 0$

$$N_{i,0}(\xi) = \begin{cases} 1, & \alpha\nu \xi_i \leq \xi < \xi_{i+1} \\ 0, & \epsilon\kappa\acute{\omega}\varsigma \end{cases}, \quad (3.4)$$

και για πολυωνυμικό βαθμό  $p \geq 1$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad , \quad (3.5)$$

με την παραδοχή ότι ο λόγος  $\frac{0}{0} = 0$ .

### 3.3.5 Ιδιότητες βασικών συναρτήσεων B-SPLines

#### 1. Άθροισμα ίσο με τη μονάδα ( Partition of unity)

Σε οποιοδήποτε εύρος κόμβων  $[\xi_i, \xi_{i+1})$  το άθροισμα των βασικών συναρτήσεων είναι ίσο με τη μονάδα. Έτσι, η καμπύλη και το πολύγωνα ελέγχου παραμένουν αναλλοίωτα σε μετασχηματισμούς.

$$\sum_{i=1}^n N_{i,p}(\xi) = 1 \quad \forall \xi, p \quad ,$$

#### 2. Μη αρνητικότητα (Nonnegativity)

Κάθε βασική συνάρτηση παίρνει μόνο θετικές ή μηδενικές τιμές, το οποίο αποτελεί ιδιότητα του κυρτού περιβλήματος.

$$N_{i,p}(\xi) \geq 0 \quad \forall \xi, i, p \quad ,$$

#### 3. Οι βασικές συναρτήσεις $N_{i,p}(\xi)$ είναι μη μηδενικές στα $p+1$ διαστήματα κόμβων.

#### 4. Συνέχεια $C^{p-m}$

Η συνέχεια κάθε Knot εξαρτάται από την πολλαπλότητα του και υπολογίζεται από τη σχέση  $C^{p-m}$ . Τα ακραία Knots έχουν πάντα  $C^{-1}$  συνέχεια ενώ τα ενδιάμεσα δεν είναι δυνατό να έχουν κάτω από  $C^0$ . Γι' αυτό, πρέπει να επαναλαμβάνονται το πολύ μέχρι  $p$  φορές. Επίσης, όσο αυξάνεται η πολλαπλότητα ενός κόμβου, τόσο μειώνεται η συνέχειά του και τόσο πιο απότομη γίνεται η καμπύλη σε αυτό το σημείο. Έτσι, όταν συναντάται απότομη καμπύλη σημαίνει πως μία βασική συνάρτηση έχει μειωμένη συνέχεια.

#### 5. Full Tensor Product των B-SPLine Basis Functions

Οι βασικές συναρτήσεις B-SPLines που υπολογίστηκαν παραπάνω αφορούν καμπύλη B-SPLine σε μία διάσταση. Οι βασικές συναρτήσεις B-SPLines για καμπύλη δύο και περισσότερων διαστάσεων προκύπτουν από τον γραμμικό συνδυασμό των κατάλληλων βασικών συναρτήσεων κάθε διάστασης. Το αποτέλεσμα που προκύπτει ονομάζεται *Full Tensor Product*.

Έτσι, οι 2D B-SPLine βασικές συναρτήσεις προκύπτουν από τον συνδυασμό των αντίστοιχων 1D βασικών συναρτήσεων  $N_{i,p}(\xi)$  και  $M_{j,q}(\eta)$  ως εξής

$$R_{i,j}^{p,q}(\xi, \eta) = N_{i,p}(\xi) \cdot M_{j,q}(\eta), \quad (3.6)$$

Ομοίως, υπολογίζονται οι 3D B-Spline βασικές συναρτήσεις προκύπτουν από τον συνδυασμό των αντίστοιχων 1D βασικών συναρτήσεων  $N_{i,p}(\xi)$ ,  $M_{j,q}(\eta)$  και  $L_{k,r}(\zeta)$  ως εξής

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = N_{i,p}(\xi) \cdot M_{j,q}(\eta) \cdot L_{k,r}(\zeta), \quad (3.7)$$

Συνεπώς, οι ιδιότητες των 1D βασικών συναρτήσεων B-SPLines ισχύουν και για τις συναρτήσεις περισσότερων διαστάσεων.

### 3.3.6 Ιδιότητες καμπύλων B-SPLines

#### 1. Ιδιότητα Κυρτού Πολυγώνου

Η καμπύλη B-Spline περιλαμβάνεται πλήρως στο κυρτό πολύγωνο που δημιουργείται από τα Control Points της καμπύλης.

#### 2. Variation diminishing

Εάν μία καμπύλη B-Spline είναι επίπεδη, τότε ο αριθμός των σημείων που τέμνονται με την καμπύλη, είναι μικρότερος ή ίσος με τον αριθμό των σημείων τομής της γραμμής με το πολύγωνο

#### 3. Τοπικός έλεγχος καμπύλης

Κατά την τροποποίηση οποιουδήποτε Control Point της καμπύλης, επηρεάζονται μόνο τα  $p+1$  τμήματα και όχι όλη η καμπύλη  $N_{i,p}(\xi) = 0 \quad \forall \xi \notin [\xi_i, \xi_{i+p+1})$

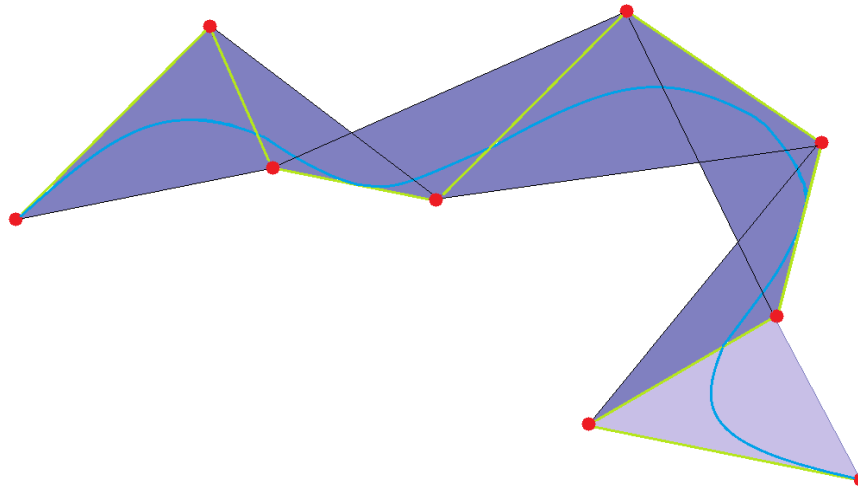
#### 4. Η ανοικτή καμπύλη περνάει από τα ακραία σημεία ελέγχου και εφάπτεται στα ακραία τμήματα του πολυγώνου ελέγχου.

#### 5. Γενικά, μία καμπύλη B-Spline $C(\xi)$ βαθμού $p$ , αποτελείται από $n$ επιμέρους τμήματα, όπου $n$ ο αριθμός των Control Points και το Knot Value Vector της περιέχει $m$ Knots. Όλα αυτά σχετίζονται με τη σχέση $m = n + p + 1$

#### 6. Περιβάλλουσα καμπύλης (Convex Hull)

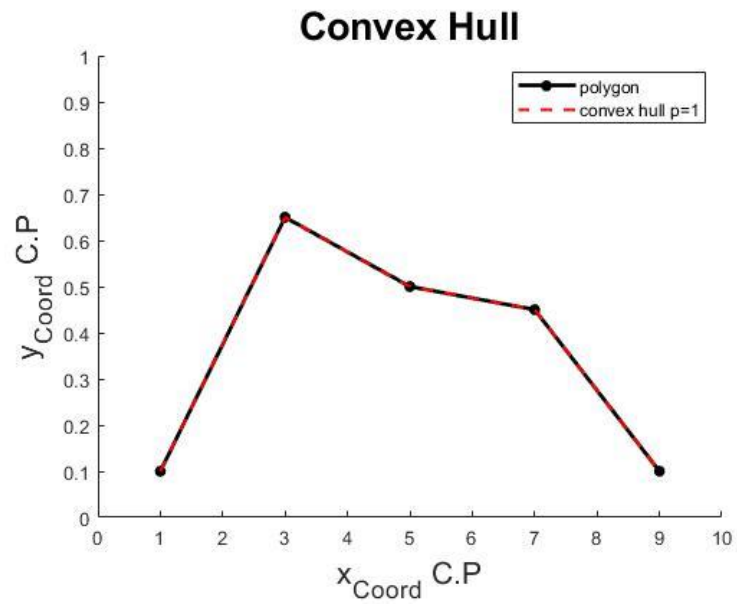
Για την δημιουργία μίας καμπύλης, καθορίζεται επιπλέον ένα πολύγωνο το οποίο πρέπει πάντα να περιέχει εντός του την καμπύλη. Το πολύγωνο αυτό, είναι η περιβάλλουσα της καμπύλης (*convex hull*) και σχηματίζεται από όλους τους πιθανούς γραμμικούς συνδυασμούς των control points. Κάθε τμήμα της καμπύλης έχει διαφορετικό Convex Hull, επιβεβαιώνοντας έτσι τον τοπικό χαρακτήρα κάθε τμήματος.



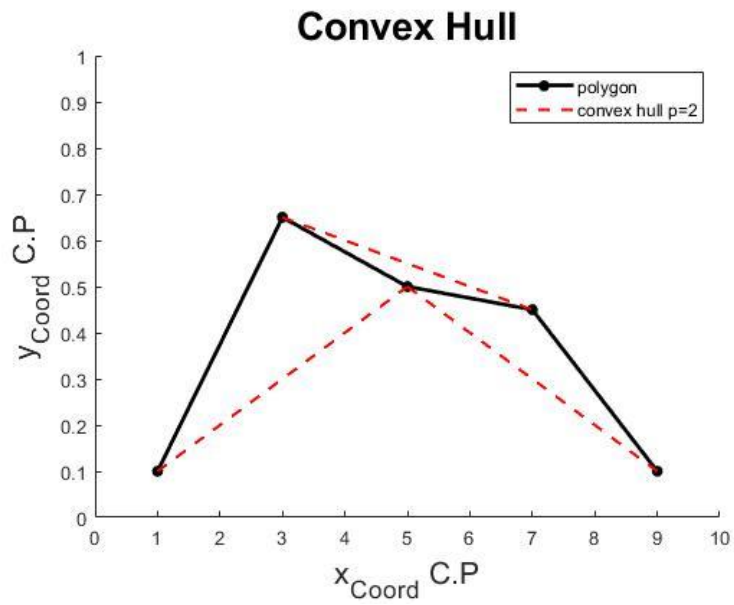


Σχήμα 5. 12 Convex Hull που εμπεριέχει την καμπύλη [20]

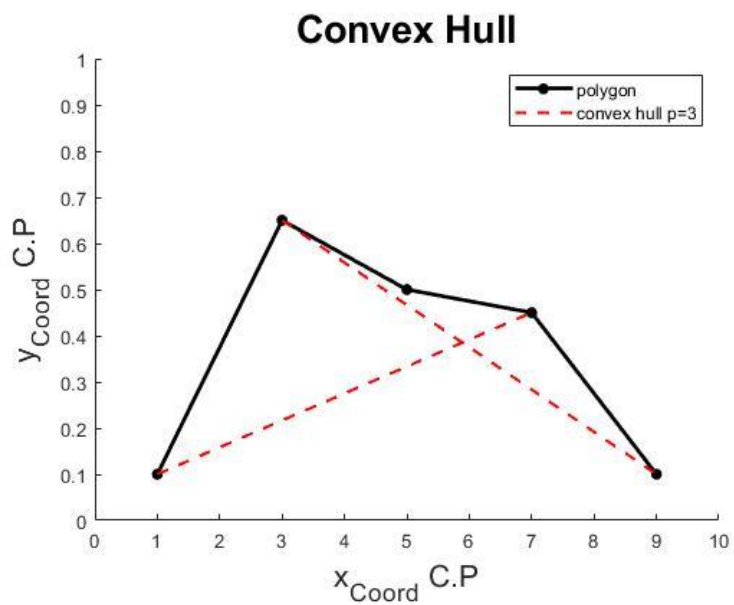
(α)



(β)



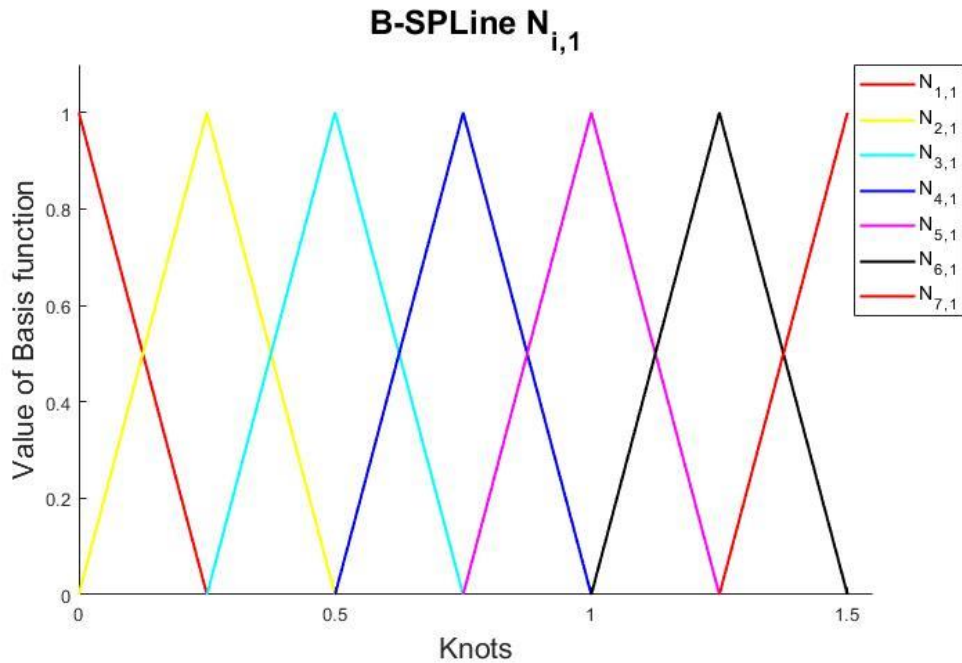
(γ)



Σχήμα 5.13 Πολύγωνο Ελέγχου και Convex Hull καμπύλης βαθμού (α) p = 1 (β) p = 2 (γ) p = 3

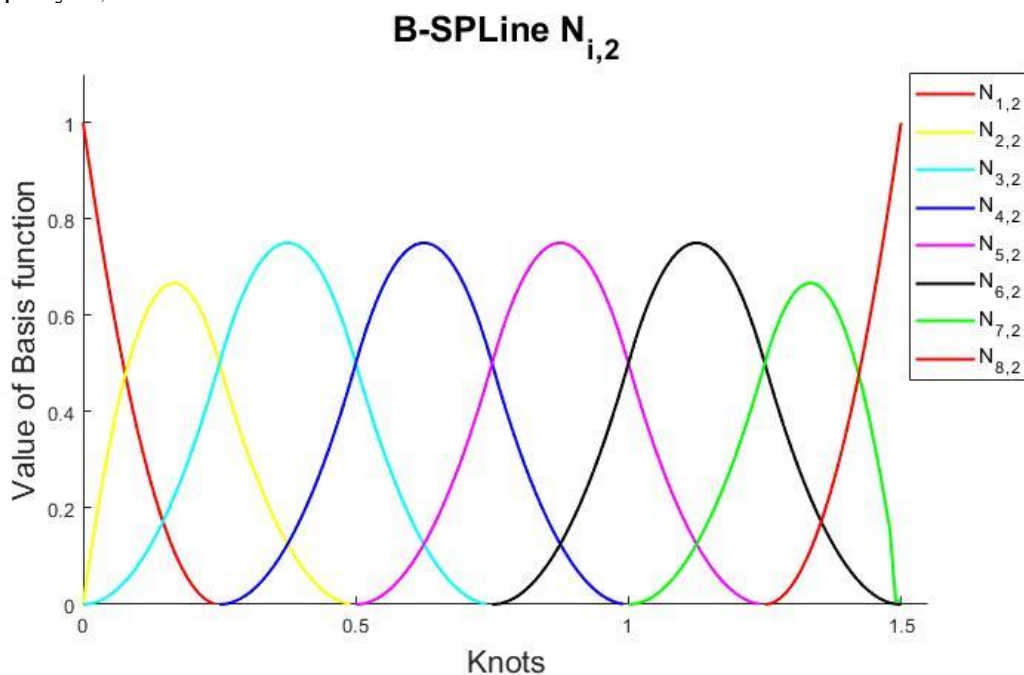
Για καλύτερη κατανόηση της μορφής και των ιδιοτήτων των καμπύλων B-SPLines που αναφέρθηκαν, ακολουθούν κάποια παραδείγματα για διάφορες περιπτώσεις του πολωνυμικού βαθμού p.

Έστω το Knot Value Vector =  $\{0 \ 0 \ 0.25 \ 0.50 \ 0.75 \ 1.00 \ 1.25 \ 1.50 \ 1.50\}$ . Αφού η πολλαπλότητα των ακραίων τιμών είναι ίση με  $k = 2$ , η καμπύλη είναι βαθμού  $p = k - 1 = 1$ . Το K.V.V αποτελείται από  $m = 9$  Knots, άρα η καμπύλη σχηματίζεται από  $n = m - p - 1 = 7$  control points. Στο παρακάτω διάγραμμα παρουσιάζονται οι βασικές συναρτήσεις  $N_{i,1}$ .



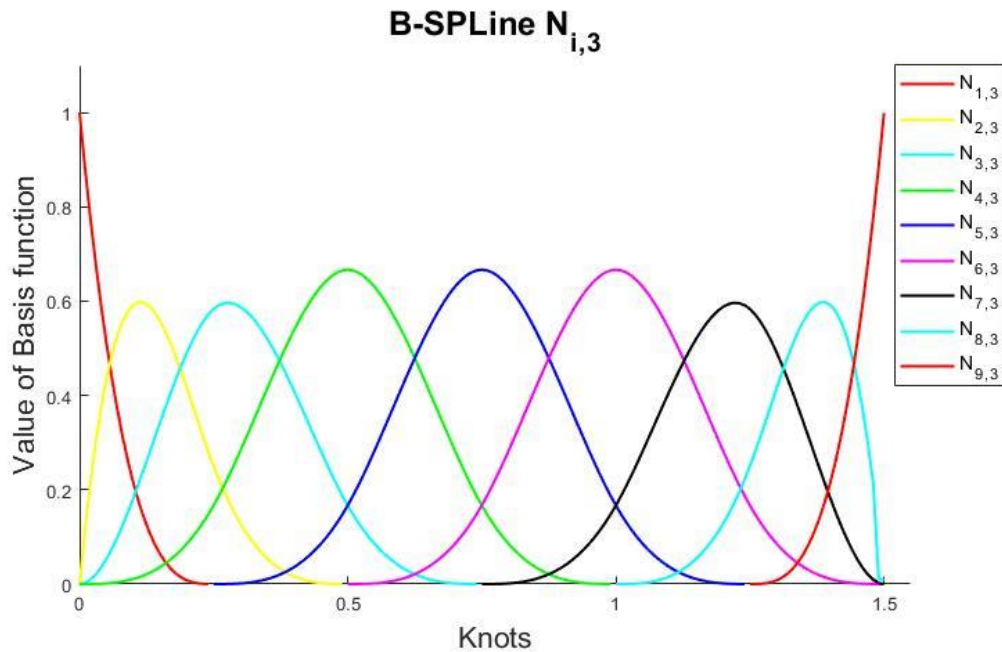
Σχήμα 5.14 Βασικές συναρτήσεις για καμπύλη βαθμού  $p=1$

Έστω το Knot Value Vector =  $\{0 \ 0 \ 0 \ 0.25 \ 0.50 \ 0.75 \ 1.00 \ 1.25 \ 1.50 \ 1.50 \ 1.50\}$ . Άρα  $p = 2$ ,  $m = 11$  Knots, άρα η καμπύλη σχηματίζεται από  $n = 8$  Control Points. Έτσι, οι βασικές συναρτήσεις  $N_{i,2}$  είναι



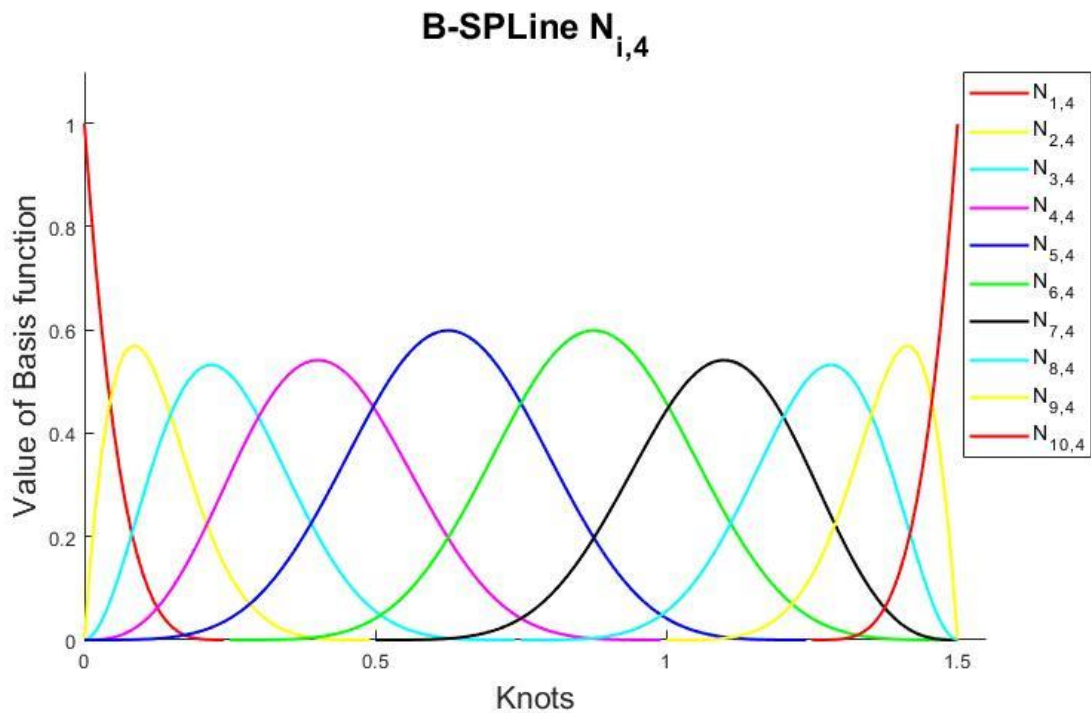
Σχήμα 5.15 Βασικές συναρτήσεις για καμπύλη βαθμού  $p=2$

Έστω το Knot Value Vector =  $\{0 \ 0 \ 0 \ 0 \ 0.25 \ 0.50 \ 0.75 \ 1.00 \ 1.25 \ 1.50 \ 1.50 \ 1.50 \ 1.50\}$ . Άρα  $p = 3$ ,  $m = 13$  Knots και  $n = 9$  Control Points. Έτσι, οι βασικές συναρτήσεις  $N_{i,3}$  είναι



Σχήμα 5.16 Βασικές συναρτήσεις για καμπύλη βαθμού  $p=3$

Έστω το Knot Value Vector =  $\{0 \ 0 \ 0 \ 0 \ 0 \ 0.25 \ 0.50 \ 0.75 \ 1.00 \ 1.25 \ 1.50 \ 1.50 \ 1.50 \ 1.50\}$ . Άρα  $p = 4$ ,  $m = 15$  Knots και  $n = 10$  Control Points. Έτσι, οι βασικές συναρτήσεις  $N_{i,4}$  είναι



Σχήμα 5.17 Βασικές συναρτήσεις για καμπύλη βαθμού  $p=4$

### 3.3.7 Παράγωγοι των βασικών συναρτήσεων ( B-Spline Basis Function Derivatives)

Στην Ισογεωμετρική Ανάλυση είναι απαραίτητος ο υπολογισμός και των παραγώγων των βασικών συναρτήσεων καθώς αυτές χρησιμοποιούνται κατά κύριο λόγο στην ανάλυση. Συγκεκριμένα, τα μητρώα Παραμορφώσεων και Ακαμψίας συντίθενται από τις παραγώγους των συναρτήσεων σχήματος.

Ο υπολογισμός των παραγώγων των συναρτήσεων γίνεται ξανά σύμφωνα με τον αναδρομικό τύπο *Cox-de Boor*

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad , \quad (3.8)$$

Επαναλαμβάνοντας τον παραπάνω τύπο  $k$  φορές, προκύπτει ο γενικός τύπος υπολογισμού της  $k$ -ιοστής παραγώγου  $N_{i,p}^{(k)}(\xi)$  μίας συνάρτησης  $N_{i,p}(\xi)$

$$N_{i,p}^{(k)} = \frac{p!}{(p-k)!} \sum_{j=0}^k a_{k,j} N_{i+j,p-k} \quad , \quad (3.9)$$

με

$$\begin{aligned} a_{0,0} &= 1 \\ a_{k,0} &= \frac{\alpha_{k-1,0}}{\xi_{i+p-k+1} - \xi_i} \\ a_{k,j} &= \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{i+p+j-k+1} - \xi_{i+j}} \quad j = 1, \dots, k-1 \\ a_{k,k} &= \frac{-\alpha_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}} \end{aligned}$$

όπου το  $k$  δεν έχει πρακτικά σημασία για τιμές μεγαλύτερες του  $p$ , καθώς όλοι οι παράγωγοι βαθμού μεγαλύτερου του  $p$  είναι μηδενικές.

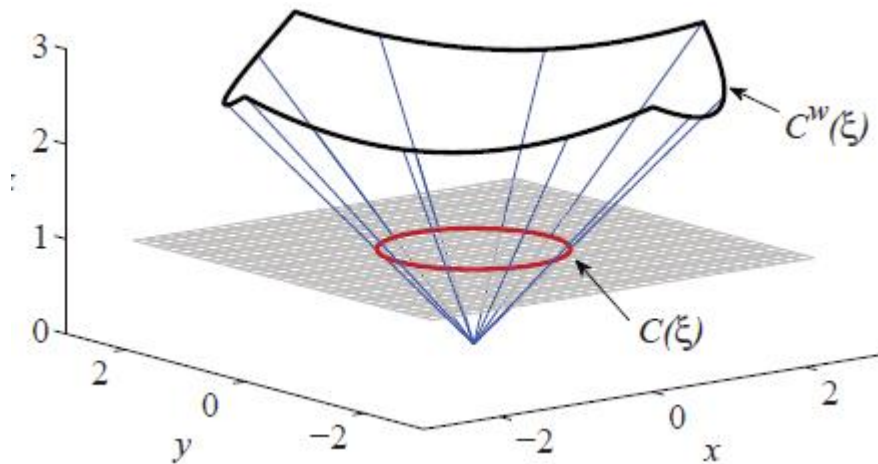
### 3.4 Μη ομοιόμορφες ρητές καμπύλες B-SPLines (Non Uniform Rational B-SPLines - NURBS)

Με τη χρήση των καμπύλων B-Spline αν και πραγματοποιήθηκε αρκετά ακριβής προσομοίωση ακόμα και των πιο περίπλοκων σχημάτων, παρουσιάζουν κάποιες αδυναμίες στην αναπαράσταση κάποιων καμπυλών, όπως του κύκλου και γενικά των κωνικών τομών. Στο πλαίσιο των προγραμμάτων σχεδίασης με ηλεκτρονικούς υπολογιστές για τη γεωμετρική αναπαράσταση των καμπυλών των οποίων η αναπαράσταση δεν είναι δυνατή με πολώνυμα, εισήχθησαν οι μη ομοιόμορφες ρητές καμπύλες *B-SPLines (Non Uniform Rational B-SPLines – NURBS)* [20].

Οι καμπύλες NURBS προκύπτουν από την τομή του κώνου με ένα επίπεδο και δίνουν καλύτερη απεικόνιση των κωνικών τομών. Οι καμπύλες B-SPLines  $n$ -διαστάσεων, προκύπτουν από την προβολή καμπύλων NURBS διαστάσεων  $(n+1)$ . Έτσι, μία καμπύλη

B-SPLine  $C^w(\xi)$  με 3D Control Points  $\{X_i^w, Y_i^w, Z_i^w\}$  προβάλεται στο επίπεδο  $z=1$  και προκύπτει η αντίστοιχη καμπύλη NURBS  $C(\xi)$  με τα 2-D Control Points  $\{X_i, Y_i\}$  και το βάρος (Weight) των NURBS είναι  $W = \{Z^w\}$ ,

$$\text{όπου } \{X_i, Y_i\} = \left\{ \frac{X_i^w}{Z_i^w} \right\}$$



Σχήμα 5.18 Προβολή καμπύλης B-SPLine για παραγωγή της αντίστοιχης καμπύλης NURBS [14]

### 3.4.1 Συναρτήσεις σχήματος NURBS

Για τον υπολογισμό των καμπύλων NURBS, πρέπει να προσδιοριστεί η συνάρτηση βάρους  $W(\xi)$ .

$$W(\xi) = \sum_{i=1}^n \{N_{i,p}(\xi) \cdot w_i\}, \quad (3.10)$$

όπου  $w_i$ , τα βάρη των Control Points.

Ο υπολογισμός των συναρτήσεων σχήματος NURBS σε μία διάσταση, γίνεται ως εξής

$$R_i^p(\xi) = \frac{N_{i,p}(\xi) \cdot w_i}{W(\xi)} = \frac{N_{i,p}(\xi) \cdot w_i}{\sum_{i=1}^n \{N_{i,p}(\xi) \cdot w_i\}}, \quad (3.11)$$

Αντιστοίχως, όπως οι βασικές συναρτήσεις B-SPLines δύο ή περισσότερων διαστάσεων προκύπτουν από τον κατάλληλο συνδυασμό των 1D βασικών συναρτήσεων B-SPLines, έτσι και οι βασικές συναρτήσεις NURBS δύο ή περισσότερων διαστάσεων προκύπτουν από τον κατάλληλο συνδυασμό των 1D βασικών συναρτήσεων NURBS.

Για τον υπολογισμό των 2D βασικών συναρτήσεων NURBS και της αντίστοιχης συνάρτησης βάρους

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta) \cdot w_{ij}}{\sum_{i'=1}^n \sum_{j'=1}^m \{N_{i',p}(\xi) \cdot M_{j',q}(\eta) \cdot w_{i'j'}\}}, \quad (3.12)$$

$$W(\xi, \eta) = \sum_{i'=1}^n \sum_{j'=1}^m \{N_{i',p}(\xi) \cdot M_{j',q}(\eta) \cdot w_{i'j'}\}, \quad (3.13)$$

### 3.4.2 Ιδιότητες καμπύλων NURBS

1. Ιδιότητα κυρτού πολυγώνου

Η καμπύλη εμπεριέχεται στο κυρτό πολύγωνο που έχει οριστεί από τα Control Points.

2. Δεν μεταβάλλονται από τους γεωμετρικούς μετασχηματισμούς.
3. Ο αριθμός τομών επιπέδου (ή ευθείας) με την καμπύλη, είναι μικρότερος ή ίσος από τον αριθμό τομών του επιπέδου με το πολύγωνο ελέγχου.
4. Διέρχεται από τα ακραία σημεία έτσι ώστε  $C(0)=P_0$  και  $C(1)=P_n$ .
5. Μία ρητή καμπύλη χωρίς εσωτερικούς κόμβους είναι μία καμπύλη Bézier.
6. Τοπικός έλεγχος

Κατά την μετατόπιση ενός Control Point  $P_i$ , ή κατά την μεταβολή του βάρους του Control Point, επηρεάζεται μόνο το τμήμα της καμπύλης  $[\xi_i, \xi_{i+p+1})$ .

7. Στα εσωτερικά σημεία της καμπύλης υπάρχει η παράγωγος οποιουδήποτε βαθμού ενώ στους κόμβους υπάρχει η παράγωγος  $k-p$  βαθμού, με  $k$  την πολλαπλότητα του κόμβου.

Γενικά, οι πολυωνυμικές καμπύλες B-Spline είναι μία ειδική περίπτωση των ρητών καμπύλων B-Spline γι' αυτό και πολλές από τις ιδιότητες είναι κοινές.

### 3.4.3 Ιδιότητες συναρτήσεων NURBS

1. Άθροισμα ίσο με τη μονάδα ( Partition of unity)

Όλες οι ρητές συναρτήσεις στο διάστημα  $[0,1]$  έχουν σταθερό άθροισμα που είναι ίσο με τη μονάδα.

$$\sum_{i=1}^n R_{i,p}(\xi) = 1 \quad \forall i, p \quad ,$$

2. Μη αρνητικότητα (Nonnegativity)

Για όλα τα ζεύγη  $i, n$  και με  $\xi \in [0,1]$ , οι συναρτήσεις NURBS  $R_{i,n}(\xi)$  παίρνουν θετικές ή αρνητικές τιμές.

$$R_{i,p}(\xi) \geq 0 \quad \forall i, p \quad ,$$

3. Διαφορισιμότητα

Όλες οι παράγωγοι της  $R_{i,p}(\xi)$  υπάρχουν στο εσωτερικό ενός διαστήματος που είναι μη μηδενική και σε κάθε κόμβο η  $R_{i,p}(\xi)$  παραγωγίζεται  $p-k$  φορές όπου  $k$  η πολλαπλότητα του αντίστοιχου κόμβου.

4. Όταν  $w_i=1$ , για κάθε  $i$ , τότε οι NURBS ταυτίζονται με τις B-SPLines κι έτσι  $R_{i,p}(\xi) = N_{i,p}(\xi)$ .

5. Η  $R_{i,p}(\xi)$  έχει ένα μέγιστο στο διάστημα  $[0,1]$ .

6. Τοπικότητα

Όταν το  $\xi$  δεν ανήκει στο διάστημα  $[\xi_i, \xi_{i+p+1})$ , η  $R_{i,p}(\xi)=0$ . Έτσι, σε κάθε διάστημα κόμβων υπάρχουν  $p+1$  μη μηδενικές συναρτήσεις  $R_{i,p}(\xi)$  οι οποίες είναι οι  $R_{i-p,p}(\xi), \dots, R_{i,p}(\xi)$  για το  $[\xi_i, \xi_{i+p+1})$ .

**3.4.4 Παράγωγοι των συναρτήσεων σχήματος NURBS**

Όπως οι βασικές συναρτήσεις σχήματος NURBS προκύπτουν από τις βασικές συναρτήσεις B-SPLines, έτσι αντίστοιχα και οι παράγωγοι των συναρτήσεων NURBS προκύπτουν από τις συναρτήσεις B-SPLines και τις παραγώγους τους.

Συγκεκριμένα, για την εύρεση της 1D συνάρτησης παραγώγου χρησιμοποιείται η σχέση

$$\frac{d}{d\xi} R_i^p(\xi) = w_i \cdot \frac{\left( \frac{d}{d\xi} N_{i,p}(\xi) \right) \cdot W(\xi) - \left( \frac{d}{d\xi} W(\xi) \right) \cdot N_{i,p}(\xi)}{(W(\xi))^2}, \quad (3.14)$$

$$\frac{d}{d\xi} W(\xi) = \sum_{i=1}^n \left\{ \frac{d}{d\xi} N_{i,p}(\xi) \cdot w_i \right\}, \quad (3.15)$$



Για την 2D συνάρτηση παραγώγου, υπολογίζονται οι παράγωγοι ως προς και τους δύο άξονες

$$\frac{\partial}{\partial \xi} R_{i,j}^{p,q}(\xi, \eta) = w_{ij} \cdot \frac{\left( \frac{d}{d\xi} N_{i,p}(\xi) \right) \cdot M_{j,q}(\eta) \cdot W(\xi, \eta) - \left( \frac{\partial}{\partial \xi} W(\xi, \eta) \right) \cdot N_{i,p}(\xi) \cdot M_{j,q}(\eta)}{(W(\xi, \eta))^2}, \quad (3.16)$$

$$\frac{\partial}{\partial \eta} R_{i,j}^{p,q}(\xi, \eta) = w_{ij} \cdot \frac{N_{i,p}(\xi) \cdot \left( \frac{d}{d\eta} M_{j,q}(\eta) \right) \cdot W(\xi, \eta) - \left( \frac{\partial}{\partial \eta} W(\xi, \eta) \right) \cdot N_{i,p}(\xi) \cdot M_{j,q}(\eta)}{(W(\xi, \eta))^2}, \quad (3.17)$$



## 4 Μητρώο Στιβαρότητας

### 4.1 Βήματα για την μόρφωση του Μητρώου Στιβαρότητας

Για την πραγματοποίηση της Ισογεωμετρικής ανάλυσης, όπως και σε κάθε υπολογιστική μέθοδο, απαιτείται η μόρφωση του μητρώου στιβαρότητας του φορέα. Η διαδικασία είναι παρόμοια με αυτή που ακολουθείται στην μόρφωση μητρώου στιβαρότητας στην ανάλυση με Πεπερασμένα Στοιχεία με κύρια διαφορά το είδος των συναρτήσεων σχήματος που χρησιμοποιούνται για την προσέγγιση του πεδίου λύσης του προβλήματος.

#### 4.1.1 Συναρτήσεις Σχήματος

Στη μέθοδο Πεπερασμένων Στοιχείων οι συναρτήσεις σχήματος είναι πολυώνυμα, όπως τα Πολυώνυμα Lagrange, αλλά εμφανίζουν μειονεκτήματα ακρίβειας λύσης όπως ήδη αναφέρθηκε. Στην Ισογεωμετρική ανάλυση, για τις συναρτήσεις σχήματος χρησιμοποιούνται συναρτήσεις B-PLines (NURBS προκειμένω) δίνοντας καλύτερη προσέγγιση του φυσικού προβλήματος, καθώς υπάρχει αλληλεπίδραση μεταξύ των κοντινών στοιχείων.

#### 4.1.2 Control Points

Στην μέθοδο Πεπερασμένων στοιχείων οι κόμβοι ενός στοιχείου βρίσκονται στα άκρα του ή και στο μέσο κάθε πλευράς. Οι άγνωστοι του προβλήματος είναι οι βαθμοί ελευθερίας σε αυτούς τους κόμβους. Αφού υπολογιστούν οι επικόμβιες μετακινήσεις, με τη χρήση των συναρτήσεων σχήματος μπορούν να υπολογιστούν οι μετακινήσεις σε οποιοδήποτε σημείο.

Στην Ισογεωμετρική ανάλυση η απεικόνιση ενός στοιχείου προκύπτει από τον συνδυασμό των Control Points και των αντίστοιχων συναρτήσεων σχήματος και οι άγνωστοι του προβλήματος είναι οι βαθμοί ελευθερίας στις θέσεις των Control Points.

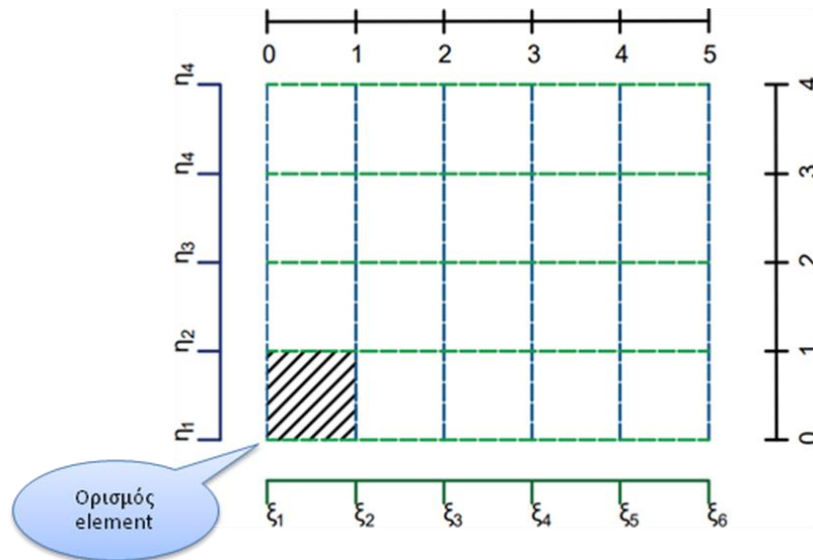
Ο υπολογισμός των Control Points γίνονται χωριστά σε κάθε άξονα και εν συνεχεία συνδυάζονται καταλλήλως για να σχηματιστούν τα Control Points κάθε στοιχείου. Ο υπολογισμός τους γίνεται ως εξής

$$\xi_{CP,i} = \sum_{k=1}^p \frac{\xi_{i+k}}{p}, \quad (4.1)$$

και στη συνέχεια πολλαπλασιάζονται με το αντίστοιχο βάρος του  $w_{CP,i}$  ώστε να αφορούν Control Points των NURBS.

### 4.1.3 Elements

Ο σχηματισμός των elements γίνεται με τον συνδυασμό κάθε μη μηδενικού διαστήματος του Knot Vector, το οποίο ονομάζεται *Knot Span*, της μίας διεύθυνσης με τα Knot Spans των άλλων διευθύνσεων. Οι συναρτήσεις σχήματος δεν περιορίζονται μόνο στο element στο οποίο ανήκουν, αλλά παίρνουν τιμές μη μηδενικές για τα  $p+1$  επόμενα Knot spans.



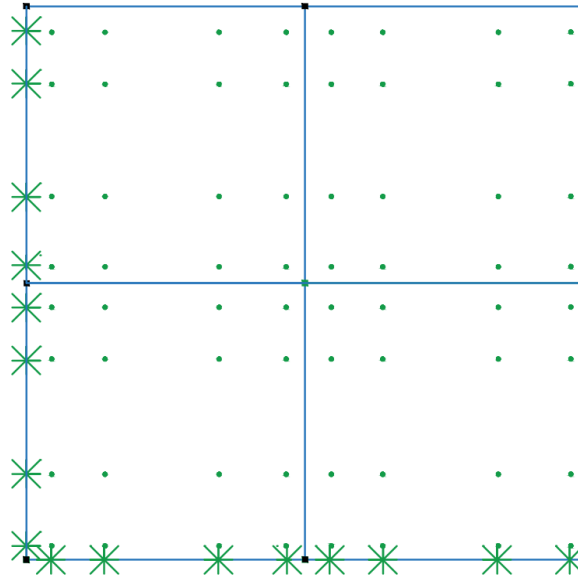
Σχήμα 4. 1 Ορισμός 2D element στην IGA

### 4.1.4 Gauss Points

Αφού καθοριστούν τα στοιχεία που συνθέτουν τον φορέα προσδιορίζονται τα Gauss Points κάθε στοιχείου. Συγκεκριμένα καθορίζονται τα Gauss Points των Knot spans σε κάθε διεύθυνση του στοιχείου και στην συνέχεια συνδυάζονται κατάλληλα. Ο αριθμός των Gauss Points που περιλαμβάνονται σε ένα Knot Span μίας διεύθυνσης είναι ίσος με  $n = p+1$ , όπου  $p$  ο βαθμός του πολωνύμου στην αντίστοιχη διεύθυνση. Τα Gauss points και τα αντίστοιχα βάρη τους παίρνουν τιμές στο διάστημα  $[-1,1]$  ενώ τα Knot spans μπορούν να έχουν οποιοσδήποτε τιμές. Γι' αυτό απαιτείται ο κατάλληλος μετασχηματισμός των συντεταγμένων των Gauss Points  $\xi^{GP}$  και των αντίστοιχων βαρών  $w^{GP}$  ώστε να περιλαμβάνονται στο επιθυμητό διάστημα  $[\xi_i, \xi_{i+1})$ . Ο μετασχηματισμός πραγματοποιείται ως εξής

$$\xi = \frac{(\xi_{i+1} - \xi_i) \cdot \xi^{GP}}{2} + \frac{(\xi_i + \xi_{i+1})}{2}, \quad (4.2)$$

$$w^\xi = \frac{(\xi_{i+1} - \xi_i)}{2} \cdot w^{GP^\xi}, \quad (4.3)$$



Σχήμα 4. 2 Gauss Points για  $\Xi = \{0\ 0\ 0\ 1\ 2\ 2\ 2\}$  και  $H = \{0\ 0\ 0\ 1\ 2\ 2\ 2\}$   
όπου τα πράσινα σημεία είναι τα Gauss Points του στοιχείου

## 4.2 Μόρφωση Μητρώου Στιβαρότητας

### 4.2.1 Μητρώο Στιβαρότητας 1D

Τα προβλήματα μονοδιάστατης ελαστικότητας δεν έχουν τόσο πρακτική εφαρμογή, αλλά χρησιμοποιούνται κυρίως για ακαδημαϊκούς λόγους καθώς είναι πιο εύκολη η κατανόηση τους. Μία απλή περίπτωση της μόρφωσης μητρώου στιβαρότητας για ένα πρόβλημα 1D, είναι το δικτύωμα με αξονική ένταση του οποίου όλοι οι βαθμοί ελευθερίας θα είναι παράλληλοι ως προς τον άξονα  $\xi$ .

Επομένως, ο φορέας παραμορφώνεται μόνο αξονικά και το μητρώο παραμόρφωσης του φορέα θα είναι

$$\left\{ \varepsilon \right\}_{(1 \times 1)} = [\varepsilon_x] = \left[ \frac{\partial u}{\partial x} \right], \quad (4.4)$$

Η μετακίνηση στον άξονα  $x$  ορίζεται ως  $u(x) = u(C(\xi)) = u(\xi)$  ενώ για τον υπολογισμό της παραγώγου της  $u$  προηγείται ο μετασχηματισμός από τον Physical Space στον Parameter Space. Ο μετασχηματισμός μίας συνάρτησης  $\varphi$  από τον Physical Space στον Parameter Space και αντιστρόφως γίνεται μέσω του Ιακωβιανού μητρώου  $[J]$ .

$$\frac{\partial \varphi}{\partial x} = \frac{\partial \varphi}{\partial \xi} \frac{\partial \xi}{\partial x}$$

$$\frac{\partial \varphi}{\partial \xi} = \frac{\partial \varphi}{\partial x} \frac{\partial x}{\partial \xi}$$

$$\left[ \frac{\partial \varphi}{\partial \xi} \right] = [J] \cdot \left[ \frac{\partial \varphi}{\partial x} \right], \quad (4.5)$$

Το Ιακωβιανό μητρώο υπολογίζεται με τη χρήση των συναρτήσεων σχήματος  $R_i(\xi)$  και των Control Points με Καρτεσιανές συντεταγμένες  $X_i$ .

$$[J(\xi)] = \begin{bmatrix} R_{1,\xi}(\xi) & R_{2,\xi}(\xi) & \dots & \dots & R_{n,\xi}(\xi) \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_n \end{bmatrix}, \quad (4.6)$$

$\begin{matrix} (1 \times 1) & & & (1 \times n) & & (n \times 1) \end{matrix}$

όπου  $R_{i,\xi}(\xi) = \frac{\partial}{\partial \xi} R_i(\xi)$ .

Υπολογίζεται επίσης το αντίστροφο του Ιακωβιανού μητρώου  $[J]^{-1}$ .

$$[J]^{-1} = \frac{1}{[J]}, \quad (4.7)$$

$\begin{matrix} (1 \times 1) & & (1 \times 1) \end{matrix}$

Στην πορεία για τον υπολογισμό του μητρώου παραμορφώσεως  $[B]$ , υπολογίζονται τα επιμέρους μητρώα  $[B_1]$  και  $[B_2]$ . Το μητρώο  $[B_1]$  μεταφέρει τις παραμορφώσεις του στοιχείου στον Parameter Space με τον Physical Space ενώ τον μητρώο  $[B_2]$  συνδέει τις επικόμβιες μετακινήσεις του στοιχείου με τις παραμορφώσεις του στον Parameter Space. Επομένως, τα μητρώα  $[B_1]$  και  $[B_2]$  υπολογίζονται ως εξής

$$\begin{aligned} \left\{ \varepsilon \right\}_{(1 \times 1)} &= \left[ \frac{\partial u}{\partial x} \right] = [J]^{-1} \cdot \left[ \frac{\partial u}{\partial \xi} \right] \\ [B_1(\xi)]_{(1 \times 1)} &= \left[ \frac{1}{J_{11}} \right], \\ \left\{ \varepsilon \right\}_{(1 \times 1)} &= [B_1]_{(1 \times 1)} \cdot \left\{ u_\xi \right\}_{(1 \times 1)} \end{aligned} \quad (4.8)$$

$$\left[ \frac{\partial \mathbf{u}}{\partial \xi} \right] = \begin{bmatrix} R_{1,\xi}(\xi) & R_{2,\xi}(\xi) & \dots & \dots & R_{n,\xi}(\xi) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_n \end{bmatrix}$$

$$[\mathbf{B}_2(\xi)] = \begin{bmatrix} R_{1,\xi}(\xi) & R_{2,\xi}(\xi) & \dots & \dots & R_{n,\xi}(\xi) \end{bmatrix}, \quad (4.9)$$

$$\{u\} = [\mathbf{B}_2] \{d\}$$

$\begin{matrix} (1 \times 1) & & (1 \times n) & & (n \times 1) \end{matrix}$

Μπορεί να υπολογιστεί πλέον το μητρώο παραμορφώσεως  $[\mathbf{B}]$ , το οποίο συνδέει τις παραμορφώσεις του φορέα με τις επικόμβιες μετακινήσεις του.

$$[\mathbf{B}(\xi)] = [\mathbf{B}_1(\xi)] [\mathbf{B}_2(\xi)], \quad (4.10)$$

$\begin{matrix} (1 \times n) & & (1 \times 1) & & (1 \times n) \end{matrix}$

$$\{\varepsilon(\xi)\} = [\mathbf{B}(\xi)] \{d\}$$

$\begin{matrix} (1 \times 1) & & (1 \times n) & & (n \times 1) \end{matrix}$

Για τον υπολογισμό του καθολικού μητρώου στιβαρότητας χρησιμοποιείται η εξής σχέση

$$[\mathbf{K}] = \int_{\xi_0}^{\xi_{n+p+1}} \begin{bmatrix} \mathbf{B}(\xi) \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{B}(\xi) \end{bmatrix} A \cdot \det[\mathbf{J}] d\xi, \quad (4.11)$$

$\begin{matrix} (n \times n) & & \xi_0 & & (n \times 1) & & (1 \times 1) & & (1 \times n) \end{matrix}$

Ωστόσο, δεν είναι δυνατός ο άμεσος υπολογισμός του ολοκληρώματος γι' αυτό ακολουθεί αριθμητική ολοκλήρωση, όπου υπολογίζεται το άθροισμα του παραπάνω γινομένου για κάθε Gauss Point πολλαπλασιασμένο με το αντίστοιχο βάρος του.

$$[\mathbf{K}] = \sum_{i=1}^{GP\xi} \left\{ \begin{bmatrix} \mathbf{B}(\xi_i) \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{B}(\xi_i) \end{bmatrix} A \cdot \det[\mathbf{J}] w_i^{GP\xi} \right\}, \quad (4.12)$$

$\begin{matrix} (n \times n) & & \sum_{i=1}^{GP\xi} & & \left\{ \begin{matrix} (n \times 1) & & (1 \times 1) & & (1 \times n) \end{matrix} \right\} & & & & & & \end{matrix}$

όπου

- $A$  : το εμβαδόν της διατομής
- $GP\xi$  : τα συνολικά Gauss Points που ανήκουν σε ένα διάστημα
- $\xi_i$  : οι συντεταγμένες των Gauss Points
- $w_i^{GP\xi}$  : τα αντίστοιχα βάρη

#### 4.2.2 Μητρώο Στιβαρότητας 2D

Τα προβλήματα διδιάστατης ελαστικότητας έχουν ευρεία εφαρμογή καθώς όταν ισχύουν οι συνθήκες για διδιάστατη ανάλυση απλοποιείται αρκετά το πρόβλημα. Η λογική που ακολουθείται είναι ανάλογη με αυτή των 1D προβλημάτων, με μόνη διαφορά την προσθήκη μίας επιπλέον διάστασης. Έτσι, στον Physical Space ορίζονται δύο άξονες αναφοράς, οι (x,y) και αντίστοιχα στον Parameter Space οι (ξ,η).

Το μητρώο παραμόρφωσης ορίζεται ως εξής

$$\begin{Bmatrix} \varepsilon \\ \gamma \end{Bmatrix}_{(3 \times 1)} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} \Rightarrow \begin{Bmatrix} \varepsilon \\ \gamma \end{Bmatrix}_{(3 \times 1)} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix}, \quad (4.13)$$

Οι μετακινήσεις στους άξονες (x,y) ορίζονται ως  $u(x,y) = u(S(\xi,\eta)) = u(\xi,\eta)$  και αντίστοιχα  $v(x,y) = v(\xi,\eta)$ .

Ο μετασχηματισμός μίας συνάρτησης  $\phi$  από τον Physical Space στον Parameter Space και αντιστρόφως γίνεται μέσω του Ιακωβιανού μητρώου [J].

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \phi}{\partial \eta} \frac{\partial \eta}{\partial x}$$

$$\frac{\partial \phi}{\partial y} = \frac{\partial \phi}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \phi}{\partial \eta} \frac{\partial \eta}{\partial y}$$

$$\frac{\partial \phi}{\partial \xi} = \frac{\partial \phi}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial \phi}{\partial y} \frac{\partial y}{\partial \xi}$$

$$\frac{\partial \phi}{\partial \eta} = \frac{\partial \phi}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial \phi}{\partial y} \frac{\partial y}{\partial \eta}$$

Επομένως το 2D Ιακωβιανό Μητρώο ορίζεται ως

$$\begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \end{bmatrix} = \underset{(2 \times 2)}{[J]} \cdot \begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \end{bmatrix}, \quad (4.14)$$

και αντιστρέφεται



$$\begin{bmatrix} \frac{\partial \varphi}{\partial x} \\ \frac{\partial \varphi}{\partial y} \end{bmatrix} = \underset{(2 \times 2)}{[J]}^{-1} \cdot \begin{bmatrix} \frac{\partial \varphi}{\partial \xi} \\ \frac{\partial \varphi}{\partial \eta} \end{bmatrix}, \quad (4.15)$$

Το Ιακωβιανό μητρώο υπολογίζεται με τη χρήση των συναρτήσεων σχήματος  $R_i(\xi, \eta)$  και των Control Points με Καρτεσιανές συντεταγμένες  $X_i, Y_i$ .

$$\underset{(2 \times 2)}{[J]} = \begin{bmatrix} R_{1,\xi}(\xi, \eta) & R_{2,\xi}(\xi, \eta) & \dots & \dots & R_{N,\xi}(\xi, \eta) \\ R_{1,\eta}(\xi, \eta) & R_{2,\eta}(\xi, \eta) & \dots & \dots & R_{N,\eta}(\xi, \eta) \end{bmatrix} \cdot \begin{bmatrix} X_1 & Y_1 \\ X_2 & Y_2 \\ \vdots & \vdots \\ \vdots & \vdots \\ X_N & Y_N \end{bmatrix}, \quad (4.16)$$

(2xN) (N x 2)

όπου  $R_{i,\xi}(\xi, \eta) = \frac{\partial}{\partial \xi} R_i(\xi, \eta)$ ,  $R_{j,\eta}(\xi, \eta) = \frac{\partial}{\partial \eta} R_j(\xi, \eta)$  και  $N = n \cdot m$ , ο συνολικός αριθμός των Control Points.

Υπολογίζεται επίσης το αντίστροφο του Ιακωβιανού μητρώου  $[J]^{-1}$ .

$$\underset{(2 \times 2)}{[J]}^{-1} = \begin{bmatrix} J_{11}^* & J_{12}^* \\ J_{21}^* & J_{22}^* \end{bmatrix} = \frac{1}{\det[J]} \cdot \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix}, \quad (4.17)$$

όπου η ορίζουσα του Ιακωβιανού μητρώου ισούται με

$$\det[J] = J_{11} \cdot J_{22} - J_{21} \cdot J_{12}, \quad (4.18)$$

Στην πορεία για τον υπολογισμό του μητρώου παραμορφώσεως  $[B]$ , υπολογίζονται τα επιμέρους μητρώα  $[B_1]$  και  $[B_2]$ .

$$\underset{(3 \times 1)}{\{\varepsilon\}} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} = \frac{1}{\det[J]} \cdot \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{bmatrix}, \quad (4.19)$$

Άρα,

$$[B_1(\xi, \eta)]_{(3 \times 4)} = \frac{1}{\det[J]} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix}, \quad (4.20)$$

Για τον υπολογισμό του μητρώου  $[B_2]$

$$\begin{bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{bmatrix} = \begin{bmatrix} R_{1,\xi} & 0 & R_{2,\xi} & 0 & \dots & \dots & \dots & R_{N,\xi} & 0 \\ R_{1,\eta} & 0 & R_{2,\eta} & 0 & \dots & \dots & \dots & R_{N,\eta} & 0 \\ 0 & R_{1,\xi} & 0 & R_{2,\xi} & \dots & \dots & \dots & 0 & R_{N,\xi} \\ 0 & R_{1,\eta} & 0 & R_{2,\eta} & \dots & \dots & \dots & 0 & R_{N,\eta} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ \vdots \\ u_N \\ v_N \end{bmatrix}$$

$$[B_2(\xi, \eta)]_{(4 \times 2N)} = \begin{bmatrix} R_{1,\xi} & 0 & R_{2,\xi} & 0 & \dots & \dots & \dots & R_{N,\xi} & 0 \\ R_{1,\eta} & 0 & R_{2,\eta} & 0 & \dots & \dots & \dots & R_{N,\eta} & 0 \\ 0 & R_{1,\xi} & 0 & R_{2,\xi} & \dots & \dots & \dots & 0 & R_{N,\xi} \\ 0 & R_{1,\eta} & 0 & R_{2,\eta} & \dots & \dots & \dots & 0 & R_{N,\eta} \end{bmatrix}, \quad (4.21)$$

Μπορεί να υπολογιστεί πλέον το μητρώο παραμορφώσεως  $[B]$ , το οποίο συνδέει τις παραμορφώσεις του φορέα με τις επικόμβιες μετακινήσεις του.

$$[B(\xi, \eta)]_{(3 \times 2N)} = [B_1(\xi, \eta)]_{(3 \times 4)} [B_2(\xi, \eta)]_{(4 \times 2N)}, \quad (4.22)$$

Για τον υπολογισμό του ολοκληρώματος του μητρώου στιβαρότητας, γίνεται αριθμητική ολοκλήρωση για όλους τους πιθανούς συνδυασμούς Gauss Points των αξόνων  $(\xi, \eta)$

$$[\mathbf{K}]_{(2N \times 2N)} = \int_{\xi_0}^{\xi_{n+p+1}} \int_{\eta_0}^{\eta_{m+p+1}} \underbrace{[\mathbf{B}(\xi, \eta)]^T}_{(2N \times 3)} \cdot \underbrace{[\mathbf{E}]}_{(3 \times 3)} \cdot \underbrace{[\mathbf{B}(\xi, \eta)]}_{(3 \times 2N)} t \cdot \det[\mathbf{J}] d\xi d\eta, \quad (4.23)$$

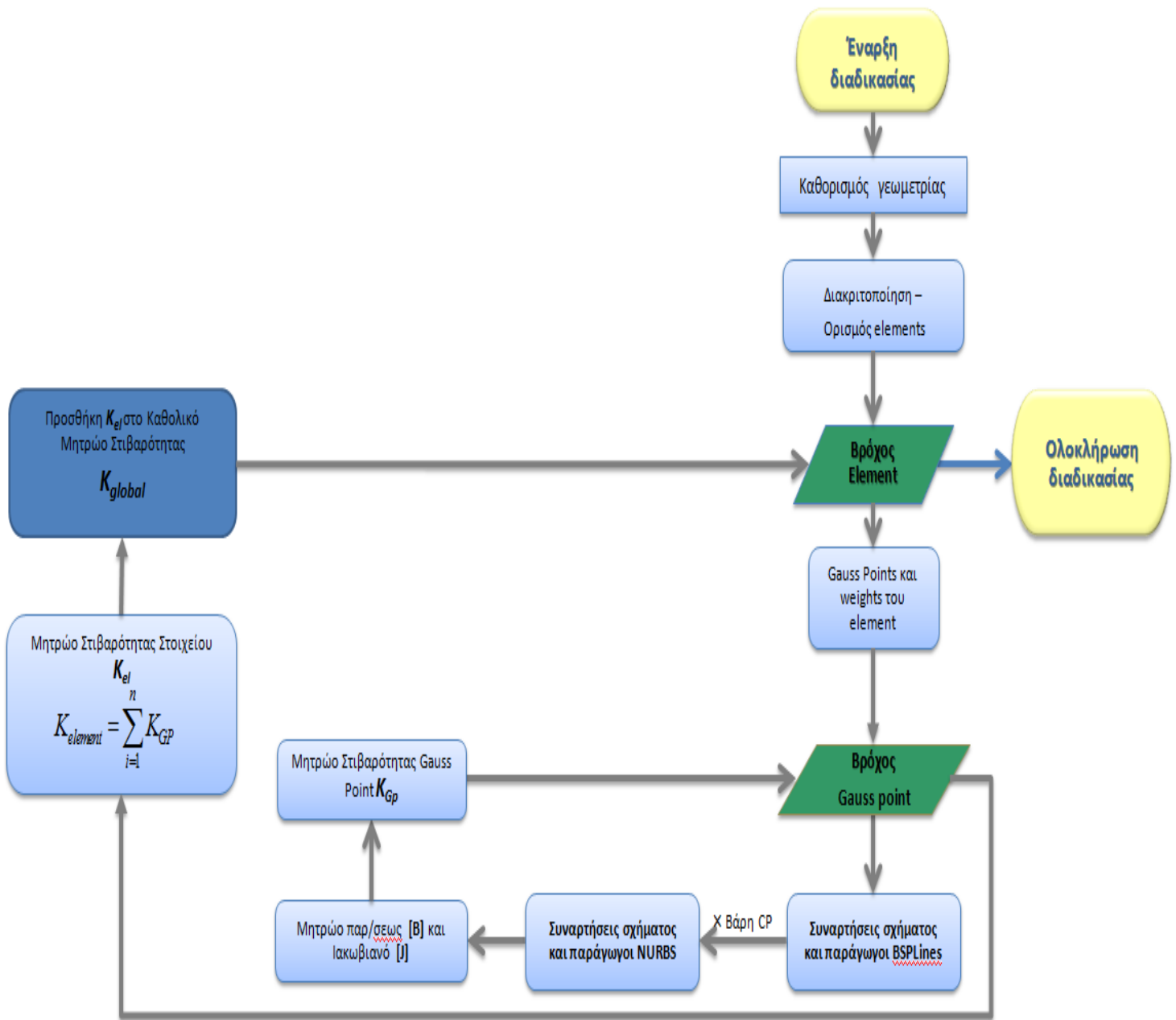
Ωστόσο, δεν είναι δυνατός ο άμεσος υπολογισμός του ολοκληρώματος γι' αυτό ακολουθεί αριθμητική ολοκλήρωση, όπου υπολογίζεται το άθροισμα του παραπάνω γινομένου για κάθε Gauss Point πολλαπλασιασμένο με το αντίστοιχο βάρος του.

$$[\mathbf{K}]_{(2N \times 2N)} = \sum_{i=1}^{GP_\xi} \sum_{j=1}^{GP_\eta} \left\{ \underbrace{[\mathbf{B}(\xi_i, \eta_j)]^T}_{(2N \times 3)} \cdot \underbrace{[\mathbf{E}]}_{(3 \times 3)} \cdot \underbrace{[\mathbf{B}(\xi_i, \eta_j)]}_{(3 \times 2N)} t \cdot \det[\mathbf{J}] w_i^{GP_\xi} w_j^{GP_\eta} \right\}, \quad (4.24)$$

όπου

- $t$  : το πάχος της διατομής
- $GP_\xi$  : τα συνολικά Gauss Points που ανήκουν σε ένα διάστημα, στον άξονα  $\xi$
- $GP_\eta$  : τα συνολικά Gauss Points που ανήκουν σε ένα διάστημα, στον άξονα  $\eta$
- $\xi_i, \eta_j$  : ο συνδυασμός των συντεταγμένων των Gauss Points  $i, j$
- $w_i^{GP_\xi}, w_j^{GP_\eta}$  : τα αντίστοιχα βάρη

# Τοπολογική Βελτιστοποίηση με Ισογεωμετρικά Πεπερασμένα Στοιχεία



Σχήμα 4.3 Διάγραμμα ροής για τον υπολογισμό του Μητρώου Στιβαρότητας IGA

## 5 Τοπολογική Βελτιστοποίηση –Topology Optimization

### 5.1 Εισαγωγή

Με τον όρο βελτιστοποίηση περιγράφεται το σύνολο όλων των ενεργειών με στόχο την τροποποίηση ενός συστήματος ή ενός σχεδίου, ώστε να υλοποιείται με τον πλέον αποδοτικό τρόπο, πληρώντας συγχρόνως κάποια κριτήρια και ικανοποιώντας τους απαραίτητους περιορισμούς. Η εξέλιξη της έννοιας της βελτιστοποίησης πραγματοποιείται παράλληλα με την εξέλιξη της επιστήμης. Συγκεκριμένα, η ανάπτυξη τεχνολογίας των υπολογιστών με υψηλές ταχύτητες και η βελτίωση των αλγορίθμων που χρησιμοποιούνται στο σχεδιασμό βελτιστοποίησης, δίνουν τη δυνατότητα υλοποίησης των μεθόδων βελτιστοποίησης αλλά και της περαιτέρω έρευνας για την δημιουργία νέων μεθόδων και επέκταση της εφαρμογής τους και σε άλλους επιστημονικούς κλάδους [8].

### 5.2 Βέλτιστος Σχεδιασμός Κατασκευών

Η εφαρμογή της βελτιστοποίησης στον τομέα της μηχανικής των κατασκευών, στοχεύει στη βελτίωση της μορφής μίας κατασκευής υπό την ικανοποίηση περιορισμών και απαιτήσεων, όπως μείωση βάρους για ελαχιστοποίηση κόστους κατασκευής, μείωση περιβαλλοντικών επιπτώσεων, εξασφαλίζοντας παράλληλα την υψηλή απόδοση και ασφαλή λειτουργία της. Για να σχεδιαστεί το δομικό σύστημα λοιπόν, με το βέλτιστο δυνατό τρόπο πραγματοποιείται μία επαναληπτική διαδικασία εύρεσης βέλτιστου σχεδιασμού.

Η διαδικασία περιλαμβάνει δύο στάδια, αυτό της μαθηματικής διατύπωσης του προβλήματος βελτιστοποίησης και αυτό του αλγόριθμου βελτιστοποίησης. Στο στάδιο της μαθηματικής διατύπωσης, ορίζεται ένας συνδυασμός ανεξάρτητων μεταβλητών, οι οποίες ονομάζονται μεταβλητές σχεδιασμού (*design variables*) και λαμβάνουν πραγματικές ή ακέραιες τιμές, η μεταξύ τους σχέση, οι περιορισμοί του προβλήματος αλλά προσδιορίζεται και η προς βελτιστοποίηση συνάρτηση, η οποία ονομάζεται αντικειμενική συνάρτηση (*objective function*). Εν συνεχεία, επιλέγεται ο κατάλληλος αλγόριθμος βελτιστοποίησης για την επίλυση του προβλήματος που έχει οριστεί.

#### 5.2.1 Κατηγορίες Αλγορίθμων Βελτιστοποίησης

Ένας απλός τρόπος εφαρμογής της διαδικασίας, είναι η μέθοδος της δοκιμής, ελέγχου και διόρθωσης διαφόρων υποψήφιων βέλτιστων σχεδιασμών. Ωστόσο, αυτή η μέθοδος, που πρόκειται για μία εμπειρική μέθοδο, δεν είναι δυνατόν να εφαρμοστεί σε προβλήματα αυξημένης πολυπλοκότητας και απαιτήσεων. Αναγκαία λύση κατέστη η αυτοματοποίηση του σχεδιασμού, αξιοποιώντας παράλληλα την εξέλιξη της τεχνολογίας των υπολογιστών, και της ανάπτυξης αλγορίθμων βελτιστοποίησης, ώστε οι δοκιμές της επαναληπτικής διαδικασίας να πραγματοποιούνται αυτόματα και έτσι να προκύπτει η βέλτιστη λύση σε λιγότερο χρόνο αλλά και με μεγαλύτερη ακρίβεια.

Οι πρώτοι αλγόριθμοι που χρησιμοποιήθηκαν στο σχεδιασμό κατασκευών, ήταν από τον κλάδο των μαθηματικών, των οικονομικών και της επιχειρησιακής έρευνας και βασίζονταν στον μαθηματικό προγραμματισμό (*mathematical programming*). Υπάρχουν

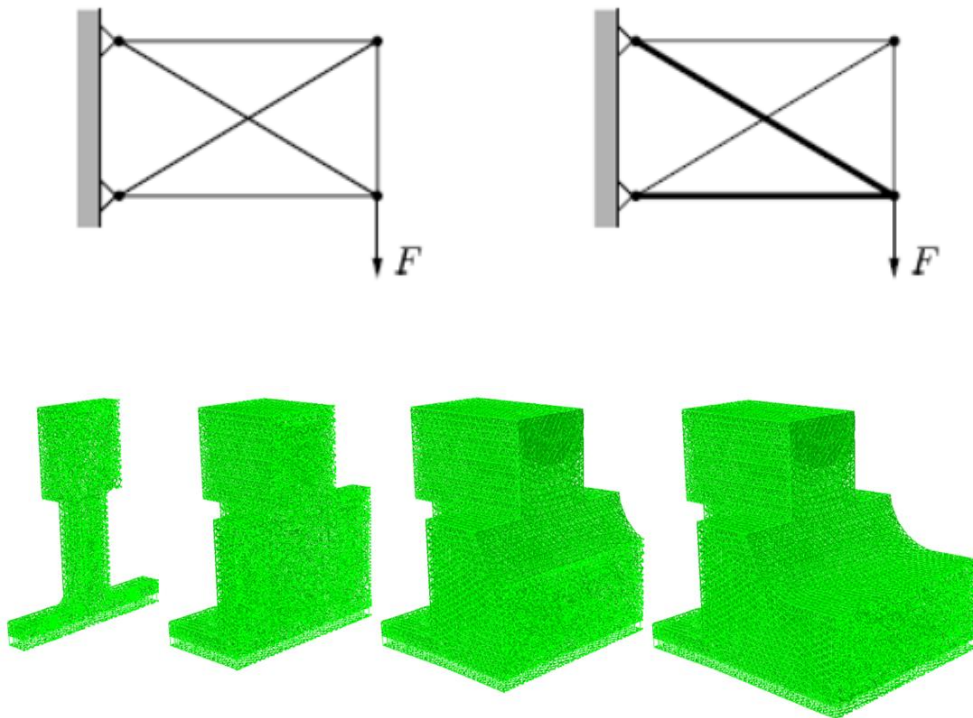
επίσης οι τυχηματικές μέθοδοι βελτιστοποίησης, οι οποίες πλεονεκτούν έναντι των μαθηματικών μεθόδων καθώς έχουν μεγαλύτερο φάσμα εφαρμογής σε προβλήματα .

### 5.2.2 Κατηγορίες Βελτιστοποίησης Κατασκευών

Έχει αναπτυχθεί μεγάλος αριθμός προβλημάτων βελτιστοποίησης που ποικίλουν βάσει του τύπου, του στόχου ή και του σκοπού του προβλήματος. Τρία βασικά είδη προβλημάτων βελτιστοποίησης είναι τα εξής

#### 5.2.2.1 Βελτιστοποίηση Διαστασιολόγησης Διατομής ( Sizing Optimization )

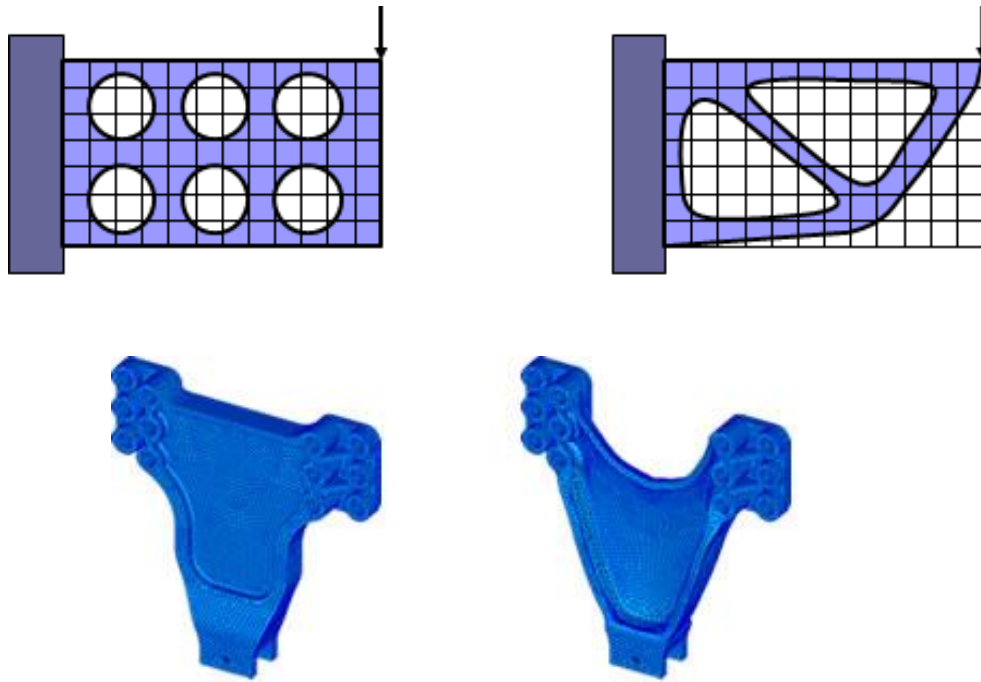
Τα προβλήματα βελτιστοποίησης διαστασιολόγησης διατομής, έχουν μελετηθεί αρκετά στο παρελθόν και έχουν ως μεταβλητές σχεδιασμού τις διαστάσεις και τα πάχη των διατομών, αλλά και τις ιδιότητες των υλικών. Ως αντικειμενική συνάρτηση, τίθενται το βάρος της κατασκευής και ως περιορισμοί οι τάσεις και η ενέργεια παραμόρφωσης.



Σχήμα 3.1 Παραδείγματα Βελτιστοποίησης Διαστασιολόγησης Διατομής

#### 5.2.2.2 Βελτιστοποίηση Σχήματος Κατασκευής ( Shape Optimization )

Κατά τη βελτιστοποίηση σχήματος, αναζητείται το βέλτιστο σχήμα διατομών ράβδων ή ολόσωμων κατασκευών, αφού έχει καθοριστεί ήδη ο αριθμός των δομικών μελών, με στόχο την πιο οικονομική και λειτουργική χρήση τους, πληρώνοντας πάντα τους περιορισμούς που έχουν τεθεί.

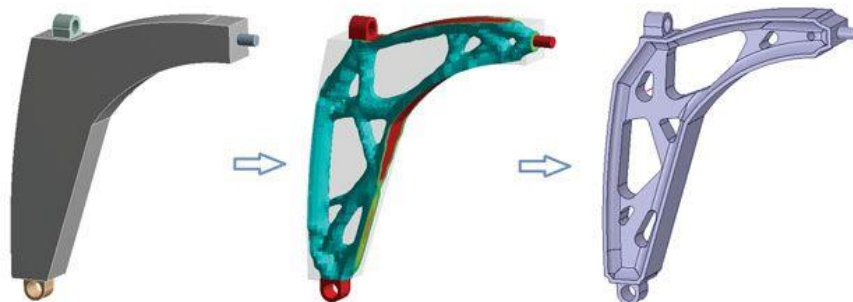


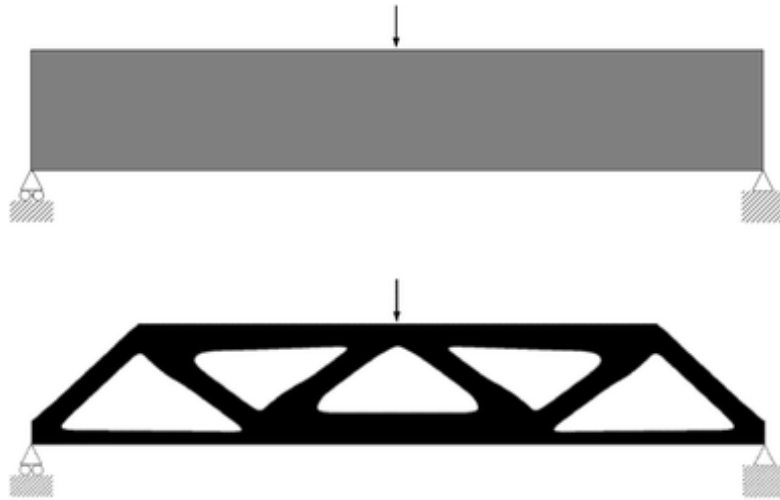
Σχήμα 3.2 Παραδείγματα Βελτιστοποίησης Σχήματος Κατασκευής

### 5.2.2.3 Βελτιστοποίηση Τοπολογίας Κατασκευής ( Topology Optimization )

Ως βελτιστοποίηση τοπολογίας σε προβλήματα κατασκευών, ορίζεται μία μαθηματική προσέγγιση με στόχο την εύρεση του βέλτιστου τρόπου κατανομής υλικού σε μία κατασκευή, για δεδομένη φόρτιση και συνοριακές συνθήκες στήριξης, ώστε να ελαχιστοποιείται ένα κριτήριο και συγχρόνως να ικανοποιούνται οι απαιτήσεις σχεδιασμού. Ο βέλτιστος σχεδιασμός, είναι η δομή για την οποία η κατασκευή αποκτά τη μέγιστη αντοχή και ακαμψία της, με το ελάχιστο δυνατό υλικό.

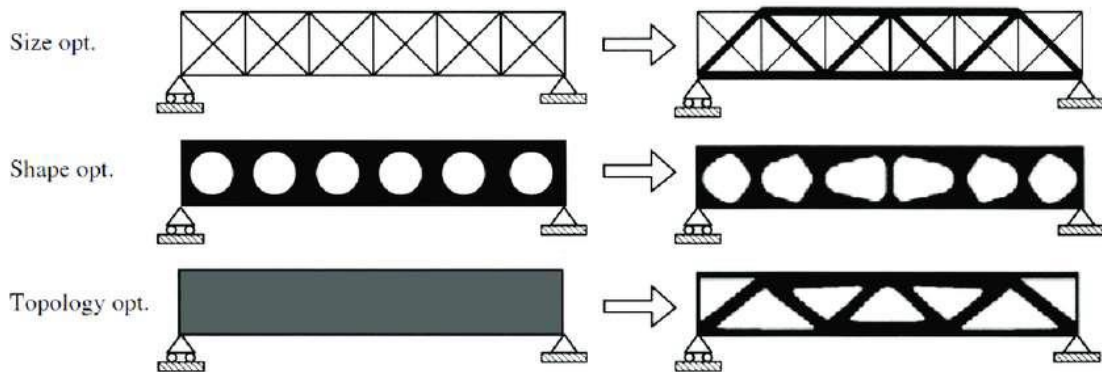
Η κατάλληλη κατανομή του υλικού επιτυγχάνεται με τη χρήση των αλγόριθμων βελτιστοποίησης. Κατά την εκτέλεση του αλγορίθμου, ορίζεται αρχικά ο χώρος σχεδιασμού ή αναφοράς (domain reference), ο τύπος υλικού, οι συνθήκες στήριξης και οι φορτίσεις. Στη συνέχεια, πραγματοποιείται η ανάλυση του φορέα ώστε να ξεκινήσει η επαναληπτική διαδικασία εύρεσης της βέλτιστης τοπολογίας της κατασκευής. Η διαδικασία ολοκληρώνεται όταν υπάρξει σύγκλιση λύσης, που ικανοποιεί όλους τους περιορισμούς.





Σχήμα 3.3 Παραδείγματα Βελτιστοποίησης Τοπολογίας Κατασκευής

Οι παραπάνω τύποι προβλήματος αναφέρθηκαν κατά σειρά αύξουσας πολυπλοκότητας και φάσματος πεδίου εφαρμογών. Συγκριτικά τα αποτελέσματα εφαρμογή τους σε αμφιέρειστο φορέα παρουσιάζονται παρακάτω. Ωστόσο, είναι δυνατό να υπάρξει συνεργασία μεταξύ των μεθόδων. Συγκεκριμένα, η βελτιστοποίηση σχήματος μπορεί να έπεται της βελτιστοποίησης τοπολογίας, ώστε το ως προς βελτιστοποίηση πρόβλημα να έχει δεδομένη τοπολογία κατασκευής, καταλήγοντας επομένως σε αποτελέσματα μεγαλύτερης ακρίβειας.



Σχήμα 3.4 Συγκριτική απεικόνιση της βελτιστοποίησης διαστασιολόγησης, σχήματος και τοπολογίας [8]

### 5.3 Μαθηματικό Μοντέλο Βελτιστοποίησης

Για την εφαρμογή βελτιστοποίησης τοπολογίας, απαιτείται να σχηματιστεί πρώτα το κατάλληλο πρόβλημα βελτιστοποίησης. Το πρόβλημα, μπορεί να είναι είτε συνεχές είτε διακριτό, ανάλογα με το είδος του πεδίου τιμών των παραμέτρων σχεδιασμού.

Γενικά, ένα πρόβλημα βελτιστοποίησης διατυπώνεται με το παρακάτω μαθηματικό μοντέλο



Minimize  $f(x)$

Subject to:  $g_i(x) \leq 0, \quad i = 1, m$   
 $h_j(x) = 0, \quad j = 1, p$   
 $x_k^l \leq x_k \leq x_k^u$

όπου  $f(x)$  είναι η αντικειμενική συνάρτηση (objective function), της οποίας αναζητείται η ελάχιστη τιμή, ενώ πληρούνται οι συναρτήσεις περιορισμού (constraints) ανισοτήτων και ισοτήτων,  $g_i(x)$  και  $h_j(x)$  αντίστοιχα. Το  $x$  είναι το διάνυσμα που περιέχει τις μεταβλητές σχεδιασμού, πλήθους  $n$ . Επιπλέον, η μεταβλητή  $x_k$  έχει άνω και κάτω όριο.

### 5.3.1 Αντικειμενική Συνάρτηση

Στόχος της βελτιστοποίησης είναι, αφού εντοπιστούν διάφοροι πιθανοί σχεδιασμοί, να επιλεγεί ο βέλτιστος σχεδιασμός για την κατασκευή. Είναι απαραίτητη λοιπόν, η διατύπωση κάποιου κριτηρίου, του οποίου η ικανοποίηση οδηγεί στην τελική επιλογή του σχεδιασμού. Το κριτήριο αυτό, εκφράζεται μέσω της αντικειμενικής συνάρτησης  $f(x)$ , η οποία λαμβάνει κάποια τιμή για κάθε σχεδιασμό. Συνήθη παραδείγματα αντικειμενικών συναρτήσεων είναι η ελαχιστοποίηση του βάρους, του κόστους, της απώλειας ενέργειας ή και η μεγιστοποίηση του κέρδους και της απόδοσης του συστήματος.

### 5.3.2 Μεταβλητές Σχεδιασμού

Για την κατάλληλη διατύπωση του μαθηματικού μοντέλου ώστε ο σχεδιασμός να είναι πράγματι ο βέλτιστος αλλά να είναι και πραγματοποιήσιμος, είναι απαραίτητη η σωστή επιλογή των μεταβλητών σχεδιασμού  $x$ . Οι μεταβλητές σχεδιασμού είναι οι παράμετροι του προβλήματος και παίρνουν συνεχείς ή διακριτές τιμές. Συνήθως ορίζονται ως μεταβλητές σχεδιασμού γεωμετρικά χαρακτηριστικά της κατασκευής. Πρέπει επίσης να ελέγχεται η σχέση μεταξύ των μεταβλητών, καθώς όταν μια μεταβλητή είναι εξαρτώμενη από μία άλλη, μετατρέπεται σε παράμετρο και οι τιμές της εξαρτώνται από την μεταβλητή σχεδιασμού.

Τέλος, είναι απαραίτητη η πραγματοποίηση μίας ανάλυσης ευαισθησίας. Κατά την ανάλυση αυτή, ελέγχεται το μέγεθος ευαισθησίας της αντικειμενικής συνάρτησης σε σχέση με τις παραμέτρους σχεδιασμού, διαπιστώνοντας έτσι την επιρροή κάθε παραμέτρου στην συνάρτηση.

### 5.3.3 Συναρτήσεις Περιορισμού

Κάθε κατασκευή, εκτός από το κριτήριο που καλείται να ικανοποιεί, πρέπει να πληροί κάποιες απαιτήσεις οι οποίες αποτελούν τους περιορισμούς του μαθηματικού μοντέλου. Για να ληφθούν υπόψη στο πρόβλημα, εισάγονται με την μορφή ισοτήτων  $h(x) = 0$  και ανισοτήτων  $g(x) \leq 0$  ή  $g(x) \geq 0$ . Κάποια παραδείγματα περιορισμών σε προβλήματα κατασκευών είναι ο περιορισμός τάσεων ή μετατοπίσεων σε κάποια όρια τιμών.

## 5.4 Επίλυση μαθηματικού μοντέλου με τη μέθοδο Optimality Criteria (OC)

Αφού διατυπωθεί η αντικειμενική συνάρτηση και προσδιοριστούν οι απαραίτητοι περιορισμοί για την τοπολογική βελτιστοποίηση του μοντέλου, ακολουθεί η επίλυσή του, η οποία υλοποιείται μέσω μίας επαναληπτικής διαδικασίας. Δύο από τις μεθόδους επίλυσης είναι η μέθοδος Optimality Criteria και η μέθοδος Moving Asymptotes (MMA). Για τους σκοπούς της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε η μέθοδος Optimality Criteria.

Βασική ιδέα της μεθόδου OC είναι να ελαχιστοποιηθεί ένα μέρος της αντικειμενικής συνάρτησης σύμφωνα με κάποιο κριτήριο. Η διαδικασία που ακολουθείται, συνοπτικά είναι η εξής

**Επιλογή αρχικών πυκνοτήτων  $x_k$  – ομοιόμορφη κατανομή**

**Υπολογισμός παραγώγου αντικειμενικής συνάρτησης ως προς  $x_e$**

$$C(x) = F^T u(x) - \lambda(K(x)u(x) - F), \quad (5.1)$$

$$\frac{\partial C(x)}{\partial x_e} = F^T \frac{\partial u_e(x)}{\partial x_e} - \lambda \frac{\partial K_e(x)}{\partial x_e} u_e(x) - \lambda K_e(x) \frac{\partial u_e(x)}{\partial x_e}, \quad (5.2)$$

**Ορισμός συνάρτησης Lagrange του προβλήματος**

Με την συνάρτηση Lagrange εκφράζεται ο περιορισμός ως προς τον όγκο. Ελαχιστοποίηση της συνάρτησης Lagrange ως προς  $x_e$ , για δεδομένα  $x_e$ , μεγιστοποίηση της  $L(x, \lambda)$  και υπολογισμός των αντίστοιχων  $x_e$ . Έλεγχος και συνέχιση επαναλήψεων μέχρι  $\sum_{e=1}^n (x_e a_e) = V$

**Τερματισμός διαδικασίας** όταν οι πυκνότητες  $x_e$  λύνουν το αρχικό πρόβλημα βελτιστοποίησης τοπολογίας και συγχρόνως ικανοποιείται το κριτήριο που έχει επιλεγεί

Συγκεκριμένα, σε κάθε επανάληψη υπολογίζονται οι νέες τιμές των πυκνοτήτων με τις οποίες θα ξεκινήσει η επόμενη επανάληψη. Η διαδικασία ολοκληρώνεται όταν πραγματοποιηθεί το κριτήριο, κατά το οποίο όταν η μέγιστη διαφορά όλων των στοιχείων είναι μικρότερη από 0.01. Ωστόσο, όταν η πραγματοποίηση αυτού του κριτηρίου είναι χρονοβόρα, τότε το κριτήριο τερματισμού για το οποίο ολοκληρώνεται η διαδικασία, μπορεί να είναι ο αριθμός επαναλήψεων.

## 5.5 Εφαρμογή βελτιστοποίησης τοπολογίας σε έργα Πολιτικού Μηχανικού

Η βελτιστοποίηση κατασκευών εφαρμόστηκε αρχικά σε εφαρμογές στον τομέα της ναυπηγικής και της βιομηχανίας παραγωγής αυτοκινήτων. Στα έργα του πολιτικού μηχανικού, η εφαρμογή της είναι πιο πρόσφατη.

Η αρχική ιδέα ήταν των Guo και Zhang, οι οποίοι προσπάθησαν να χρησιμοποιήσουν ως κριτήριο βελτιστοποίησης, τον έλεγχο των τάσεων των υλικών καθώς διαπίστωσαν πως η υπέρβαση των τάσεων εκτός από αστοχία στοιχείων οδηγεί και σε αβεβαιότητα για τον

χρόνο ζωής της κατασκευής. Στην πορεία, ακολούθησαν δημοσιεύσεις πολλών ερευνών και παρουσιάσεις μοντέλων βελτιστοποίησης. Παρακάτω, παρουσιάζονται ορισμένες εφαρμογές της βελτιστοποίησης τοπολογίας σε έργα Πολιτικού Μηχανικού.

### **Qatar National Convention Centre (QNCC)**

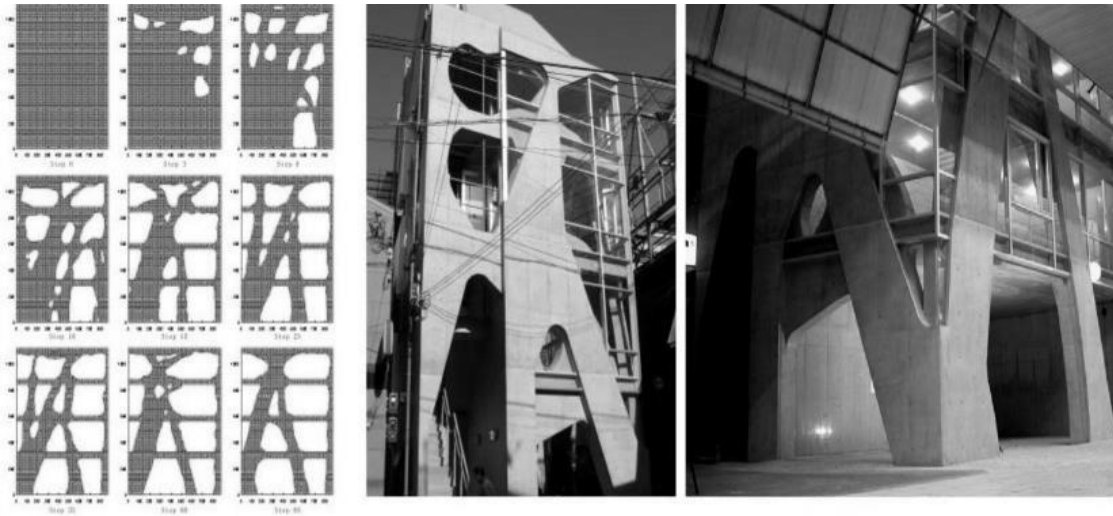
Το Qatar National Convention Centre βρίσκεται στην πόλη Ντόχα, σχεδιάστηκε από τον Ιάπωνα αρχιτέκτονα Arata Isozaki σε συνεργασία με τους αρχιτέκτονες RHWL και λειτουργεί από το 2011. Η στήριξη της οροφής πραγματοποιείται από τεράστια μέλη δενδροειδούς μορφής. Το πρόβλημα της βελτιστοποίησης επιλύθηκε από τον πολιτικό μηχανικό Buro Happold.



**Σχήμα 3.5 Qatar National Convention Centre (QNCC)**

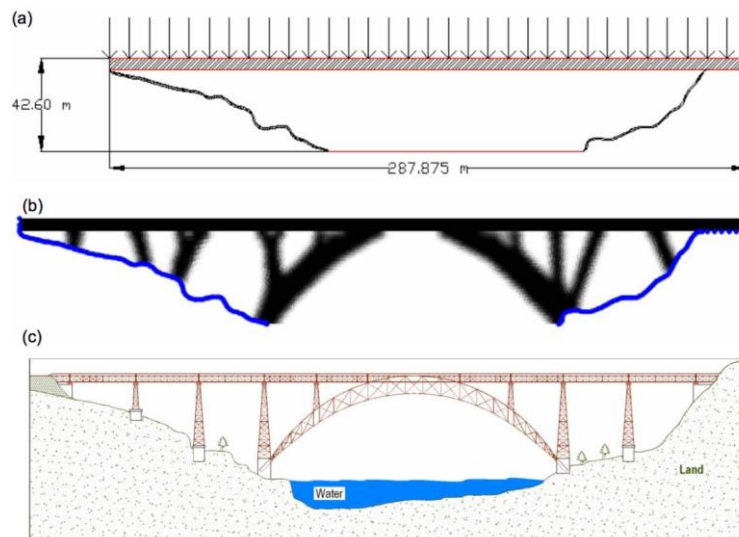
### Τοίχος κτιρίου στην Ιαπωνία

Στην Ιαπωνία χρησιμοποιήθηκε επίσης, για τον σχεδιασμό τοίχων ενός κτιρίου από τους Ohmori et al.



Σχήμα 3.6 Akutagawa River Side in Takatsuki, Ιαπωνία 2004 [9]

Τέλος, ο αρχιτέκτονας Alexandre Gustave Eiffel, σχεδίασε κατασκευές που προέκυπταν μετά από μία βελτιστοποίηση τοπολογίας εν μέρει, χωρίς ωστόσο να ακολουθεί αυστηρά τις μαθηματικές μεθόδους. Παραδείγματα που αναφέρθηκαν σε άρθρο των M. Meenakshi Sundaram και G. K. Ananthasuresh [10], είναι οι προτάσεις του Eiffel για τον πύργο του Eiffel και για την σιδηροδρομική γέφυρα Maria Pia πάνω από τον ποταμό Ντούρο στην Πορτογαλία.



Σχήμα 3.7 (a) Προσομοίωση δύο διαστάσεων σιδηροδρομικής γέφυρας Maria Pia-(b) Βέλτιστη λύση-(c) Σχέδιο Eiffel [10]

## 6 Επίλυση προβλημάτων με τη Μέθοδο SIMP σε κώδικα MATLAB

### 6.1 Εισαγωγή

Η μέθοδος βελτιστοποίησης τοπολογίας έχει άμεση εφαρμογή μέσω της χρήσης υπολογιστικών μέσων. Για το σκοπό αυτό, δημιουργούνται διάφορες αλγοριθμικές διαδικασίες. Για την εκπόνηση της διπλωματικής εργασίας, επιλέχθηκε να συνταχθεί κώδικας σε MATLAB χρησιμοποιώντας ως μέθοδο επίλυσης, τη μέθοδο SIMP. [8]

Η επιλογή της MATLAB έγινε καθώς παρέχει ευκολία χρήσης και καλού συντακτικού αλλά και τη δυνατότητα άμεσου εντοπισμού λαθών. Επιπλέον, δίνει τη δυνατότητα εικονικής αποτύπωσης των αποτελεσμάτων. Στα πλαίσια της παρούσας διπλωματικής εργασίας, για την επίλυση απλών εφαρμογών χρησιμοποιήθηκε ως βάση ο ακαδημαϊκός κώδικας SIMP99, [11], ο οποίος επεκτάθηκε σε κώδικα Βελτιστοποίησης Τοπολογίας Διδιάστατων Ισοπαραμετρικών Στοιχείων.

### 6.2 Μέθοδος SIMP (Solid Isotropic Material with Penalization)

Η βελτιστοποίηση τοπολογίας, βασίζεται στην επιστήμη των μαθηματικών και της μηχανικής, χρησιμοποιώντας ως εργαλεία ηλεκτρονικούς υπολογιστές. Αν και είναι ένας καινούργιος τομέας ενασχόλησης, έχει δώσει λύση σε ένα εύρος προβλημάτων καθώς είναι δυνατόν να εφαρμόζεται σε πολλούς επιστημονικούς κλάδους.

Οι πρώτες εφαρμογές της μεθόδου στηρίχτηκαν στη θεωρία ομογενοποίησης, σύμφωνα με την οποία το συνεχές μέσο περιέχει κάποια μικροδομή και πραγματοποιείται βελτιστοποίηση ως προς την πυκνότητα του υλικού. Ωστόσο, πρόκειται για μία μέθοδο αυξημένης δυσκολίας με αποτέλεσμα να μην εφαρμόζεται ευρέως σε προβλήματα μηχανικού. Έτσι, για λόγους απλοποίησης, αντί για θεώρηση μικροδομής οι ερευνητές θεώρησαν την ύπαρξη κάποιας πυκνότητας στο μέσο.

Συγχρόνως, για να είναι δυνατή η κατασκευή της βέλτιστης λύσης που προέκυπτε, δημιουργήθηκε η απαίτηση για κυρίως σχήματα που λαμβάνουν τιμές πυκνότητας 0 ή 1 και όχι ενδιάμεσες. Δημιουργήθηκε τότε η ιδέα μοντελοποίησης των μηχανικών ιδιοτήτων του υλικού με σκοπό την αποφυγή ενδιάμεσων πυκνοτήτων στο βελτιστοποιημένο μοντέλο. Έτσι προέκυψε ένα αρκετά αποδοτικό μοντέλο, το λεγόμενο ποινικοποιημένο αναλογικής δυσκαμψίας *SIMP*.

Η πρώτη παρουσίαση της μεθόδου έγινε από τους Bendsoe και Kikouchi, οι οποίοι πρότειναν για τον υπολογισμό της πυκνότητας, τη σχέση

$$\rho = x^p, \quad p > 1 \quad (6.1)$$

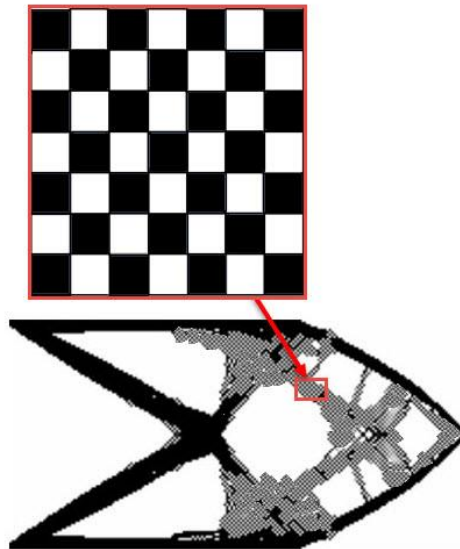
όπου  $x$ , η δυσκαμψία ενός ισότροπου υλικού. Όσα αυξάνεται η δύναμη  $p$ , τόσο επιτυγχάνεται η προσέγγιση της μορφής 0-1. Έχει αποδειχθεί εμπειρικά πως για  $p=3$ , είναι δυνατόν να κατασκευαστούν ισότροπα υλικά με τις ίδιες ελαστικές σταθερές για κάθε τιμή πυκνότητας.

### 6.3 Προβλήματα της μεθόδου

Η SIMP, όπως και κάθε μέθοδος, εμφανίζει κάποια πλεονεκτήματα αλλά και μειονεκτήματα. Δύο από τα βασικά προβλήματα που προέκυψε ήταν αυτό της εμφάνισης μοτίβου τύπου σκακιέρας και η εξάρτηση των αποτελεσμάτων από τη διακριτοποίηση.

Το πρόβλημα μοτίβου τύπου σκακιέρας (Checkerboard problem) πρόκειται για την απότομη εναλλαγή μεταξύ συμπαγή στοιχείων και κενών. Τέτοιου είδους φορείς είναι δύσκολο να κατασκευαστούν έως και αδύνατο.

Το δεύτερο πρόβλημα, αφορά την επίδραση του τρόπου διακριτοποίησης στα αποτελέσματα. Δηλαδή, ανάλογα με το μέγεθος του καννάβου προκύπτουν, διαφορετικά αποτελέσματα. Γι' αυτό, εφαρμόζονται τα κατάλληλα φίλτρα για περιορισμό ή και εξάλειψη τέτοιων προβλημάτων.



### 6.4 Κώδικες Βελτιστοποίησης γραμμένοι σε MATLAB

#### 6.4.1 Κώδικας SIMP Μοντελοποίησης στοιχείων σε δύο διαστάσεις

##### 6.4.1.1 Κώδικας SIMP99 [11]

Ο SIMP99 είναι ένας διδιάστατος κώδικας, αναπτύχθηκε στα πλαίσια ακαδημαϊκής έρευνας από τον Ole Sigmund το 2001 και στόχο είχε την εκπαίδευση μηχανικών. Έχει διατυπωθεί σε 99 γραμμές και χρησιμοποιείται σε απλές εφαρμογές βελτιστοποίησης τοπολογίας.

Στον κώδικα εφαρμόζεται η μέθοδος SIMP και ο φορέας διακριτοποιείται με πεπερασμένα στοιχεία, τα οποία έχουν σταθερές ιδιότητες υλικού. Το πρόβλημα που καλείται να επιλυθεί είναι η μεγιστοποίηση στιβαρότητας για δεδομένο όγκο υλικού. Ως μεταβλητή σχεδιασμού θεωρείται η σχετική πυκνότητα των στοιχείων. Περιλαμβάνει επίσης το απαραίτητο φίλτρο, το οποίο εφαρμόζεται στις παραγώγους των μεγεθών, για αποφυγή του προβλήματος εμφάνισης σκακιέρας.

Ο κώδικας αποτελείται από 99 γραμμές και έχει συνταχθεί σε ένα script. Παρουσιάζονται παρακάτω, αναλυτικά τα βήματα που πραγματοποιούνται κατά την εκτέλεση του κώδικα

- Το πρόγραμμα καλείται από την MATLAB με την εντολή *top(nelx, nely, volfrac, penal, rmin)*

όπου  $n_{elx}$ ,  $n_{ely}$  ο αριθμός στοιχείων στην οριζόντια και στην κατακόρυφη διεύθυνση αντίστοιχα,  $\nu$  το κλάσμα όγκου,  $p$  η δύναμη  $p$  της SIMP για την ποινικοποίηση ενδιάμεσων πυκνοτήτων, και  $r_{min}$  η ακτίνα επιρροής του κάθε στοιχείου. Κατά την εκτέλεση του κώδικα στο τέλος κάθε επανάληψης, προκύπτει η εικόνα της τρέχουσας κατανομής πυκνοτήτων.

- Κυρίως πρόγραμμα (γραμμές 1-36)

Το κυρίως πρόγραμμα (γραμμές 1-36) ξεκινά με την ομοιόμορφη κατανομή υλικού στη περιοχή του σχεδίου (γραμμή 4). Αφού οριστούν τα απαραίτητα δεδομένα εκτελείται η πρώτη επανάληψη του προβλήματος όπου μέσω των εντολών του υποπρογράμματος FE υπολογίζεται το διάνυσμα της μετατόπισης  $U$ . Καθώς το μητρώο στιβαρότητας  $K_0$  είναι ίδιο για όλα τα επιμέρους στοιχεία, αυτό καλείται μόνο μία φορά (γραμμή 14).

Αφού πραγματοποιηθεί μια επανάληψη για όλα τα στοιχεία (γραμμές 16-24) καθορίζεται η αντικειμενική συνάρτηση και οι ευαισθησίες. Η ανάλυση ευαισθησίας ακολουθείται από το κάλεσμα φίλτρου (γραμμή 26) και η μεταβολή των πυκνοτήτων γίνεται βάσει της Optimality Criteria Method, η οποία εκτελείται μέσω της εντολής OC (γραμμή 28). Η τιμή της πυκνότητας κάθε στοιχείου τυπώνεται με τις γραμμές 30-33 και σχεδιάζεται με τη γραμμή 35. Η διαδικασία ολοκληρώνεται όταν μέγιστη διαφορά πυκνοτήτων της τρέχουσας επανάληψης και της αμέσως προηγούμενης είναι μικρότερη από 1%. Διαφορετικά η διαδικασία συνεχίζεται πραγματοποιώντας κι άλλες επαναλήψεις, μέχρι τη σύγκλιση του αποτελέσματος.

- Optimality Criteria Method based optimizer (γραμμές 37-48)

Αφού έχουν οριστεί οι αρχικές πυκνότητες ξεκινά η επαναληπτική διαδικασία όπου θα υπολογιστούν οι νέες πυκνότητες οι οποίες μεταβάλλονται βάσει της Optimality Criteria Method. Υπολογίζονται οι τιμές του πολλαπλασιαστή Lagrange με τις αντίστοιχες πυκνότητες και ελέγχεται αν το άθροισμα των πυκνοτήτων της τρέχουσα επανάληψης είναι ίσο με τον διαθέσιμο όγκο του υλικού ( $\text{sum}(\text{sum}(x_{new}))$ ). Μέσω της διχοτόμησης του αρχικού διαστήματος του πολλαπλασιαστή Lagrange, και ανάλογα με την τιμή του αθροίσματος των πυκνοτήτων, θα προσδιοριστεί αν ο πολλαπλασιαστής τελικά ανήκει στο πρώτο ή το δεύτερο μισό του αρχικού διαστήματος.

- Mesh – Independency filtering (γραμμές 49-64)

Στις γραμμές 49-64 εκτελείται το φίλτρο ευαισθησίας (sensitivity filter). Σύμφωνα με το φίλτρο ελέγχονται μόνο τα στοιχεία που βρίσκονται στην περιοχή επιρροής του εξεταζόμενου στοιχείου, η οποία ορίζεται από την ακτίνα  $r_{min}$ .

Για να λειτουργήσει το φίλτρο πρέπει η τιμή της  $r_{min}$  να είναι μεγαλύτερη από ένα. Στόχος του φίλτρου είναι να εξασφαλιστεί ένα ελάχιστο πάχος των μελών της κατασκευής ενώ θα αυξάνεται η διακριτοποίησή της.

- Finite Element Code (γραμμές 65-99)

Αφού έχουν υπολογιστεί τα τοπικά μητρώα στιβαρότητας κάθε στοιχείου, τοποθετούνται στις κατάλληλες θέσεις του καθολικού μητρώου στιβαρότητας. Ο τρόπος αρίθμησης των κόμβων γίνεται βάσει την ωρολογιακή φορά και είναι κοινός με την περίπτωση του τοπικού μητρώου, όπου ο  $n1$  είναι ο κόμβος στη πάνω αριστερή γωνία του στοιχείου και ο  $n2$  είναι ο κόμβος στη πάνω δεξιά γωνία του στοιχείου.

- Συνθήκες Στήριξης

Στις γραμμές 79 και 80 ορίζονται οι συνοριακές συνθήκες και οι συνθήκες στήριξης. Τροποποιώντας αυτές τις γραμμές είναι δυνατή η μοντελοποίηση και η επίλυση διαφόρων περιπτώσεων προβλημάτων. Οι δεσμευμένοι βαθμοί ελευθερίας (fixeddofs) προσδιορίζονται διαγράφοντας τους από τις γραμμικές εξισώσεις. Ο καθορισμός των μη δεσμευμένων βαθμών ελευθερίας (freedofs) γίνεται μέσω των δεσμευμένων, που έχουν ήδη υπολογιστεί, αυτόματα με τη χρήση της εντολής *setdiff*, που βρίσκει τη διαφορά μεταξύ όλων των βαθμών ελευθερίας και των δεσμευμένων. Το μητρώο στιβαρότητας του στοιχείου υπολογίζεται στις γραμμές 86-99.

- Φορτίσεις

Ορίζονται επίσης, διάφορα είδη φορτίων ή και ακόμα και περιπτώσεις πολλαπλών φορτίσεων, προσθέτοντας κάποιες επιπλέον γραμμές εντολών και τροποποιώντας κάποιες υπάρχουσες. Στη γραμμή 79, με την εντολή  $F(2,1) = -1$  εφαρμόζεται μια κατακόρυφη δύναμη στην πάνω αριστερή γωνία με τιμή ίση με 1.

#### 6.4.1.2 Επέκταση κώδικα SIMP99 για μοντελοποίηση με ισοπαραμετρικά πεπερασμένα στοιχεία

Γενικά, όπως αναφέρθηκε προηγουμένως, η διακριτοποίηση του φορέα με χρήση ισοπαραμετρικών στοιχείων δίνει τη δυνατότητα προσομοίωσης πραγματικών κατασκευών πιο πολύπλοκων σχημάτων απ' ό,τι τα κυβικά στοιχεία, αλλά ακόμα και καμπύλων σχημάτων. Έτσι, επιτυγχάνεται πιο ακριβής προσομοίωση, δίνοντας και πιο αξιόπιστα αποτελέσματα.

Γι' αυτό λοιπόν, μέρος της παρούσας διπλωματικής εργασίας είναι η επέκταση του παραπάνω κώδικα, σε κώδικα επίλυσης με χρήση ισοπαραμετρικών στοιχείων. Για την υλοποίηση αυτή, πραγματοποιήθηκαν οι εξής αλλαγές

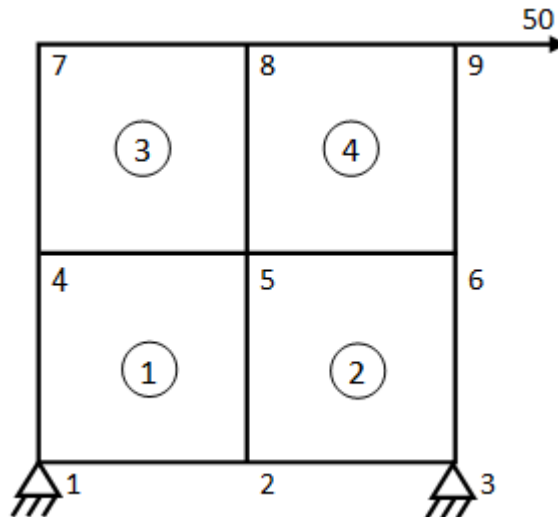
Ο αλγόριθμος αποτελείται από ένα κύριο script το οποίο καλείται από την συνάρτηση *topIsoparametric(nelx, nely, volfrac, penal, rmin)*. Το script περιλαμβάνει όλες τις επιμέρους συναρτήσεις που απαιτούνται για την πραγματοποίηση της ανάλυσης, καθώς και τις αρχικές συνθήκες του προβλήματος και το κριτήριο σύγκλισης.

Η λογική που ακολουθείται είναι ανάλογη αυτής που αναφέρθηκε προηγουμένως με μόνη διαφορά το τρόπο διακριτοποίησης του φορέα. Συγκεκριμένα, εδώ ο φορέας διακριτοποιείται σε επιμέρους στοιχεία ορίζοντας τις συντεταγμένες των κόμβων τους. Κατά την διακριτοποίηση, 'εξαντλούνται' πρώτα τα στοιχεία κατά τον οριζόντιο άξονα και στη συνέχεια κατά τον κατακόρυφο άξονα. Στη συνέχεια, ορίζονται οι κόμβοι του κάθε στοιχείου σύμφωνα με την αριστερόστροφη φορά και υπολογίζονται οι συντεταγμένες του  $[(x_1, x_2, x_3, x_4), (y_1, y_2, y_3, y_4)]$ .

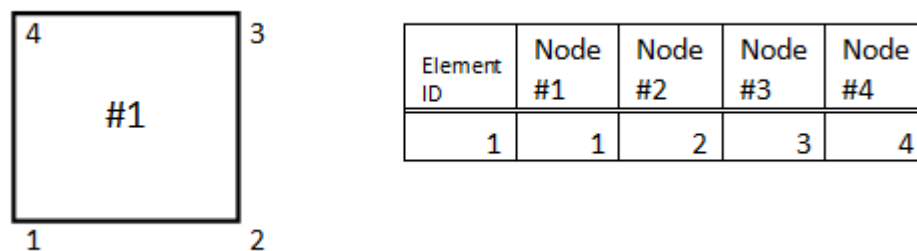
Τέλος, κατά την ανάλυση πεπερασμένων στοιχείων, ο υπολογισμός του μητρώου στιβαρότητας γίνεται χωριστά για το κάθε στοιχείο, εν αντιθέσει με την προηγούμενη περίπτωση που όλα τα στοιχεία έχουν το ίδιο τοπικό μητρώο στιβαρότητας. Ο υπολογισμός των τοπικών μητρώων γίνεται με τη συνάρτηση *function ElementStiffnessMatrixIsoparametric(Coords, thickness, youngsModulus,poissonsRatio)* η οποία καλείται κατά την εκτέλεση της *function Finite\_Analysis(nelx, nely, penal, x)*.

Παραδείγματος χάριν, η αντίστοιχη διακριτοποίηση εντός φορέα και η αρίθμηση των κόμβων των επιμέρους στοιχείων του είναι





Σχήμα 4. 4 Διακριτοποίηση φορέα, εξαντλώντας πρώτα τα στοιχεία στον οριζόντιο άξονα κι έπειτα τον κατακόρυφο



Σχήμα 4. 5 Αρίθμηση κόμβων στοιχείου σύμφωνα με την αριστερόστροφη φορά

## 6.5 Βελτιστοποίηση Τοπολογίας με Ισογεωμετρική Ανάλυση σε κώδικα MATLAB – IGA for Topology Optimization

### 6.5.1 Εισαγωγή

Οι Les Pieg1 και Wayne Tiller, στο βιβλίο “The NURBS Book” [15], αναφέρουν κάποιους αλγόριθμους για τον υπολογισμό των τιμών των βασικών συναρτήσεων και των παραγώγων τους. Οι συναρτήσεις που χρησιμοποιούν είναι B-SPLines. Στα πλαίσια της διπλωματικής εργασίας, χρησιμοποιήθηκαν οι αλγόριθμοι των Les Pieg1 και Wayne Tiller ως βάση για την ανάπτυξη νέων αλγόριθμων υπολογισμού συναρτήσεων NURBS για την πραγματοποίηση Ισογεωμετρικής ανάλυσης. Στη συνέχεια, ενσωματώθηκε ο αλγόριθμος της Ισογεωμετρικής ανάλυσης στον αλγόριθμο Τοπολογικής βελτιστοποίησης. Η λογική ανάπτυξης του αλγόριθμου της Τοπολογικής βελτιστοποίησης με IGA είναι ίδια με την αντίστοιχη της FEM, καθώς η μόνη διαφορά είναι στην μέθοδο διακριτοποίησης και ανάλυσης του φορέα. Σκοπός είναι να συγκριθούν τα αποτελέσματα ενός προβλήματος Τοπολογικής βελτιστοποίησης, που προκύπτουν από τις δύο μεθόδους ανάλυσης.

### 6.5.2 Αλγόριθμος Ισογεωμετρικής ανάλυσης (IGA)

Πρώτο βήμα για την εκτέλεση της Ισογεωμετρικής ανάλυσης είναι η ανάπτυξη ενός αλγορίθμου για τον υπολογισμό των συναρτήσεων σχήματος και των παραγώγων τους σε οποιοδήποτε σημείο. Όπως αναφέρθηκε, στην IGA από κάθε Knot span προκύπτει ένα element. Γι' αυτό πρέπει να υπολογίζεται πρώτα το Knot span που ανήκει το element και στην συνέχεια υπολογίζονται οι συναρτήσεις σχήματος που αντιστοιχούν στο Knot span.

Στη συνέχεια μετατρέπονται οι βασικές συναρτήσεις B-PLines σε συναρτήσεις NURBS. Αφού υπολογιστούν και οι βασικές συναρτήσεις NURBS, ο αλγόριθμος θα υπολογίσει τις παραγώγους των βασικών συναρτήσεων. Οι συναρτήσεις (functions) που χρησιμοποιήθηκαν από τον αλγόριθμο σε αυτό το στάδιο της ανάλυσης είναι

```
function span = FindSpan(numberOfFunctions, degree, parametricCoordinate, KnotValueVector)
```

```
function NURBFunctions = findNURBFunctions(numberOfFunctions, degree, parametricCoordinate, KnotValueVector)
```

```
function NURBSDerivatives = findNURBSDerivatives(degree, parametricCoordinate, KnotValueVector)
```

όπου

- **numberOfFunctions** : ο αριθμός των συναρτήσεων σε κάθε διάστημα, που είναι ίσος με  $(p+1)$
- **degree** : ο πολωνυμικός βαθμός  $p$
- **parametricCoordinate** : η συντεταγμένη ενός σημείου στον παραμετρικό χώρο
- **KnotValueVector** : το διάνυσμα κόμβων του προβλήματος

Οι παραπάνω συναρτήσεις αφορούν προβλήματα 1D και χρησιμοποιήθηκαν για καλύτερη κατανόηση της μεθόδου. Πρακτικά όμως για την επίλυση διαφόρων εφαρμογών στα πλαίσια της διπλωματικής εργασίας χρησιμοποιούνται κώδικες για προβλήματα 2D και 3D. Με τις παρακάτω συναρτήσεις υπολογίζονται πρώτα οι συναρτήσεις B-SPLines στους άξονες  $\xi$  και  $\eta$ , μετατρέπονται στις αντίστοιχες συναρτήσεις NURBS και στη συνέχεια υπολογίζονται οι παράγωγοι των συναρτήσεων NURBS περί τους άξονες  $\xi$  και  $\eta$ .

```
function [dR_dKsi , dR_dHeta] = NURBSDerivs2D (Ksi_degree, Heta_degree, Ksi_parametricCoordinate, Heta_parametricCoordinate, Ksi_KnotVector, Heta_KnotVector)
```

όπου

- **Ksi\_degree** : ο πολωνυμικός βαθμός  $p$  του  $\xi$  άξονα
- **Heta\_degree** : ο πολωνυμικός βαθμός  $q$  του  $\eta$  άξονα
- **Ksi\_parametricCoordinate** : η συντεταγμένη ενός σημείου του  $\xi$  άξονα στον παραμετρικό χώρο
- **Heta\_parametricCoordinate** : η συντεταγμένη ενός σημείου του  $\eta$  άξονα στον παραμετρικό χώρο
- **Ksi\_KnotValueVector** : το διάνυσμα κόμβων του προβλήματος στον  $\xi$  άξονα
- **Heta\_KnotValueVector** : το διάνυσμα κόμβων του προβλήματος στον  $\eta$  άξονα

Επόμενο βήμα για την Ισογεωμετρική ανάλυση είναι η μόρφωση του μητρώου στιβαρότητας κάθε στοιχείου και μετά η σύνθεση τους για τον σχηματισμό του καθολικού μητρώου στιβαρότητας. Για διευκόλυνση της διαδικασίας υπολογισμού των μητρώων στιβαρότητας κάθε στοιχείου, ομαδοποιήθηκαν οι ιδιότητες (*create struct*) κάθε στοιχείου σε μία μεταβλητή, την *elementsData*, έτσι ώστε να πραγματοποιείται ίδια διαδικασία υπολογισμού των μητρώων στιβαρότητας κάθε στοιχείου, δίνοντας κάθε φορά ως δεδομένο το *element* του οποίου θα υπολογιστεί το μητρώο.

Οι πληροφορίες που περιέχονται σε κάθε *struct* είναι το διάστημα *Knot span* το οποίο συνθέτει και το αντίστοιχο *element*, ο βαθμός του πολυωνύμου σε κάθε άξονα, τα *Gauss Points* με τα αντίστοιχα βάρη και τα *Control Points* με τα αντίστοιχα βάρη του κάθε *element*, σε όλους τους άξονες. Οι συναρτήσεις που χρησιμοποιήθηκαν είναι οι εξής

```
function [gaussCoordinates gaussWeights] = UnivariateGaussPoints(degree, span)
```

```
function [Ksi_CP Heta_CP weightfactorCP] = ControlPoints (Ksi_degree, Heta_degree, Ksi_KnotValueVector, Heta_KnotValueVector)
```

```
function elementsData = CreateElements (Ksi_degree, Heta_degree, Ksi_elements, Heta_elements)
```

όπου

- *Ksi\_elements* : το πλήθος των στοιχείων στον άξονα  $\xi$
- *Heta\_elements* : το πλήθος των στοιχείων στον άξονα  $\eta$

Αφού έχουν υπολογιστεί και οι ιδιότητες των *elements*, το επόμενο βήμα για την ανάλυση είναι ο σχηματισμός του μητρώου στιβαρότητας κάθε *element*. Καλείται λοιπόν μία νέα συνάρτηση με την οποία υπολογίζονται ανάλογα με το *element* που δέχεται ως όρισμα το αντίστοιχο μητρώο στιβαρότητας.

```
function ElementStiffnessMatrixNURBS = getElementStiffnessMatrixNURBS(... element)
```

Τέλος, για την ολοκλήρωση της διαδικασίας Ισογεωμετρικής ανάλυσης αφού έχουν ήδη υπολογιστεί τα τοπικά μητρώα όλων των επιμέρους *elements*, καλείται μία νέα συνάρτηση στην οποία θα υπολογιστεί το καθολικό μητρώο στιβαρότητας, και αφού καθοριστούν οι συνθήκες στήριξης και οι φορτίσεις του φορέα, υπολογίζονται οι μετακινήσεις των ελεύθερων βαθμών ελευθερίας.

```
function Displacement = IsogeometricAnalysisNurbs(Ksi_CP, Heta_CP, x, penal, elementsData)
```

όπου

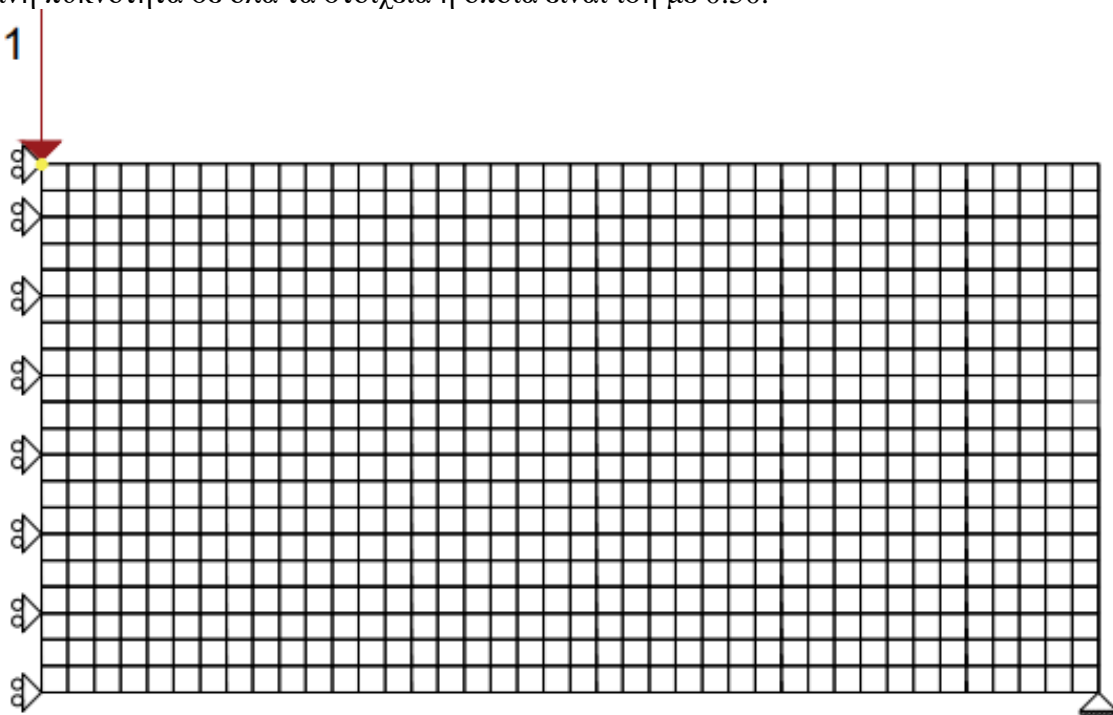
- *Ksi\_CP* : οι συντεταγμένες των *Control Points* στον άξονα  $\xi$
- *Heta\_CP* : οι συντεταγμένες των *Control Points* στον άξονα  $\eta$
- *x* : η πυκνότητα κάθε *element*
- *penal* : η τιμή της δύναμης  $p$  στην μέθοδο SIMP
- *elementsData* : η μεταβλητή που περιέχει όλες τις ιδιότητες των *elements*

## 6.6 Απλές εφαρμογές Τοπολογικής Βελτιστοποίησης

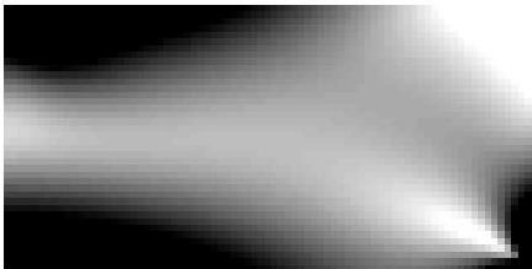
Έχοντας ολοκληρωθεί οι διαδικασίες ανάλυσης με Πεπερασμένα Στοιχεία και της Ισογεωμετρικής Ανάλυσης, θα συνδυαστούν με τη μέθοδο SIMP. Ακολουθούν κάποιες απλές εφαρμογές της Τοπολογικής Βελτιστοποίησης με Ισογεωμετρικά Πεπερασμένα Στοιχεία όπου εξετάζεται η περίπτωση Ισογεωμετρικής Ανάλυσης για συναρτήσεις πολυωνυμικού βαθμού μεγαλύτερου της μονάδας και της ανάλυσης με Πεπερασμένα Στοιχεία. Παρακάτω παρουσιάζονται τα αποτελέσματα που προέκυψαν κατά την επίλυση στο MATLAB.

### I. Finite Element Analysis

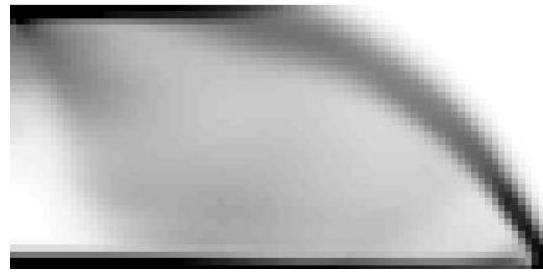
Έστω μία αμφιέρειστη δοκός με συνθήκες επίπεδης παραμόρφωσης, που αποτελείται από 80 στοιχεία στον άξονα  $x$  και άλλα 40 στον άξονα  $y$ . Για λόγους συμμετρίας επιλύσουμε το μισό φορέα. Το πάχος και το μέτρο Ελαστικότητας του φορέα λαμβάνεται ίσο με 1. Ο λόγος του Poisson είναι ίσος με 0.3. Στον άνω αριστερά κόμβο του φορέα επιβάλλεται κατακόρυφο φορτίο ίσο με τη μονάδα, με φορά προς τα κάτω. Για τη μέθοδο SIMP, τα δεδομένα που χρησιμοποιήθηκαν είναι 3 για τη τιμή του εκθέτη  $p$ , 0.30 για τον διαθέσιμο όγκο του υλικού ο οποίος είναι και σταθερός. Οι επαναλήψεις ξεκινούν με κοινή πυκνότητα σε όλα τα στοιχεία η οποία είναι ίση με 0.50.



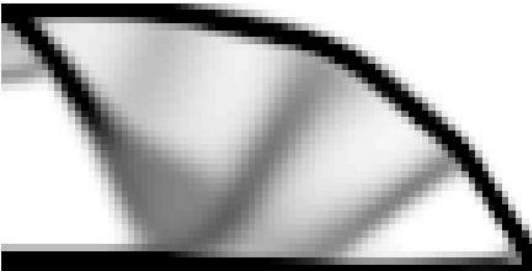
I. Finite Element Analysis



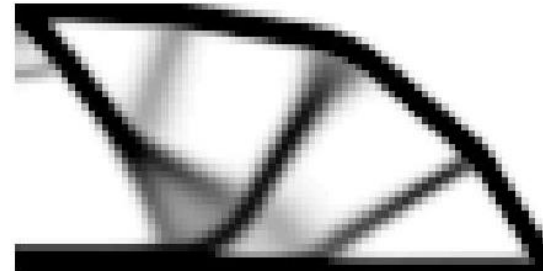
Iteration 1  
Objective Function : 363.7529



Iteration 6  
Objective Function : 507.8815



Iteration 12  
Objective Function : 240.9853



Iteration 15  
Objective Function : 178.3670



Iteration 22  
Objective Function : 131.5176



Iteration 25  
Objective Function : 128.2002



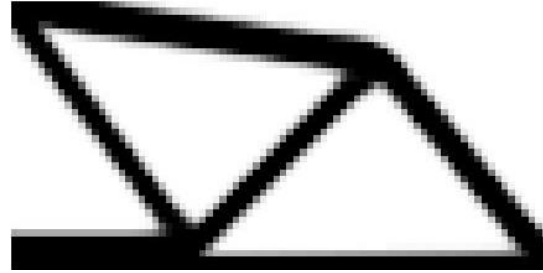
Iteration 47  
Objective Function : 125.5382



Iteration 71  
Objective Function : 122.9914



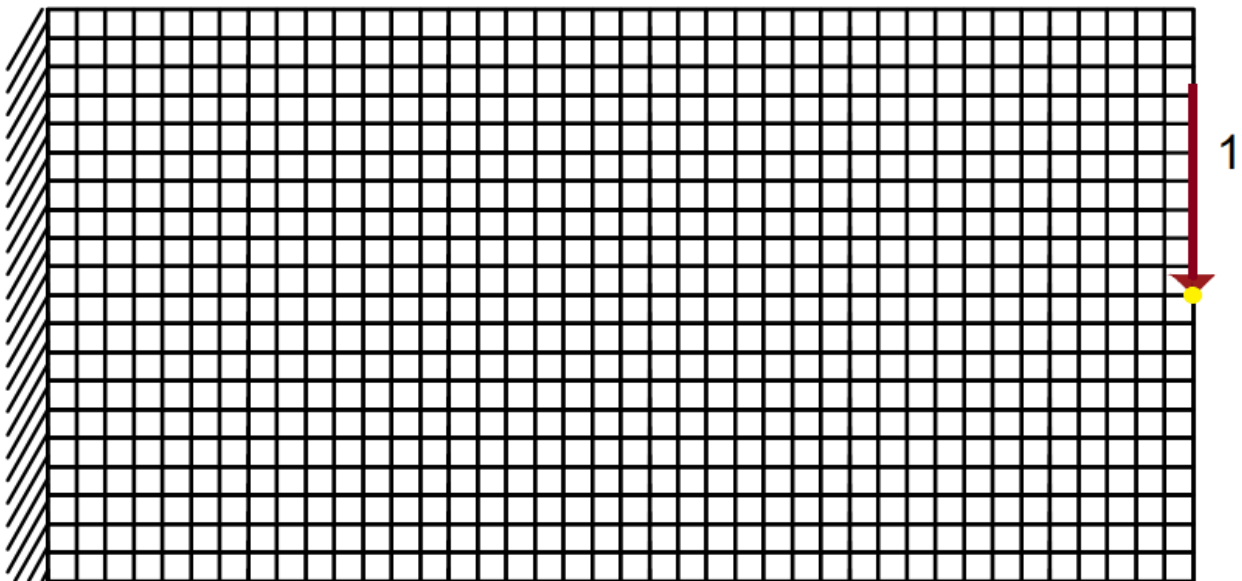
Iteration 93  
Objective Function : 122.7051



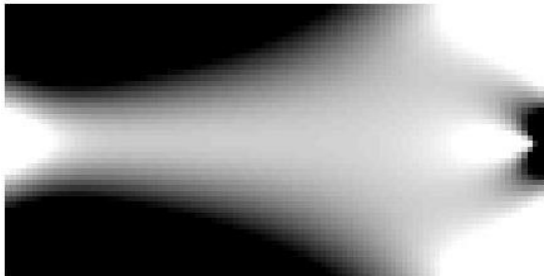
Iteration 173  
Objective Function : 120.2267

### I. Isogeometric Analysis

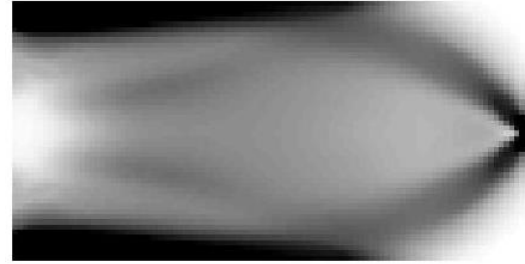
Έστω ένας πρόβλος με συνθήκες επίπεδης παραμόρφωσης, που αποτελείται από 100 στοιχεία στον άξονα  $\xi$  και άλλα 50 στον άξονα  $\eta$ . Το πάχος και το μέτρο Ελαστικότητας του φορέα λαμβάνεται ίσο με 1. Επιπλέον, ο λόγος του Poisson είναι ίσος με 0.3. Η φόρτιση του φορέα είναι κατακόρυφη και ίση με τη μονάδα, με φορά προς τα κάτω και εφαρμόζεται στον κόμβο όπως φαίνεται παρακάτω στο μοντέλο. Ο βαθμός του πολυωνύμου και στις δύο διευθύνσεις είναι ίσος με 2. Για τη μέθοδο SIMP, τα δεδομένα που χρησιμοποιήθηκαν είναι 3 για τη τιμή του εκθέτη  $p$ , 0.50 για τον διαθέσιμο όγκο του υλικού ο οποίος είναι και σταθερός. Οι επαναλήψεις ξεκινούν με κοινή πυκνότητα σε όλα τα στοιχεία  $\eta$  οποία είναι ίση με 0.50.



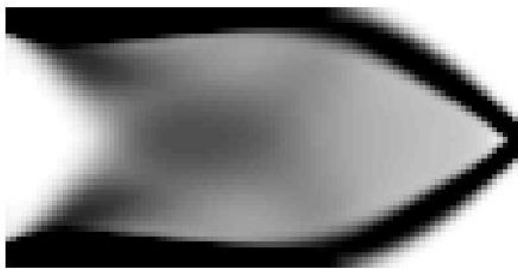
II. Isogeometric Analysis



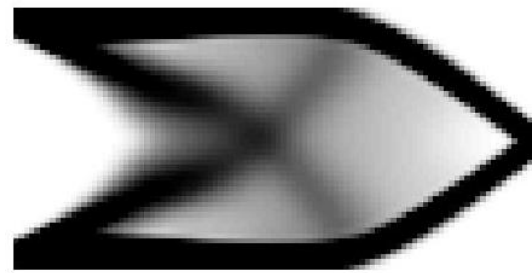
Iteration 1  
Objective Function : 301.3335



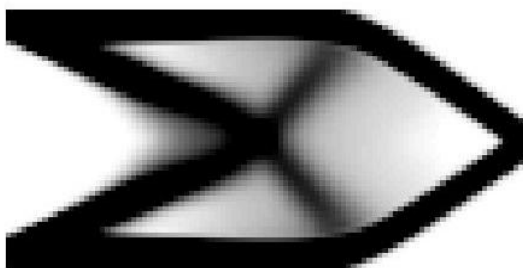
Iteration 4  
Objective Function : 114.4017



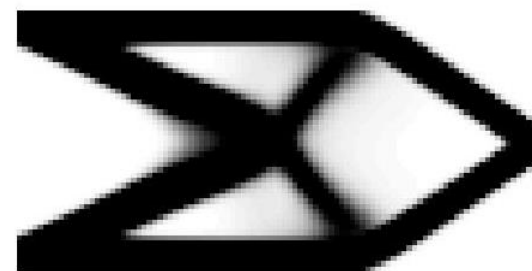
Iteration 6  
Objective Function : 94.2245



Iteration 9  
Objective Function : 77.9681



Iteration 11  
Objective Function : 71.2426



Iteration 13  
Objective Function : 65.2869



Iteration 25  
Objective Function : 60.0928



Iteration 90  
Objective Function : 60.0900



Iteration 98  
Objective Function : 60.0700



Iteration 138  
Objective Function : 60.0845

## 7 Συμπεράσματα

Σκοπός της διπλωματικής εργασίας ήταν, αφού διεξαχθούν κάποια παραδείγματα βελτιστοποίησης τοπολογίας διαφόρων περιπτώσεων ως προς την μέθοδο ανάλυσης, τον διαθέσιμο όγκο υλικού, ο οποίος παραμένει σταθερός κατά τη διαδικασία επίλυσης, και τον πολυωνυμικό βαθμό στην περίπτωση της Ισογεωμετρικής Ανάλυσης, να προκύψουν συμπεράσματα σχετικά με την εφαρμογή της Τοπολογικής Βελτιστοποίησης σε τυπικά προβλήματα αλλά και με τις μεθόδους ανάλυσης. Για το σκοπό αυτό επιλύθηκαν δύο μοντέλα προσομοίωσης, μία αμφιέριστη και ένας πρόβολος. Επιπλέον τα δύο αυτά μοντέλα εξετάστηκαν για διαφορετικό αριθμό στοιχείων διακριτοποίησης, την περίπτωση των  $80 \times 40$  στοιχείων και την περίπτωση των  $100 \times 50$  στοιχείων. Όσον αφορά στις τιμές της μεθόδου SIMP χρησιμοποιήθηκαν οι  $p = 3$  για την τιμή ποινικοποίησης της πυκνότητας και  $r = 1.50$  η τιμή της ακτίνας επιρροής κάθε στοιχείου για το φίλτρο ευαισθησίας, ενώ ο διαθέσιμος όγκος εξετάστηκε για τις περιπτώσεις των  $V = 0.30, 0.50$  και  $0.70$ . Τέλος, για την μέθοδο Ισογεωμετρικής Ανάλυσης πραγματοποιήθηκαν επιπλέον αναλύσεις για διαφορετικούς βαθμούς πολυωνύμου, συγκεκριμένα από ένα έως τέσσερα, με στόχο τη βαθύτερη διερεύνηση της μεθόδου ως προς την ακρίβειά της, για όλες τις παραπάνω περιπτώσεις,

Τα αποτελέσματα συγκρίθηκαν ως προς την τιμή της αντικειμενικής συνάρτησης  $c$  και το πλήθος των επαναλήψεων αλλά και τον τρόπο σύγκλισης στην τελική λύση. Οι παρατηρήσεις που έγιναν είναι οι εξής

### I. Σύγκριση μεθόδων ανάλυσης FEA – IGA 1<sup>ου</sup> βαθμού

Κατά την πραγματοποίηση ανάλυσης για μία αμφιέριστη δοκό και έναν πρόβολο, παρατηρήθηκαν ότι τα αποτελέσματα των δύο μεθόδων ταυτίζονται πλήρως ως προς την τιμή της αντικειμενικής συνάρτησης και τον αριθμό των επαναλήψεων.

### II. Σύγκριση ως προς τον διαθέσιμο όγκο

Σε όλα τα παραδείγματα παρατηρήθηκε πως η αύξηση όγκου οδηγεί σε μικρότερες τιμές της αντικειμενικής συνάρτησης. Επιπλέον, για την περίπτωση διαθέσιμου όγκου ίσο με  $0.50$  φαίνεται πως η μέθοδος αρχίζει να συγκλίνει στην τελική λύση πιο άμεσα.

### III. Σύγκριση μεθόδου IGA ως προς τον πολυωνυμικό βαθμό

Για τις διάφορες περιπτώσεις βαθμού πολυωνύμου παρατηρείται πως όσο αυξάνεται ο βαθμός, αυξάνεται και η τιμή της αντικειμενικής συνάρτησης. Μεγαλύτερη αύξηση φαίνεται να υπάρχει στο μοντέλο της αμφιέριστης δοκού. Ο αριθμός των απαιτούμενων επαναλήψεων δεν έδειξαν κάποια συγκεκριμένη συμπεριφορά.

Σύμφωνα λοιπόν με τις παρατηρήσεις που προέκυψαν μετά την πραγματοποίηση των αναλύσεων, τα συμπεράσματα στα οποία καταλήξαμε είναι τα εξής:

I. Στην περίπτωση γραμμικών συναρτήσεων σχήματος τα αποτελέσματα της IGA ταυτίζονται πλήρως με τα αντίστοιχα της FEA.

II. Όταν αυξάνεται ο διαθέσιμος όγκος του υλικού, ο φορέας θα γίνεται και πιο δύσκαμπτος. Γι' αυτό και παρατηρήθηκε μείωση ενδοτικότητας του φορέα.



III. Καθώς αυξάνεται ο βαθμός πολωνύμου των συναρτήσεων σχήματος το σύστημα γίνεται πιο εύκαμπτο. Έτσι εξηγείται η αύξηση της τιμής της ενδοτικότητας του συστήματος.

Γενικά παρατηρήθηκε πως καθώς οι συναρτήσεις που χρησιμοποιεί η IGA έχουν συνέχεια υψηλής τάξης, υπολογίζονται τάσεις και παραμορφώσεις με υψηλότερη ακρίβεια.

Για καλύτερη κατανόηση των παρατηρήσεων που προέκυψαν, παρατίθενται στο παράρτημα Α τα αντίστοιχα διαγράμματα για την περιγραφή της συμπεριφοράς της αντικειμενικής συνάρτησης και του τρόπου σύγκλισης κατά την επαναληπτική διαδικασία.

## 8 Βιβλιογραφία

- [1] “The Rayleigh-Ritz Method,” *Math. Sci. Eng.*, 1972, doi: 10.1016/S0076-5392(09)60387-8.
- [2] A. Hrennikoff, “Solution of Problems of Elasticity by the Frame-Work Method,” *Appl. Sci. Res.*, 1941.
- [3] R. Courant, “Variational methods for the solution of problems of equilibrium and vibrations,” *Bulletin of the American Mathematical Society*. 1943, doi: 10.1090/S0002-9904-1943-07818-4.
- [4] M. J. TURNER, R. W. CLOUGH, H. C. MARTIN, and L. J. TOPP, “Stiffness and Deflection Analysis of Complex Structures,” *J. Aeronaut. Sci.*, 1956, doi: 10.2514/8.3664.
- [5] J. H. Argyris and S. Kelsey, *Energy Theorems and Structural Analysis*. 1960.
- [6] Y. K. Zienkiewicz, O. C. Cheung, *The Finite Element Method in Structural and Continuum Mechanics*. 1967.
- [7] Μ.Παπαδρακάκης, *Ανάλυση Φορέων με τη Μέθοδο των Πεπερασμένων Στοιχείων*. 2001.
- [8] O. Bendsoe, M.P. , Sigmund, *Topology Optimization Theory, Methods, and Applications*, 2nd ed. 2003.
- [9] K. Januszkiewicz and M. Banachowicz, “Nonlinear Shaping Architecture Designed with Using Evolutionary Structural Optimization Tools,” in *IOP Conference Series: Materials Science and Engineering*, 2017, doi: 10.1088/1757-899X/245/8/082042.
- [10] M. M. Sundaram and G. K. Ananthasuresh, “Gustave Eiffel and his optimal structures,” *Resonance*, 2009, doi: 10.1007/s12045-009-0081-x.
- [11] O. Sigmund, “A 99 line topology optimization code written in matlab,” *Struct. Multidiscip. Optim.*, 2001, doi: 10.1007/s001580050176.
- [12] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, “Efficient topology optimization in MATLAB using 88 lines of code,” *Struct. Multidiscip. Optim.*, 2011, doi: 10.1007/s00158-010-0594-7.
- [13] K. Liu and A. Tovar, “An efficient 3D topology optimization code written in Matlab,” *Struct. Multidiscip. Optim.*, 2014, doi: 10.1007/s00158-014-1107-x.
- [14] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. 2009.

- [15] L. Piegl and W. Tiller, “The NURBS book, 2nd edition,” *Springer-Verlag*. 1997.
- [16] F. G. C. A. Gibson, *Concorde*. 1975.
- [17] R. L. Norton, “Cam Design and Manufacturing Handbook,” *J. Mech. Des.*, 2004, doi: 10.1115/1.1723467.
- [18] “Westlawn Institute of Marine Technology.” [Online]. Available: [www.westlawn.edu](http://www.westlawn.edu).
- [19] Ν.Μπιλάλης Ε. Μαραβελάκης, *Συστήματα CAD/CAM και τρισδιάστατη μοντελοποίηση*, 2η. 2014.
- [20] D. Tsapetis, “Isogeometric Static Analysis with T-SPLines,” National Technical University of Athens, 2014

## Παράρτημα Α. Αποτελέσματα Αναλύσεων

### Αμφιέρειστη δοκός κατακόρυφο φορτίο

1. Σύγκριση FEM – IGA (1<sup>ο</sup> βαθμού)
- a) Αμφιέρειστη δοκός 80\*40

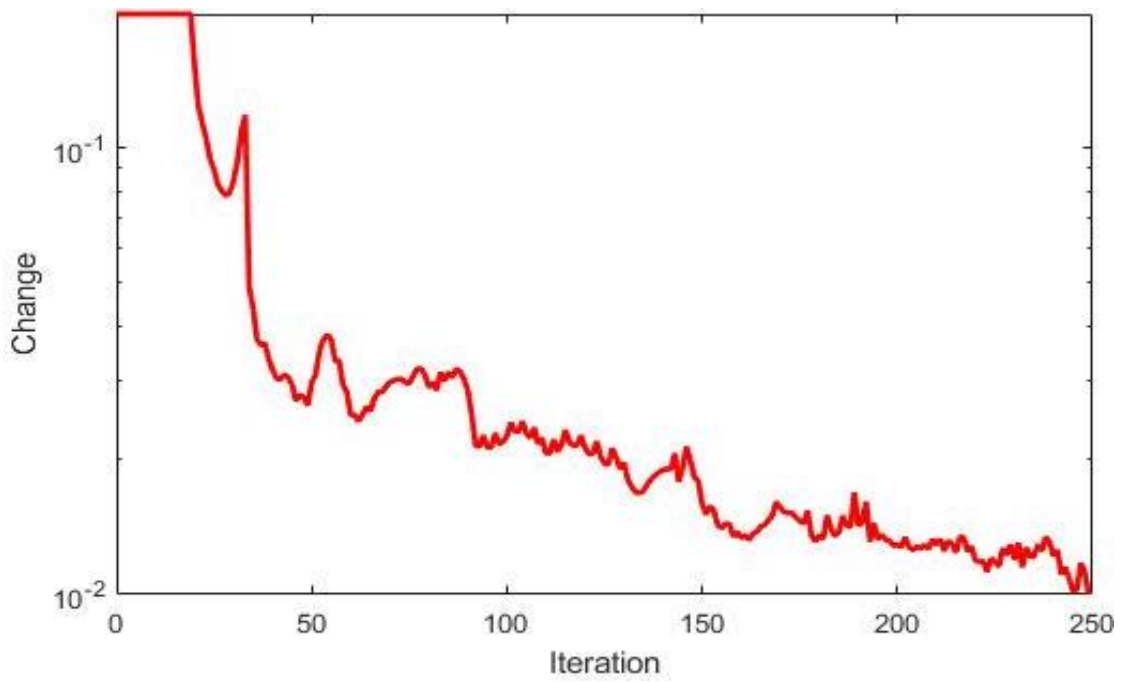
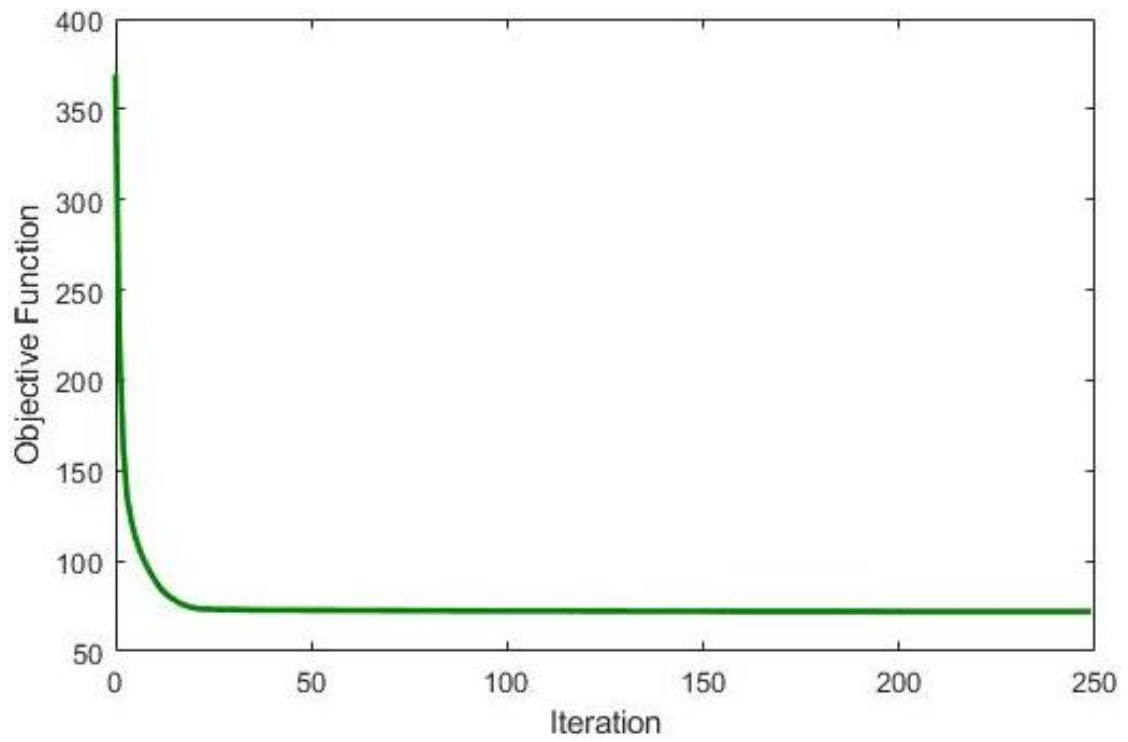
Volume = 0.50	FEM	IGA
<b>Objective function</b>	72.7718	72.7718
<b>Iterations</b>	287	287

- b) Αμφιέρειστη δοκός 100\*50

Volume = 0.50	FEM	IGA
<b>Objective function</b>	72.0281	72.0281
<b>Iterations</b>	250	250



FEM & IGA volume = 0.50 - elements 100\*50



FEM & IGA volume = 0.50 - elements 100\*50

2. Σύγκριση αναλύσεων ανάλογα τον διαθέσιμο όγκο

a) Αμφιέρειστη δοκός 80\*40

Μέθοδος Ανάλυσης	FEM	FEM	FEM	IGA	IGA	IGA
<b>Volume</b>	<b>0.30</b>	<b>0.50</b>	<b>0.70</b>	<b>0.30</b>	<b>0.50</b>	<b>0.70</b>
<b>Objective function</b>	120.2267	72.7718	54.8552	120.2267	72.7718	54.8552
<b>Iterations</b>	173	287	189	173	287	189

b) Αμφιέρειστη δοκός 100\*50

Μέθοδος Ανάλυσης	FEM	FEM	FEM	IGA	IGA	IGA
<b>Volume</b>	<b>0.30</b>	<b>0.50</b>	<b>0.70</b>	<b>0.30</b>	<b>0.50</b>	<b>0.70</b>
<b>Objective function</b>	118.9979	72.0281	55.2892	118.9979	72.0281	55.2892
<b>Iterations</b>	111	250	257	111	250	257



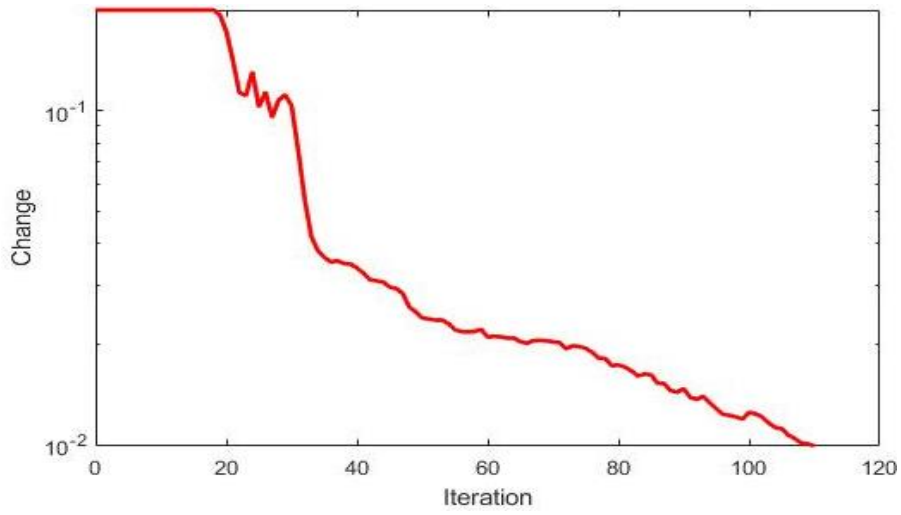
**FEM & IGA volume = 0.30 - elements 100\*50**



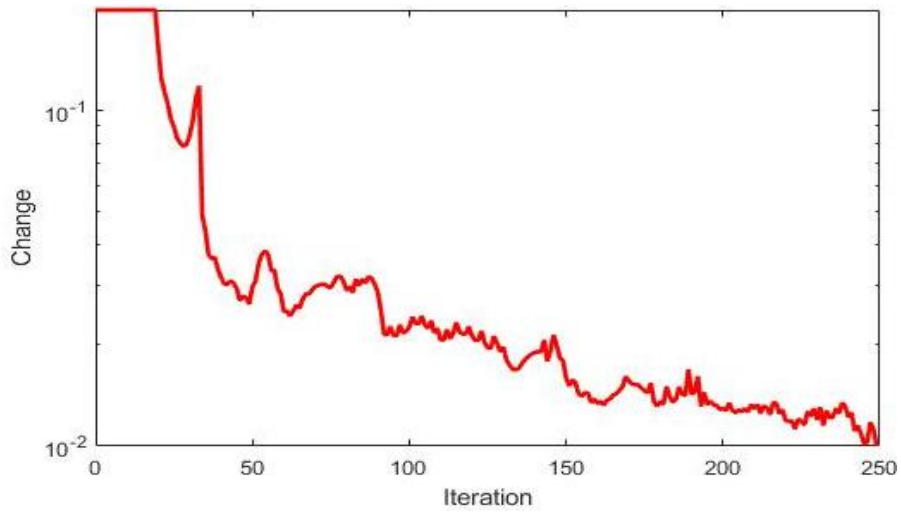
**FEM & IGA volume = 0.50 - elements 100\*50**



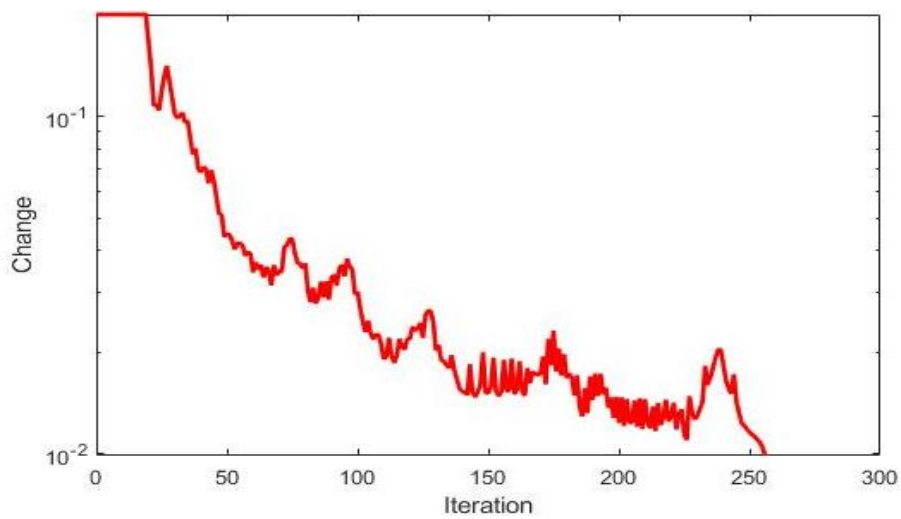
**FEM & IGA volume = 0.70 - elements 100\*50**



**FEM & IGA volume = 0.30 - elements 100\*50**



**FEM & IGA volume = 0.50 - elements 100\*50**



**FEM & IGA volume = 0.70 - elements 100\*50**



3. Σύγκριση IGA ανάλογα τον πολυωνυμικό βαθμό

a) Αμφιέριστη δοκός 80\*40, με κατακόρυφο φορτίο

- Volume = 0.30

Βαθμός πολυωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
<b>Objective function</b>	120.2267	123.0146	124.6732	125.8588
<b>Iterations</b>	173	185	176	181

- Volume = 0.50

Βαθμός πολυωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
<b>Objective function</b>	72.7718	75.5275	77.1246	78.2430
<b>Iterations</b>	287	287	278	294

- Volume = 0.70

Βαθμός πολυωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
<b>Objective function</b>	54.8552	57.6779	59.2247	60.3476
<b>Iterations</b>	189	145	225	151

b) Αμφιέρειστη δοκός 100\*50

- Volume = 0.30

Βαθμός πολωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
Objective function	118.9979	121.5506	123.3802	124.5765
Iterations	111	128	120	135

- Volume = 0.50

Βαθμός πολωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
Objective function	72.0281	74.8127	76.4217	77.5849
Iterations	250	264	281	256

- Volume = 0.70

Βαθμός πολωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
Objective function	55.2892	58.1129	59.6484	60.7611
Iterations	257	255	209	215

**Πρόβολος με κατακόρυφο φορτίο**

1. Σύγκριση FEM – IGA (1<sup>ο</sup> βαθμού)

a) Πρόβολος 80\*40

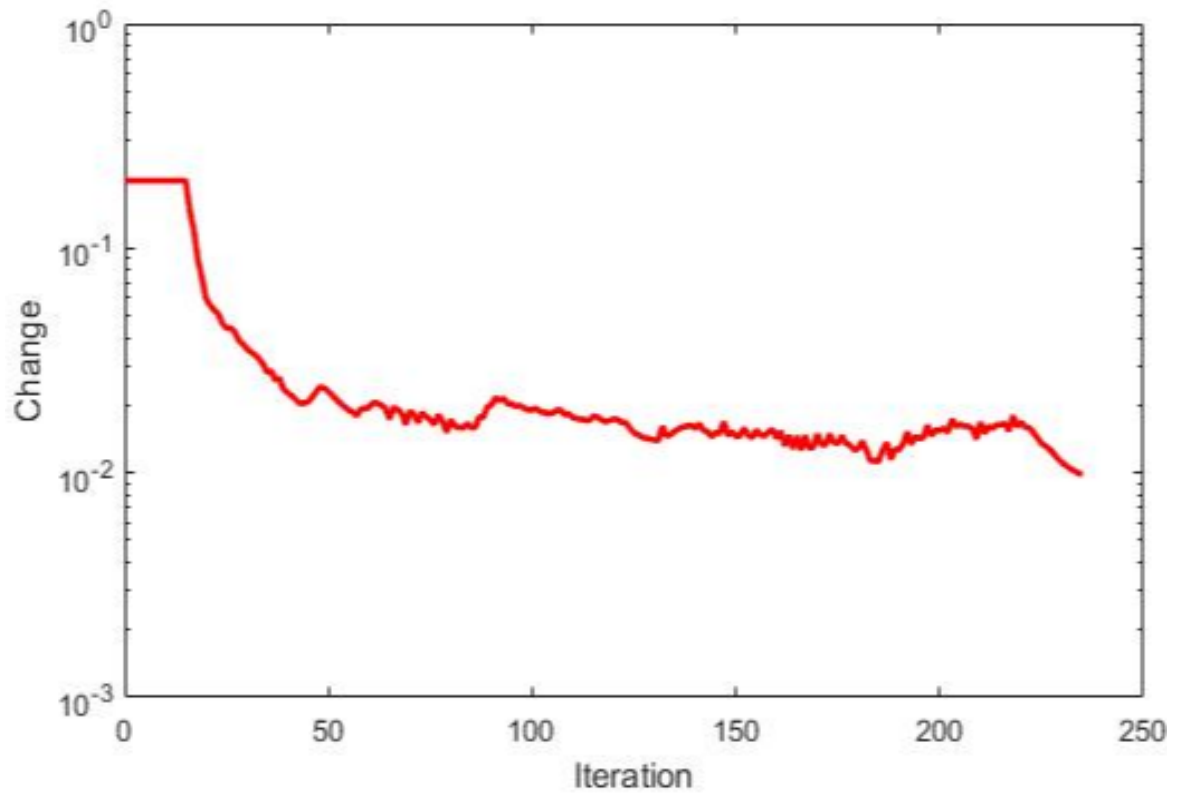
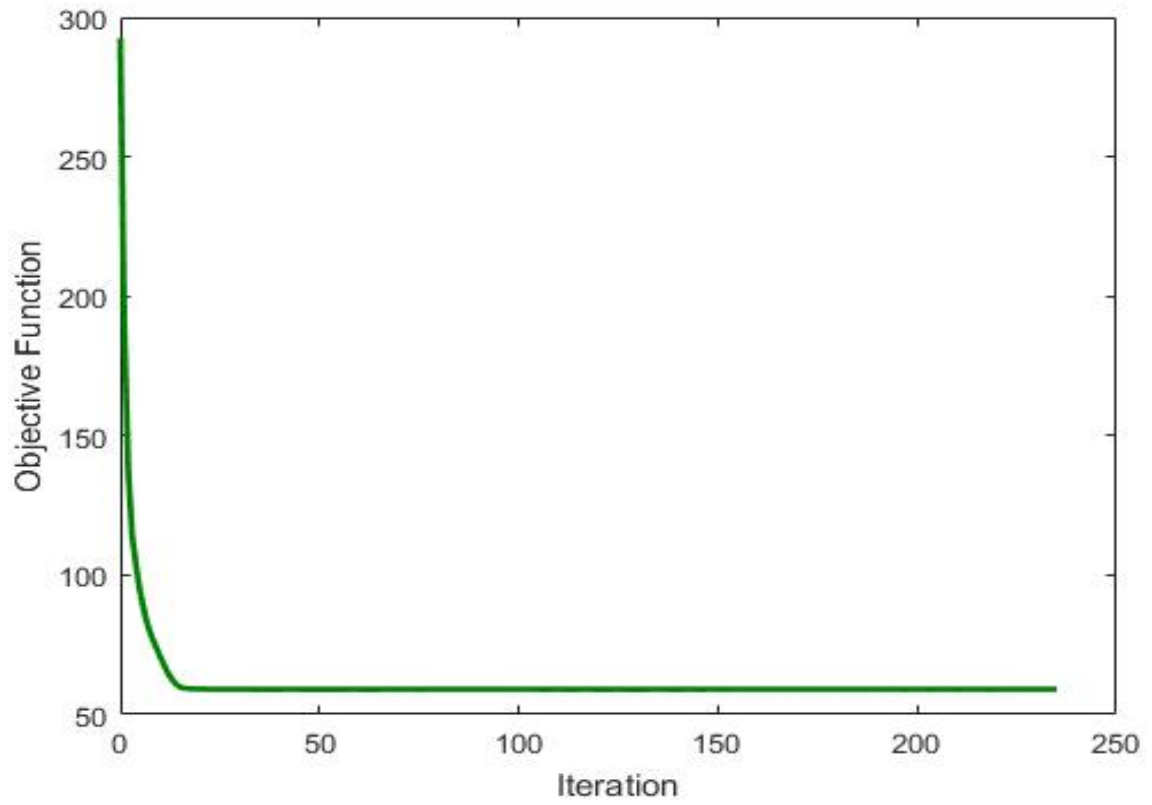
<b>Volume = 0.50</b>	<b>FEM</b>	<b>IGA</b>
<b>Objective function</b>	59.3108	59.3108
<b>Iterations</b>	172	172

b) Πρόβολος 100\*50

<b>Volume = 0.50</b>	<b>FEM</b>	<b>IGA</b>
<b>Objective function</b>	59.0378	59.0378
<b>Iterations</b>	236	236



**FEM & IGA - volume = 0.50 - elements 100\*50**



FEM - IGA volume = 0.50 - elements 100\*50

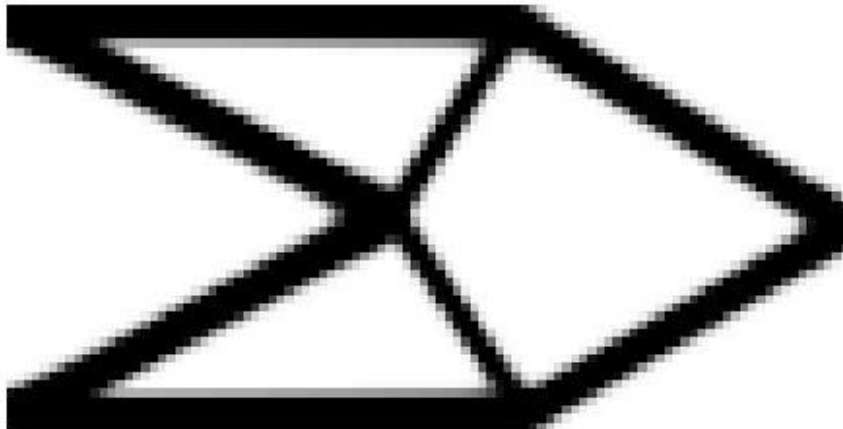
2. Σύγκριση αναλύσεων ανάλογα τον διαθέσιμο όγκο

a) Πρόβολος 80\*40

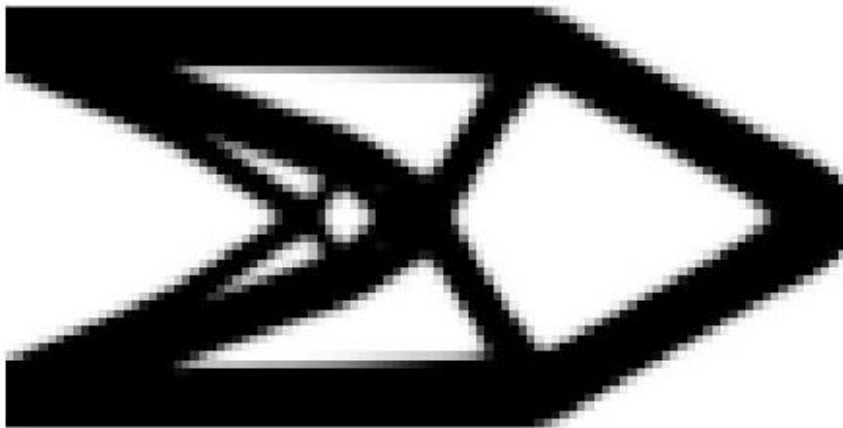
Μέθοδος Ανάλυσης	FEM	FEM	FEM	IGA	IGA	IGA
<b>Volume</b>	<b>0.30</b>	<b>0.50</b>	<b>0.70</b>	<b>0.30</b>	<b>0.50</b>	<b>0.70</b>
<b>Objective function</b>	101.0169	59.3108	44.4323	101.0169	59.3108	44.4323
<b>Iterations</b>	61	172	71	61	172	71

b) Πρόβολος 100\*50

Μέθοδος Ανάλυσης	FEM	FEM	FEM	IGA	IGA	IGA
<b>Volume</b>	<b>0.30</b>	<b>0.50</b>	<b>0.70</b>	<b>0.30</b>	<b>0.50</b>	<b>0.70</b>
<b>Objective function</b>	97.5248	59.0360	44.2683	97.5248	59.0378	44.2683
<b>Iterations</b>	75	185	113	75	236	113



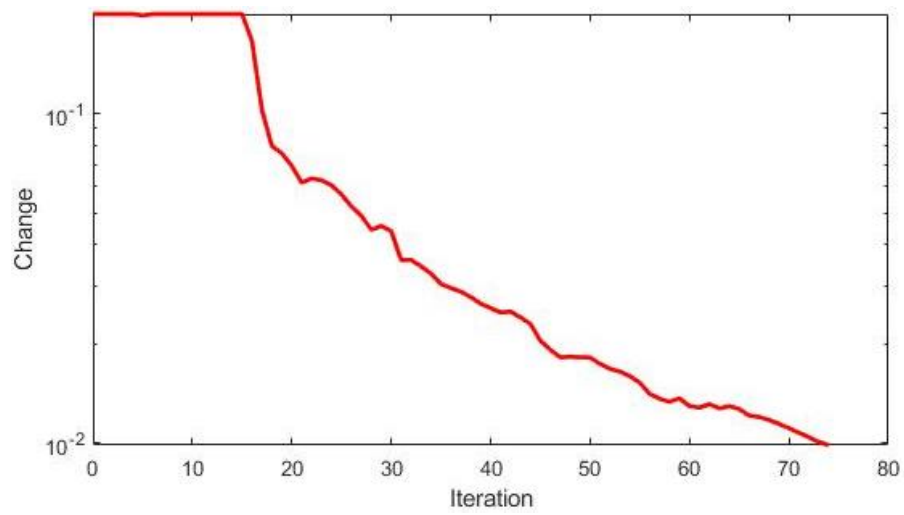
**FEM & IGA volume = 0.30 - elements 100\*50**



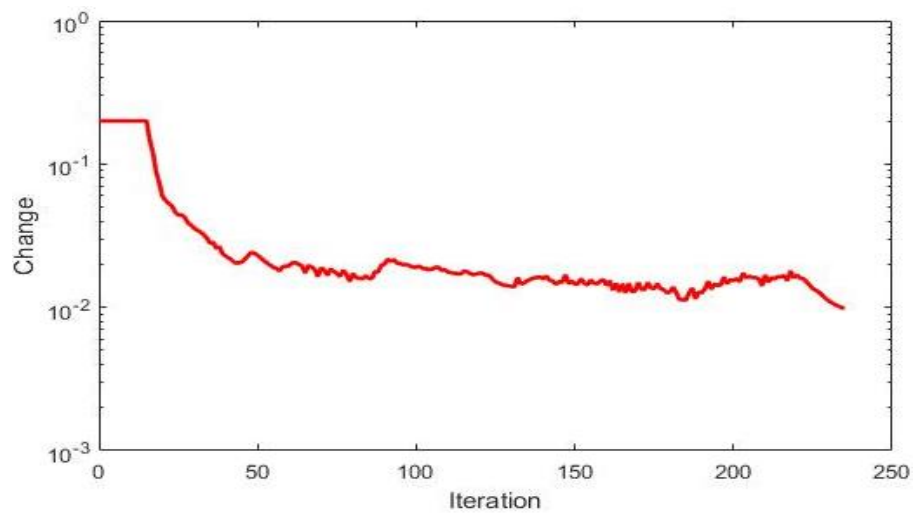
**FEM & IGA volume = 0.50 - elements 100\*50**



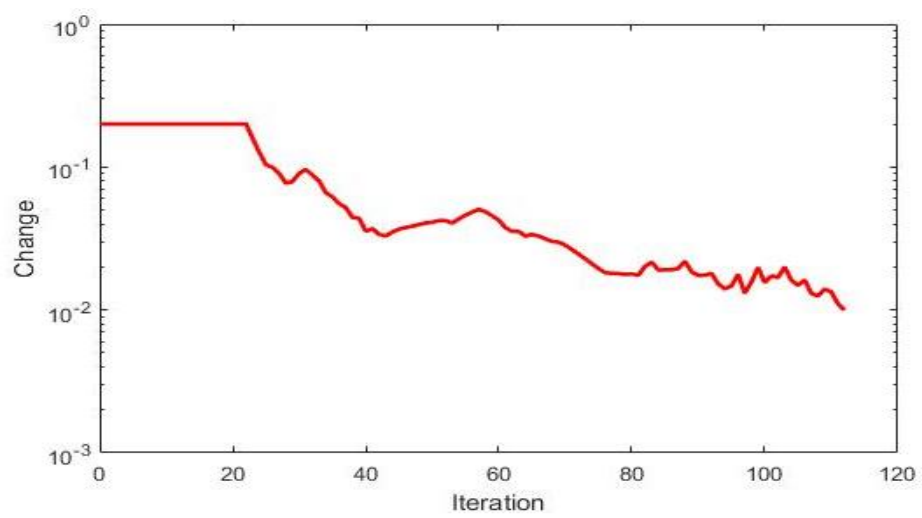
**FEM & IGA volume = 0.70 - elements 100\*50**



**FEM & IGA volume = 0.30 - elements 100\*50**



**FEM & IGA volume = 0.50 - elements 100\*50**



**FEM & IGA volume = 0.70 - elements 100\*50**

3. Σύγκριση IGA ανάλογα τον πολυωνυμικό βαθμό

a) Πρόβολος 80\*40

- Volume = 0.30

Βαθμός πολυωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
Objective function	101.0169	101.9150	104.0331	107.2072
Iterations	61	70	61	67

- Volume = 0.50

Βαθμός πολυωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
Objective function	59.3108	60.3633	61.9513	64.7595
Iterations	172	165	173	143

- Volume = 0.70

Βαθμός πολυωνύμου	1 <sup>ο</sup>	2 <sup>ο</sup>	3 <sup>ο</sup>	4 <sup>ο</sup>
Objective function	44.4323	45.4816	46.9951	49.6759
Iterations	71	73	70	71



b) Πρόβολος 100\*50

- Volume = 0.30

<b>Βαθμός πολυωνύμου</b>	<b>1<sup>ο</sup></b>	<b>2<sup>ο</sup></b>	<b>3<sup>ο</sup></b>	<b>4<sup>ο</sup></b>
<b>Objective function</b>	97.5248	98.4866	100.2657	103.2708
<b>Iterations</b>	75	71	83	82

- Volume = 0.50

<b>Βαθμός πολυωνύμου</b>	<b>1<sup>ο</sup></b>	<b>2<sup>ο</sup></b>	<b>3<sup>ο</sup></b>	<b>4<sup>ο</sup></b>
<b>Objective function</b>	59.0378	60.0845	61.6138	64.2975
<b>Iterations</b>	236	138	159	171

- Volume = 0.70

<b>Βαθμός πολυωνύμου</b>	<b>1<sup>ο</sup></b>	<b>2<sup>ο</sup></b>	<b>3<sup>ο</sup></b>	<b>4<sup>ο</sup></b>
<b>Objective function</b>	44.2683	45.3319	46.8395	49.4886
<b>Iterations</b>	113	162	133	153

# Παράρτημα Β. Κώδικες αλγορίθμων για επίλυση προβλημάτων βελτιστοποίησης με FEM και IGA

## I. Κώδικας ανάλυσης με ισοπαραμετρικά πεπερασμένα στοιχεία σε δύο διαστάσεις

```
% A TOPOLOGY OPTIMIZATION CODE FOR ISOPARAMETRIC ELEMENTS, NOVEMBER 2019%
```

```
1. function topIsoparametric(nelx, nely, volfrac, penal, rmin)
tic
x = ones (nelx, nely);
x = x * 1/2;
change = 1;
loop = 0;
objective = 0;
changeVector = 0;
[H, SH] = calculateDistance(nelx, nely, rmin);

% START ITERATION
while change > 0.01
    loop = loop+1;
    xold = x;
    Displacement = Finite_Analysis(nelx, nely, penal, x);
    [c , dc] = Objective_Function(nelx, nely, penal ,x, Displacement);
    dc = Sensitivity_Filter(nelx ,nely, rmin, dc, x, H, SH);
    x = optimalcriteria (nelx, nely, x, volfrac, dc);
    change=max(max(abs(x-xold)));
%Print Results
    disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
        ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
        ' ch.: ' sprintf('%6.3f',change )])
        objective = [objective ; c];
        changeVector = [changeVector ; change];
% PLOT DENSITIES
    plotX=zeros(nely, nelx);
    for i=1:nelx
        for j=1:nely
            plotX(j,i)=x(i,nely+1-j);
        end
    end
    colormap(gray); imagesc(-plotX); axis equal; axis tight; axis
off;pause(1e-6);
    filename = strcat(['top',num2str(loop),'.fig']);
    savefig(filename)
end
    save('test.mat')
    hold on;
    figure(2);
    plot(0:loop-1,changeVector(2:end))
    figure(3);
    plot(0:loop-1,objective(2:end))
toc
```

```

% ELEMENTSTIFFNESSMATRIXISOPARAMETRIC
2. function stiffnessMatrixIsoparametric =
ElementStiffnessMatrixIsoparametric (Coords, thickness, youngsModulus,
poissonsRatio)
E = youngsModulus*(1/((1+poissonsRatio)*(1-2*poissonsRatio)))*
[1-poissonsRatio poissonsRatio 0
poissonsRatio 1-poissonsRatio 0
0 0 (1-2*poissonsRatio)/2]; %%plane strain
h = [-0.577350269189626 ; 0.577350269189626];
x = [-0.577350269189626 ; 0.577350269189626];
ai = [1 1];
aj = [1 1];
stiffnessMatrixIsoparametric =zeros(8,8);
for i = 1:2
    for j = 1:2
        Dn = 1/4*[-(1-h(j)) 1-h(j) 1+h(j) -(1+h(j));
-(1-x(i)) -(1+x(i)) 1+x(i) 1-x(i)];
        J = Dn*Coords;
        if (det(J)<0)
            error('Negative jacobian')
        end

        B1 = 1/det(J)*[J(2,2), -J(1,2), 0, 0;
0, 0, -J(2,1), J(1,1);
-J(2,1), J(1,1), J(2,2), -J(1,2)];

        B2 = 1/4*[-(1-h(j)) 0 1-h(j) 0 1+h(j) 0 -(1+h(j)) 0
-(1-x(i)) 0 -(1+x(i)) 0 1+x(i) 0 1-x(i) 0
0 -(1-h(j)) 0 1-h(j) 0 1+h(j) 0 -(1+h(j))
0 -(1-x(i)) 0 -(1+x(i)) 0 1+x(i) 0 1-x(i) ];

        B = B1*B2;
        f = B'*E*B*thickness*det(J);
        stiffnessMatrixIsoparametric = stiffnessMatrixIsoparametric +
ai(i)*aj(j)*f;
    end
end
end

% FINITE_ANALYSIS
3. function Displacement = Finite_Analysis(nelx, nely, penal, x)
K = zeros(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
F = zeros(2*(nely+1)*(nelx+1),1);
Displacement = zeros(2*(nely+1)*(nelx+1),1);
for elx = 1:nelx
    for ely = 1:nely
        x1=elx-1;          x2=elx;          x3=elx;          x4=elx-1;
        y1=ely-1;          y2=ely-1;          y3=ely;          y4=ely;
        Coords=[x1 y1 ; x2 y2 ; x3 y3 ; x4 y4];
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1;
                2*n1+2];
        youngsModulus = 1.;          poissonsRatio = 0.3;          thickness=1;
        stiffnessMatrixIsoparametric =
            ElementStiffnessMatrixIsoparametric (Coords, thickness,
            youngsModulus, poissonsRatio) ;
        K(edof,edof) = K(edof,edof) +
            x(elx,ely)^penal*stiffnessMatrixIsoparametric;
    end
end
end

```

```

%% % DEFINE LOADS AND SUPPORTS 1.(HALF MBB-BEAM , load at mid)
LoadNode = (nely+1);
LoadDof = 2*LoadNode;
F(LoadDof,1) = -1;
fixedNode = (nely+1)*nelx+1;
fixedDofs = union([1:2:2*(nely+1)], [2*fixedNode]);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs, fixedDofs);
LoadNode = (nelx)*(nely+1)+floor((nely+1)/2);
F(LoadDof,1) = -1;

fixedNode = 1 : nely+1;
fixedDofs = [];
for fixedNode = 1 : nely+1
    fixedDofs = [fixedDofs;
                2*fixedNode-1 ;
                2*fixedNode];
end
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs, fixedDofs);

% SOLVING
K = sparse(K);
F = sparse(F);
Displacement(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
Displacement(fixedDofs,:)= 0;

```

```

% OBJECTIVE FUNCTION

```

```

4. function [c , dc ] = Objective_Function(nelx, nely, penal, x,
Displacement)
c = 0;
dc = zeros(nelx, nely);
for elx = 1 : nelx
    for ely = 1 : nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        x1=elx-1;    x2=elx;    x3=elx;    x4=elx-1;
        y1=ely-1;    y2=ely-1;    y3=ely;    y4=ely;
        Coords=[x1 y1 ; x2 y2 ; x3 y3 ; x4 y4];
        edofs = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1;
                2*n1+2];

        youngsModulus = 1.;    poissonsRatio = 0.3;    thickness=1;
        stiffnessMatrixIsoparametric = ElementStiffnessMatrixIsoparametric
            (Coords, thickness, youngsModulus, poissonsRatio);
        Ue = Displacement(edofs,1);
        c = c + x(elx,ely)^penal*Ue'*stiffnessMatrixIsoparametric*Ue;
        dc(elx,ely) = -penal*x(elx,ely)^(penal-1)*Ue'*
            stiffnessMatrixIsoparametric*Ue;
    end
end

```

```

% CALCULATE DISTANCE

```

```

5. function [H, SH] = calculateDistance(nelx, nely, rmin)
H = zeros (nelx, nely, nelx, nely);
SH = zeros(nelx, nely);
for il = 1:nelx

```

```

for j1 = 1:nely
    for i2 = 1:nelx
        for j2 = 1:nely
            distance = sqrt((i1-i2)^2+(j1-j2)^2);
            if distance <= rmin
                H(i1,j1,i2,j2) = rmin - distance;
                SH(i1,j1) = SH(i1,j1) + H(i1,j1,i2,j2);
            end
        end
    end
end
end
end

```

```
% SENSITIVITY FILTER
```

```

6. function f_filtered = Sensitivity_Filter(nelx, nely, rmin, f, x, H, SH)
f_filtered = zeros(nelx, nely);
for i1 = 1:nelx
    for j1 = 1:nely
        for i2 = 1:nelx
            for j2 = 1:nely
                f_filtered(i1,j1) = f_filtered(i1,j1)
                    + H(i1,j1,i2,j2)*x(i2,j2)*f(i2,j2);
            end
        end
    end
    f_filtered(i1,j1) = f_filtered(i1,j1)/(x(i1,j1)*SH(i1,j1)) ;
end
end

```

```
% OPTIMALITY CTITERIA
```

```

7. function xnew = optimalcriteria(nelx, nely, x, volfrac, dc)
l1 = 0; l2 = 100000; move = 0.2;
while (l2-l1 > 1e-4)
    lmid = 0.5*(l2+l1);
    for elx = 1 : nelx
        for ely = 1 : nely
            xnew(elx,ely) = max(0.001,max(x(elx,ely)-
move,min(1.,min(x(elx,ely)+move,x(elx,ely)*sqrt(-dc(elx,ely)/lmid)))));
        end
    end
    if sum(sum(xnew)) - volfrac*nelx*nely > 0;
        l1 = lmid;
    else
        l2 = lmid;
    end
end
end

```

## II. Κώδικας Ισογεωμετρικής ανάλυσης δύο διαστάσεων

```
% A TOPOLOGY OPTIMIZATION CODE FOR FINITE ISOGEOMETRIC ELEMENTS, FEBRUARY 2020%
```

```
1. function topIGA2D(Ksi_degree, Heta_degree, Ksi_elements,
Heta_elements, volfrac, penal, rmin);
tic
Ksi_KnotValueVector = calcKVV( Ksi_degree , Ksi_elements);
Heta_KnotValueVector = calcKVV( Heta_degree , Heta_elements);

Ksi_KnotVector = unique(Ksi_KnotValueVector);
elementKsi = length(Ksi_KnotVector)-1;
Heta_KnotVector = unique(Heta_KnotValueVector);
elementHeta = length(Heta_KnotVector)-1;
[elementsData , Kel]= CreateElements (Ksi_degree, Heta_degree,
Ksi_elements, Heta_elements);
[Ksi_CP Heta_CP weightfactorCP] = ControlPoints (Ksi_degree, Heta_degree,
Ksi_KnotValueVector, Heta_KnotValueVector);
[H, SH] = CalculateDistance(elementKsi, elementHeta, rmin);

x = ones (elementKsi*elementHeta,1);
x = x * 1/2;
change = 1;
loop = 0;
objective = 0;
changeVector = 0;

% START ITERATION
while change > 0.01
    loop = loop+1;
    xold = x;
    Displacement = IsogeometricAnalysisNurbs(Ksi_CP, Heta_CP,
elementsData, x, penal, Kel);
    [c, dc] = Objective_Function(elementKsi, elementHeta, penal, x,
Displacement, elementsData);
    dc =
Sensitivity_Filter(elementKsi,elementHeta,rmin,dc,elementsData,x,H, SH);
    x = calculate (elementKsi, elementHeta, volfrac, rmin, x, dc,
elementsData) ;
    change=max(max(abs(x-xold)));
    clf;
    % PRINT RESULTS
    disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
' Vol.: ' sprintf('%6.3f',sum(sum(x))/(elementKsi*elementHeta))
...
' ch.: ' sprintf('%6.3f',change)])
    objective = [objective; c];
    changeVector = [changeVector; change];

% PLOT DENSITIES
plotX=zeros(elementHeta, elementKsi);
for i=1:elementKsi
    for j=1:elementHeta
        plotX(j,i)=x((i-1)*elementHeta+elementHeta+1-j);
    end
end
colormap(gray); imagesc(-plotX); axis equal; axis tight; axis
off;pause(1e-6);
figname = strcat(['top',num2str(loop),'.fig']);
```

```

    savefig(figname)
end
save('test.mat')
hold on;
figure(2);
plot(0:loop-1,changeVector(2:end))
figure(3);
plot(0:loop-1,objective(2:end))
toc

% CALCULATE KNOT VALUE VECTOR
2. function [KVV] = calcKVV( degree , elements)
n = elements; p = degree; m = n+p+1;
[KVV] = [];
for j = (p + 2) : m
    KVV(1,j) = j - (p + 1) ;
end
for k = (m + 1) : (m + p)
    KVV(1,k) = n;
end

% FIND SPAN
3. function span = FindSpan(degree, parametricCoordinate,
KnotValueVector)
numberOfFunctions = length(KnotValueVector)-1-degree;
if (parametricCoordinate==KnotValueVector(numberOfFunctions+1))
    span= numberOfFunctions;
else
    low = degree;
    high = numberOfFunctions+1;
    mid = floor((low+high)/2);
    while
        (parametricCoordinate<KnotValueVector(mid) || parametricCoordinate>=KnotValueVector(mid+1))
        if (parametricCoordinate<KnotValueVector(mid))
            high = mid;
        else
            low = mid;
        end
        mid = floor((low+high)/2);
    end
    span = mid;
end
end

```

```

% CALCULATE B-SPLINES BASIS FUNCTIONS & DERIVATIVES
4. function derivatives = DersBasisFuns(span, parametricCoordinate,
degree, numberOfDerivatives, KnotValueVector)
derivatives = zeros(numberOfDerivatives+1, degree+1);
ndu = zeros(degree+1, degree+1);
left=zeros(degree+1,1);
right=zeros(degree+1,1);
ndu(1,1)=1.0;

for j = 1:degree
left(j+1) = parametricCoordinate-KnotValueVector(span+1-j);
right(j+1) = KnotValueVector(span+j)-parametricCoordinate;
saved = 0.0;
for r = 0:j-1
ndu(j+1,r+1) = right(r+1+1)+left(j-r+1);
temp = ndu(r+1, j-1+1)/ndu(j+1,r+1);

ndu(r+1,j+1)= saved + right(r+1+1)*temp;
saved = left(j-r+1)*temp;
end
ndu(j+1,j+1) = saved;
end

derivatives(1,:)=ndu(:,degree+1);
a = zeros(2,degree+1);
for r = 0:degree
s1 = 0;
s2 = 1;
a(1,1) = 1.0;
for k = 1:numberOfDerivatives
d = 0;
rk = r-k;
pk = degree-k;

if (r >= k)
a(s2+1,1) = a(s1+1,1) /ndu(pk+1+1, rk+1);
d = a(s2+1,1)*ndu(rk+1,pk+1);
end

if(rk >= -1)
j1 = 1;
else
j1 = -rk;
end
if (r-1 <= pk)
j2 = k-1;
else
j2 = degree-r;
end

for j = j1:j2
a(s2+1, j+1) = (a(s1+1, j+1) - a(s1+1, j - 1+1)) / ndu(pk +
1+1, rk + j+1);
d =d+ a(s2+1, j+1) * ndu(rk + j+1, pk+1);
end

if (r<=pk)
a(s2+1, k+1) = -a(s1+1, k - 1+1) / ndu(pk + 1+1, r+1);
d = d+ a(s2+1, k+1) * ndu(r+1, pk+1);
end
end

```



```

        derivatives(k+1,r+1) = d;
        temp = s1;
        s1 = s2;
        s2 = temp;
    end
end

aux = degree;
for k = 1:numberOfDerivatives
    for j = 0:degree
        derivatives (k+1,j+1) = derivatives (k+1,j+1)*aux;
    end
    aux = aux*(degree-k);
end

% CALCULATE NURBS BASIS FUNCTIONS & DERIVATIVES
5. function [dR_dKsi , dR_dHeta]= NURBSDers2D ( Ksi_degree,
Heta_degree, ...
        Ksi_parametricCoordinate,
Heta_parametricCoordinate, Ksi_KnotValueVector, Heta_KnotValueVector)

Ksi_numberOfFunctions = length(Ksi_KnotValueVector)-1-Ksi_degree;
Heta_numberOfFunctions = length(Heta_KnotValueVector)-1-Heta_degree;
Ksi_numberOfDerivatives=1;
Heta_numberOfDerivatives=1;

Ksi_span=FindSpan(Ksi_degree, Ksi_parametricCoordinate,
Ksi_KnotValueVector);
Heta_span=FindSpan(Heta_degree, Heta_parametricCoordinate,
Heta_KnotValueVector);

Ksi_functionsAndDerivatives = DersBasisFuns (Ksi_span,
Ksi_parametricCoordinate, Ksi_degree, Ksi_numberOfDerivatives,
Ksi_KnotValueVector);
Heta_functionsAndDerivatives = DersBasisFuns (Heta_span,
Heta_parametricCoordinate, Heta_degree, Heta_numberOfDerivatives,
Heta_KnotValueVector);

Ksi_NURB_weight = ones (Ksi_numberOfFunctions,1);
Heta_NURB_weight = ones (Heta_numberOfFunctions,1);

SNMw = zeros (1,1);
dWdKsi = zeros (1,1);
dWdHeta = zeros (1,1);
SNMwSquare = zeros (1,1);

indexCP=1;
for i = 1 : (Ksi_degree + 1)
    for j = 1 : (Heta_degree +1)
        SNMw = SNMw +
Ksi_functionsAndDerivatives(1,i)*Heta_functionsAndDerivatives(1,j)*Ksi_NU
RB_weight(i,1)*Heta_NURB_weight(j,1);
        SNMwSquare = SNMw*SNMw;
        dWdKsi = dWdKsi +
Ksi_functionsAndDerivatives(2,i)*Heta_functionsAndDerivatives(1,j)*Ksi_NU
RB_weight(i,1)*Heta_NURB_weight(j,1);
    end
end

```

```

        dWdHeta = dWdHeta +
Ksi_functionsAndDerivatives(1,i)*Heta_functionsAndDerivatives(2,j)*Ksi_NURB_weight(i,1)*Heta_NURB_weight(j,1);
    end
end

for i = 1 : (Ksi_degree + 1)
    for j = 1 : (Heta_degree + 1)
        dR_dKsi(indexCP) =
Ksi_NURB_weight(i,1)*Heta_NURB_weight(j,1)*(Ksi_functionsAndDerivatives(2,i)*Heta_functionsAndDerivatives(1,j)*SNMw...
-
dWdKsi*Ksi_functionsAndDerivatives(1,i)*Heta_functionsAndDerivatives(1,j)
)/SNMwSquare;
        dR_dHeta(indexCP) =
Ksi_NURB_weight(i,1)*Heta_NURB_weight(j,1)*(Ksi_functionsAndDerivatives(1,i)*Heta_functionsAndDerivatives(2,j)*SNMw...
-
dWdHeta*Ksi_functionsAndDerivatives(1,i)*Heta_functionsAndDerivatives(1,j)
)/SNMwSquare;
        indexCP=indexCP+1;
    end
end

% CALCULATE GAUSS POINTS & WEIGHT FACTORS
6. function [gaussCoordinates gaussWeights] =
UnivariateGaussPoints(degree, span)

switch degree
    case 1
        initialCoordinates=[-0.5773502691896257, +0.5773502691896257];
        initialWeight=[1.0,1.0];
    case 2
        initialCoordinates=[-0.7745966692414834, 0.0
,+0.7745966692414834];
        initialWeight=[0.5555555555555556, 0.8888888888888888,
0.5555555555555556];
    case 3
        initialCoordinates=[-0.8611363115940526 , -0.3399810435848563,
+0.3399810435848563, +0.8611363115940526];
        initialWeight=[0.3478548451374538, 0.6521451548625461 ,
0.6521451548625461 ,0.3478548451374538];
    case 4
        initialCoordinates=[-0.906179846, -0.53846931, 0,
0.53846931, 0.906179846];
        initialWeight=[0.236926885, 0.47862867, 0.568888889,
0.47862867, 0.236926885];
end

a=span(1);
b=span(2);
gaussCoordinates =(a+b)/2+(b-a)/2*initialCoordinates;
gaussWeights=(b-a)/2*initialWeight;

```

```

% CALCULATE CONTROL POINTS & WEIGHT FACTORS
7. function [Ksi_CP Heta_CP weightfactorCP] = ControlPoints
(Ksi_degree, Heta_degree, Ksi_KnotValueVector, Heta_KnotValueVector)
Ksi_KnotVector = unique(Ksi_KnotValueVector);
Heta_KnotVector = unique(Heta_KnotValueVector);
heta = Heta_KnotValueVector;
ksi = Ksi_KnotValueVector;
elementKsi = length(Ksi_KnotVector)-1;
elementHeta = length(Heta_KnotVector)-1;
i = length(Ksi_KnotValueVector) - Ksi_degree - 1 ;
j = length(Heta_KnotValueVector) - Heta_degree - 1 ;
Ksi_CP = zeros(i,1);
Heta_CP = zeros(j,1);
index = 1;id = 1;
for n = 1 : i
    for k = 1 : Ksi_degree
        Ksi_CP(index) = Ksi_CP(index) + ksi(n+k);
    end
    Ksi_CP(index) = Ksi_CP(index)/Ksi_degree;
    Index = index+1;
end
for n = 1 : j
    for k = 1 : Heta_degree
        Heta_CP(id) = Heta_CP(id) + heta(n+k);
    end
    Heta_CP(id) = Heta_CP(id)/Heta_degree;
    id=id+1;
end
weightfactorCP = ones (elementKsi*elementHeta,1);

% CALCULATE ELEMENTS DATA
8. function [elementsData, Kel] = CreateElements (Ksi_degree,
Heta_degree, Ksi_elements, Heta_elements);

Ksi_KnotValueVector = calcKVV( Ksi_degree , Ksi_elements);
Heta_KnotValueVector = calcKVV( Heta_degree , Heta_elements);

Ksi_KnotVector = unique(Ksi_KnotValueVector);
elementKsi=length(Ksi_KnotVector)-1;
Heta_KnotVector = unique(Heta_KnotValueVector);
elementHeta=length(Heta_KnotVector)-1;
[Ksi_CP Heta_CP weightfactorCP] = ControlPoints (Ksi_degree, Heta_degree,
Ksi_KnotValueVector, Heta_KnotValueVector);
id = 1;
for i = 1 : elementKsi
    for j = 1 : elementHeta
        elementsData(id).thickness = 1;
        elementsData(id).youngsModulus = 1;
        elementsData(id).Emin = 1e-9;
        elementsData(id).poissonsRatio = 0.3 ;
        elementsData(id).SpanKsi = [Ksi_KnotVector(i)
Ksi_KnotVector(i+1)];
        elementsData(id).SpanHeta = [Heta_KnotVector(j)
Heta_KnotVector(j+1)];
        elementsData(id).KnotValueVectorKsi = Ksi_KnotValueVector;
        elementsData(id).KnotValueVectorHeta = Heta_KnotValueVector;
        elementsData(id).Ksi_degree = Ksi_degree;
        elementsData(id).Heta_degree = Heta_degree;
    end
end

```

```

        [gpKsi                                weightKsi]           =
UnivariateGaussPoints(elementsData(id).Ksi_degree,
elementsData(id).SpanKsi);
        elementsData(id).Ksi_GaussPoints = gpKsi;
        elementsData(id).Ksi_GaussPoints_Weights = weightKsi;
        [gpHeta                                weightHeta]           =
UnivariateGaussPoints(elementsData(id).Heta_degree,
elementsData(id).SpanHeta);
        elementsData(id).Heta_GaussPoints_Weights = weightHeta;
        elementsData(id).Heta_GaussPoints = gpHeta;

        indexGP=1;
        for ksi = 1 : length(elementsData(id).Ksi_GaussPoints)
            for heta = 1 : length(elementsData(id).Heta_GaussPoints)
                elementsData(id).GaussPoints2D(indexGP,:) =...
                    [elementsData(id).Ksi_GaussPoints(ksi)
elementsData(id).Heta_GaussPoints(heta)
elementsData(id).Ksi_GaussPoints_Weights(ksi)*elementsData(id).Heta_Gauss
Points_Weights(heta)];
                indexGP = indexGP + 1 ;
            end
        end
        id = id + 1;
    end
end

id = 1;
for ksiCP = 1 : elementKsi
    for hetaCP = 1 : elementHeta
        elementsData(id).Ksi_ControlPoints = Ksi_CP(ksiCP : ksiCP +
Ksi_degree );
        elementsData(id).KsiCpId=ksiCP : ksiCP + Ksi_degree;
        elementsData(id).Heta_ControlPoints = Heta_CP(hetaCP : hetaCP +
Heta_degree);
        elementsData(id).HetaCpId=hetaCP : hetaCP + Heta_degree;
        id = id + 1;
    end
end

Kel=cell(elementKsi*elementHeta,1);
id = 1;
for m = 1 : elementKsi
    for n = 1 : elementHeta
        indexCP=1;
        elementsData(id).ControlPoints2D
zeros(length(elementsData(id).Ksi_ControlPoints)*length(elementsData(id).
Heta_ControlPoints),3);
        for ksi = 1 : length(elementsData(id).Ksi_ControlPoints)
            for heta = 1 : length(elementsData(id).Heta_ControlPoints)
                elementsData(id).ControlPoints2D(indexCP,:) =
[elementsData(id).Ksi_ControlPoints(ksi)
elementsData(id).Heta_ControlPoints(heta) weightfactorCP(id)];
                indexCP = indexCP + 1 ;
            end
        end
        index = 1;
        for i = 1 : length(elementsData(id).Ksi_ControlPoints)
            for j = 1 : length(elementsData(id).Heta_ControlPoints)
                indexKsi=elementsData(id).KsiCpId(i);
                indexHeta=elementsData(id).HetaCpId(j);
                elementsData(id).CpConnectivity(index)=(indexKsi-
1)*length(Heta_CP) + indexHeta;

```

```

        index = index + 1 ;
    end
end
elementsData(id).DOFs=[];
for k = 1 : length (elementsData(id).CpConnectivity)
    elementsData(id).DOFs
=[elementsData(id).DOFs,2*elementsData(id).CpConnectivity(1,k)-1
2*elementsData(id).CpConnectivity(1,k)];
end
K=getElementStiffnessMatrixNURBS (elementsData(id));
numDofs = length(Ksi_CP)*length(Heta_CP)*2;
Ktotal = sparse(numDofs, numDofs);

Ktotal(elementsData(id).DOFs,elementsData(id).DOFs)=Ktotal(elementsData(i
d).DOFs,elementsData(id).DOFs)+K;
[ row, col, value] = find(Ktotal);
Kel{id,1}=[row, col, value];
elementsData(id).ElementStiffnessMatrixNURBS =
getElementStiffnessMatrixNURBS (elementsData(id));

    id = id + 1;
end
end

% ElementStiffnessMatrixIsoparametric
9. function ElementStiffnessMatrixNURBS =
getElementStiffnessMatrixNURBS (element)
E = element.youngsModulus*(1/((1+element.poissonsRatio)*(1-
2*element.poissonsRatio)))*[1-element.poissonsRatio element.poissonsRatio
0
element.poissonsRatio 1-element.poissonsRatio 0
0 0 (1-2*element.poissonsRatio)/2]; %%plane strain
Ksi_KnotValueVector = element.KnotValueVectorKsi;
Heta_KnotValueVector = element.KnotValueVectorHeta;
numberOfKsi_ControlPoints = length(element.Ksi_ControlPoints);
numberOfHeta_ControlPoints = length(element.Heta_ControlPoints);
N = numberOfKsi_ControlPoints*numberOfHeta_ControlPoints;
elementDofs=numberOfKsi_ControlPoints*numberOfHeta_ControlPoints*2;
ElementStiffnessMatrixNURBS = zeros (elementDofs,elementDofs);
JacobianMatrix = zeros (2,2);
ElementStiffnessMatrixNURBS = zeros (elementDofs,elementDofs);
for GP = 1 : length(element.GaussPoints2D)
    gpKsi = element.GaussPoints2D(GP,1);
    gpHeta= element.GaussPoints2D(GP,2);
    [dR_dKsi , dR_dHeta]= NURBSDers2D (element.Ksi_degree,
element.Heta_degree, ...
gpKsi, gpHeta,
Ksi_KnotValueVector, Heta_KnotValueVector);
    JacobianMatrix = [dR_dKsi ; dR_dHeta]*[element.ControlPoints2D(:,1)
element.ControlPoints2D(:,2)];
    if (det (JacobianMatrix)<=0)
        error('Negative Jacobian')
    end
    inverseJacobianMatrix = inv(JacobianMatrix);
    J = JacobianMatrix;
    B1 = 1/det (J)*[J(2,2), -J(1,2), 0, 0;
0, 0, -J(2,1), J(1,1);
-J(2,1), J(1,1), J(2,2), -J(1,2)];
    B2 = zeros (4,elementDofs);
    for i = 1 : N

```

```

        B2(1,2*i-1) = dR_dKsi(i) ;
        B2(2,2*i-1) = dR_dHeta(i) ;
        B2(3,2*i) = dR_dKsi(i) ;
        B2(4,2*i) = dR_dHeta(i) ;
    end
    B = B1*B2;
    KGP
    B'*E*B*element.thickness*det(JacobianMatrix)*element.GaussPoints2D(GP,3);
    ElementStiffnessMatrixNURBS = ElementStiffnessMatrixNURBS + KGP;
end

```

```
%ISOGEOMETRIC_ANALYSIS
```

```
10. function Displacement = IsogeometricAnalysisNurbs(Ksi_CP, Heta_CP,
elementsData, x, penal, Kel)
```

```
Kel_new = Kel;
```

```
for id = 1 : length (elementsData)
```

```
    Kel_new{id}(:,3) = Kel_new{id}(:,3).*(x(id)^penal);
```

```
end
```

```
IJV = cell2mat( Kel_new );
```

```
numDofs = length(Ksi_CP)*length(Heta_CP)*2;
```

```
K = sparse(IJV(:,1),IJV(:,2),IJV(:,3),numDofs, numDofs);
```

```
allDOFs = [];
```

```
for id = 1 : length (elementsData)
```

```
    allDOFs = [allDOFs; elementsData(id).DOFs];
```

```
end
```

```
allDOFs = unique(allDOFs);
```

```
numele = length (elementsData);
```

```
F = sparse(length(allDOFs),1);
```

```
Displacement = sparse(length(allDOFs),1);
```

```
% DEFINE LOADS AND SUPPORTS Cantilever, vertical load at mid
```

```
fixedNode = 1:length(Heta_CP);
```

```
fixedDOFS = [];
```

```
for i = 1 : length(fixedNode)
```

```
fixedDOFS = [fixedDOFS
```

```
    2*fixedNode(i)-1
```

```
    2*fixedNode(i)];
```

```
End
```

```
freeDOFs = setdiff(allDOFs,fixedDOFS);
```

```
loadNode = (length(Heta_CP)*(length(Ksi_CP)-1))+floor(length(Heta_CP)/2);
```

```
loadDOF = 2 * loadNode;
```

```
F(loadDOF,1) = -1;
```

```
%%%SOLVE%%%
```

```
Displacement(fixedDOFS,:) = 0
```

```
Displacement(freeDOFs,:) = K(freeDOFs, freeDOFs) \ F(freeDOFs,:);
```

```
% OBJECTIVE FUNCTION
```

```
11. function [c, dc] = Objective_Function(elementKsi, elementHeta,
penal, x,
```

```
Displacement, elementsData)
```

```
c = 0;
```

```
dc = zeros(elementKsi*elementHeta,1);
```

```
for id = 1 : length (elementsData)
```

```
    Ue = Displacement (elementsData(id).DOFs,1);
```

```
    c = c + x(id)^penal*Ue'*elementsData(id).ElementStiffnessMatrixNURBS*Ue;
```

```

        dc(id) = -(penal*x(id)^(penal-1)*Ue'*elementsData(id).ElementStiffnessMatrixNURBS*Ue);
    end

% CALCULATE DISTANCE
12. function [H, SH]=CalculateDistance(elementsKsi, elementsHeta, rmin)
numel = elementsKsi*elementsHeta;
distanceMatrix=zeros(numel);
H = zeros (elementsKsi* elementsHeta, elementsKsi* elementsHeta);
SH = zeros(elementsKsi*elementsHeta,1);
for i = 1 : elementsKsi
    for j = 1 : elementsHeta
        indexA = (i-1)*elementsHeta + j;
        for m = 1 : elementsKsi
            for n = 1 : elementsHeta
                indexB = (m-1)*elementsHeta + n;
                distance = sqrt((i-m)^2+(j-n)^2);
                if distance <= rmin
                    H(indexA, indexB) = rmin - distance;
                    SH(indexA) = SH(indexA) + H(indexA, indexB);
                end
            end
        end
    end
end

% SENSITIVITY FILTER
13. function f_filtered = Sensitivity_Filter(elementKsi, elementHeta,
rmin, f, elementsData x, H, SH)
f_filtered = zeros(elementKsi*elementHeta,1);
for i = 1 : elementKsi*elementHeta
    for j = 1 : elementKsi*elementHeta
        f_filtered(i) = f_filtered(i)+H(i,j)*x(j)*f(j);
    end
    f_filtered(i) = f_filtered(i)/(x(i)*SH(i)) ;
end

% CALCULATE
14. function xnew = calculate (elementKsi, elementHeta, volfrac, rmin,
x, dc, elementsData)
l1 = 0; l2 = 100000; move = 0.2;
numele = elementKsi*elementHeta;
xnew = zeros (elementKsi*elementHeta,1);
while (l2-l1 > 1e-4)
    lmid = 0.5*(l2+l1);
    for id = 1 : length (elementsData)
        xnew(id) = max(0.001,max(x(id)-
move,min(1,min(x(id)+move,x(id)*sqrt(-dc(id)/lmid)))));
    end
    if sum(sum(xnew)) > volfrac*numele
        l1 = lmid;
    else
        l2 = lmid;
    end
end
end

```