ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

«ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΠΟΛΟΓΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΓΙΑ ΕΥΚΑΜΠΤΟ ΜΗΧΑΝΙΣΜΟ ΜΕ ΧΡΗΣΗ ΚΛΑΣΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ»

Κατεμής Νικόλαος-Φοίβος

Εξεταστές
Γ.Ε. Σταυρουλάκης, Καθηγητής, επιβλέπων
Α.Αντωνιάδης., Αναπληρωτής Καθηγητής
Γ.Μαρινάκης., Λέκτορας

Χανιά, Σεπτέμβριος 2010

Περιεχόμενα

Πίνακας σχημάτων και πινάκων	Σελ.

1.ΕΙΣΑΓΩΓΗ

Η παρούσα εργασία εκπονήθηκε με σκοπό την απόκτηση του διπλώματος από το τμήμα Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης. Το κύριο πρόβλημα που αντιμετωπίσαμε είναι η σύνθεση εύκαμπτων μηχανισμών με βελτιστοποίηση των διατομών των ράβδων που τους αποτελούν.

2.ΠΕΡΙΛΗΨΗ

Στο επόμενο κεφάλαιο της εργασίας υπάρχει μία σύντομη αναφορά στους εύκαμπτους μηχανισμούς. Τι είναι οι εύκαμπτοι μηχανισμοί, που χρησιμοποιούνται, ποια τα πλεονεκτήματα των εύκαμπτων μηχανισμών.

Στο τέταρτο κεφάλαιο της εργασίας περιέχονται στοιχεία θεωρίας στα οποία βασιστήκαμε για την εκπόνηση της εργασίας. Η μέθοδος των Πεπερασμένων στοιχείων , ο βέλτιστος σχεδιασμός κατασκευών καθώς και αλγόριθμοι βελτιστοποίησης παρουσιάζονται.

Στα κεφάλαια που ακολουθούν πέντε και έξι παρουσιάζονται τα αποτελέσματα για την περίπτωση μίας φόρτισης, κεφάλαιο πέντε, και σύνθεσης πολυμορφικού μηχανισμού για δύο περιπτώσεις φορτίσεων, κεφάλαιο έξι.

Στο παράρτημα παρατίθενται οι κώδικες που χρησιμοποιήσαμε για των υπολογισμό των μετακινήσεων και την εκτύπωση των γραφικών αποτελεσμάτων

3.ΕΥΚΑΜΠΤΟΙ ΜΗΧΑΝΙΣΜΟΙ

Οι εύκαμπτοι μηχανισμοί είναι παραμορφώσιμα σώματα κατάλληλης μορφής τα οποία όταν φορτιστούν κατάλληλα λειτουργούν ως μηχανισμοί, σε αντίθεση με τους παραδοσιακούς μηχανισμούς που η κινητικότητα τους προέρχεται από αρθρώσεις, έδρανα κύλισης, οδοντωτούς τροχούς, κ.α. Τα κύρια πλεονεκτήματα των μηχανισμών είναι ότι μπορούν να κατασκευαστούν χρησιμοποιώντας λιγότερα εξαρτήματα, απαιτούν σαφώς λιγότερες διαδικασίες συναρμολόγησης ενώ δεν χρειάζονται λίπανση. Μια ευρεία εφαρμογή των εύκαμπτων μηχανισμών λαμβάνει χώρα στα MEMS (Micro Electro-Mechanical Systems) τα οποία δεν μπορούν να κατασκευαστούν με συμβατικές διαδικασίες παραγωγής και συναρμολόγησης.

Η λειτουργία ενός εύκαμπτου μηχανισμού βασίζεται στην μεταφορά μιας εισερχόμενης δύναμης που ασκείται σε ένα μέρος του μηχανισμού, σε ένα άλλο σημείο με την μορφή μετακινήσεων ή μορφή εξερχόμενης δύναμης. Ο εύκαμπτος μηχανισμός αποτελείται από ένα και μόνο σώμα (μονολιθική δομή). Το σώμα αυτό θα πρέπει να είναι αρκετά εύκαμπτο ώστε να μπορεί να μεταφέρει τις εισερχόμενες φορτίσεις σε μετακινήσεις, άλλα ταυτόχρονα θα πρέπει να είναι αρκετά στιβαρό ώστε να μπορεί να αντέξει τις φορτίσεις αυτές. Επίσης κατά τον σχεδιασμό ενός εύκαμπτου μηχανισμού και την μελέτη της κινητικότητας του θα πρέπει να ληφθεί υπόψιν και ο παράγοντας της κόπωσης. Αν οι απώλειες ενέργειας και ο παράγοντας αδράνειας θεωρηθούν αμελητέα σε ένα εύκαμπτο μηχανισμό η εισερχόμενη ενέργεια είναι ίση με το άθροισμα της εξερχόμενης ενέργειας και αυτής που είναι αποθηκευμένη στον μηχανισμό (strain energy). Ο λόγος της (εξερχόμενης) δύναμης στο σημείο εξόδου (output point) του μηχανισμού προς την εισερχόμενη δύναμη στο σημείο εισόδου (input point) ορίζεται ως μηχανικό πλεονέκτημα, (Mechanical Advantage, MA) ενώ ο λόγος της εξερχόμενης μετακίνησης στο σημείο εξόδου προς την μετακίνηση στο σημείο εφαρμογής της εισερχόμενης δύναμης ορίζεται ως το Γεωμετρικό Πλεονέκτημα, (Geometric Advantage, GA).

$$MA = F_{out}/F_{in}$$
$$GA = U_{out}/U_{in}$$

4. ΘΕΩΡΗΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ

4.1 Η ΜΕΘΟΔΟΣ ΤΩΝ ΠΕΠΕΡΑΣΜΕΝΩΝ ΣΤΟΙΧΕΙΩΝ

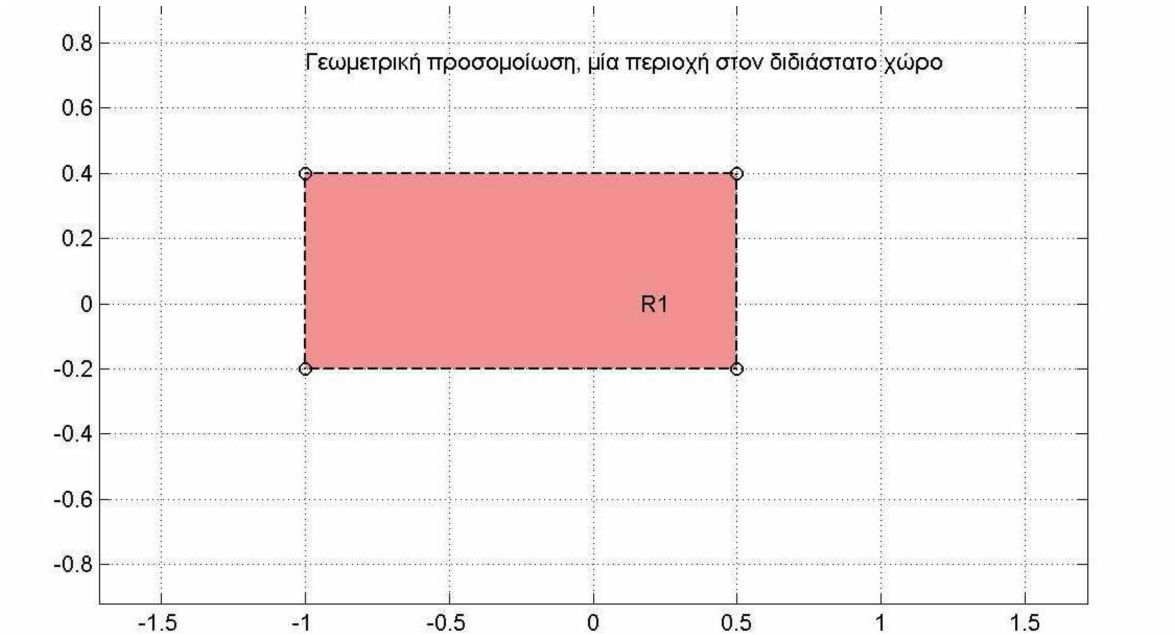
Η αναλυτική λύση των εξισώσεων με τις οποίες περιγράφονται τα διάφορα τεχνικά προβλήματα είναι δυνατή μόνο σε ειδικές περιπτώσεις, όπου οι καταπονήσεις και τα γεωμετρικά σχήματα είναι πάρα πολύ απλά. Η ανάγκη για επίλυση περισσότερο πολύπλοκων προβλημάτων , οδήγησε στην ανάπτυξη διάφορων προσεγγιστικών μεθόδων. Μια τέτοια μέθοδος είναι η μέθοδος των Πεπερασμένων Στοιχείων. Είναι μεν προσεγγιστική μέθοδος, αλλά μπορεί να δώσει αξιόπιστα αποτελέσματα και έχει το πλεονέκτημα ότι μπορεί να εφαρμοστεί σε όλα τα προβλήματα. Το μειονέκτημά της είναι οι αυξημένες απαιτήσεις σε υπολογιστική ισχύ, ιδίως όταν εφαρμόζεται σε σύνθετα μοντέλα.

Η μέθοδος των Πεπερασμένων Στοιχείων είναι μια εξέλιξη των μητρωϊκων μεθόδων που έγινε από επιστήμονες όπως ο Αργύρης Ι., ο Clough, ο Ritz και άλλοι. Οι βασικές ιδέες προήλθαν στις αρχές της δεκαετίας του 40, από εξελίξεις στην δομική ανάλυση αεροσκαφών. Αρχικά ο Hrenikoff χρησιμοποίησε τη “Μέθοδο των Δικτυωμάτων” , αργότερα O Turner δημιούργησαν μητρώα ακαμψίας για δικτυώματα, δοκούς και άλλα στοιχεία. Ο όρος Πεπερασμένα στοιχεία χρησιμοποιήθηκε το 1960. Οι μαθηματικές, βέβαια, βάσεις για την σημερινή μορφή της μεθόδου μπήκαν την δεκαετία του 70.

Πλέον αποτελεί ένα ισχυρό εργαλείο για την αριθμητική επίλυση ενός μεγάλου φάσματος προβλημάτων μηχανικού. Οι εφαρμογές εκτείνονται από την παραμόρφωση και ανάλυση τάσεων σε αυτοκίνητα, αεροπλάνα, κτίρια και γέφυρες, μέχρι την ανάλυση πεδίων ροής θερμότητας, ροής υγρών, μαγνητικής ροής, κ.α. Με την εξέλιξη των υπολογιστικών συστημάτων και των συστημάτων CAD, σύνθετα προβλήματα μπορούν να μοντελοποιηθούν πολύ εύκολα. Με αυτή τη μέθοδο μια πολύπλοκη περιοχή, διακριτοποιείται σε απλά γεωμετρικά σχήματα, τα οποία ονομάζονται Πεπερασμένα Στοιχεία (Finite Elements). Μια διαδικασία σύνθεσης, η οποία θεωρεί φορτία και περιορισμούς, έχει ως αποτέλεσμα ένα σύνολο εξισώσεων. Η επίλυση αυτών, δίνει κατά προσέγγιση(με αρκετά μεγάλη ακρίβεια) τη συμπεριφορά του αρχικού πολύπλοκου μοντέλου.

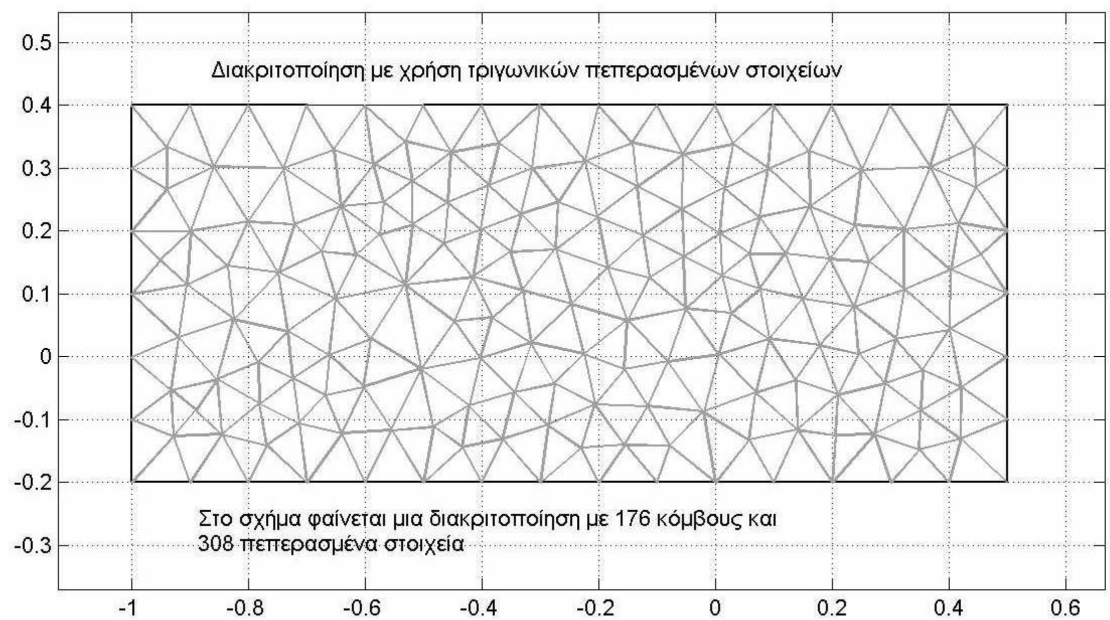
Για να εφαρμοστεί η μέθοδος απαιτούνται τα εξής στάδια:

- Εισαγωγή της γεωμετρίας της κατασκευής



Σχήμα 1: *Γεωμετρία κατασκευής*

- Χωρίζεται το μοντέλο σε Πεπερασμένα Στοιχεία και αφού ετοιμαστεί το πλέγμα επιλέγεται το είδος της επίλυσης με ταυτόχρονη εισαγωγή επιπλέον δεδομένων (pre processor πρόγραμμα)



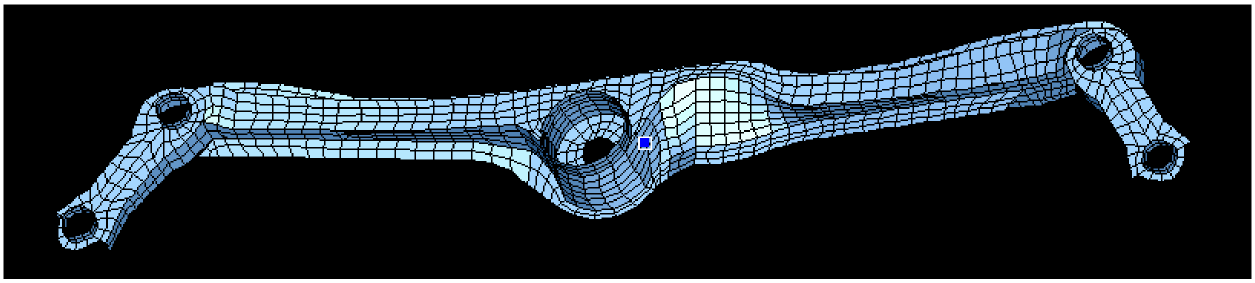
Σχήμα 2 : *Διακριτοποίηση κατασκευής*

- Γίνεται η επίλυση του προβλήματος με αριθμητικές μεθόδους (solver πρόγραμμα)

  Εδώ με την βοήθεια της σχέσης $K * u = p$, επιλύουμε ένα σύνολο εξισώσεων της μορφής $u = K^{-1} * p$, όπου *u* η μετακίνηση κάποιου κόμβου.

- Τέλος, υπάρχει δυνατότητα να εμφανίζονται τα αποτελέσματα (post processor πρόγραμμα)

Στο σχήμα της παρακάτω εικόνας φαίνεται ένα κομμάτι από το πλαίσιο ενός οχήματος, αναλυμένο με την μέθοδο των πεπερασμένων στοιχείων, έτοιμο για επίλυση. Παρατηρούμε ότι η πολύπλοκη, αρχική γεωμετρία του κομματιού, απλοποιήθηκε από πολλά μικρά, εύκολα διαχειρίσιμα, γεωμετρικά σχήματα.



Σχήμα 3: *Κομμάτι από πλαίσιο οχήματος, αναλυμένο με την μέθοδο Πεπερασμένων Στοιχείων. Πηγή: www.math.tuberlin.de*

Υπάρχουν δύο προσεγγίσεις της μεθόδου. Η προσέγγιση Galerkin και η προσέγγιση της Δυναμικής Ενέργειας.

4.1.1 Η ΠΡΟΣΕΓΓΙΣΗ ΤΗΣ ΔΥΝΑΜΙΚΗΣ ΕΝΕΡΓΕΙΑΣ

Στη μηχανική των στερεών, το πρόβλημά μας είναι να προσδιορίσουμε την μετατόπιση (τάσεις → παραμορφώσεις → μετατοπίσεις) ενός σώματος, που ικανοποιεί τις εξισώσεις ισορροπίας. Αυτό προϋποθέτει την επίλυση μερικών διαφορικών εξισώσεων δεύτερης τάξης. Η λύση του συνόλου των εξισώσεων ορίζεται ως ακριβής λύση. Τέτοιες λύσεις υπάρχουν σε απλές γεωμετρικές μορφές και απλές συνθήκες φόρτισης. Για προβλήματα, όμως, με σύνθετες γεωμετρίες, συνοριακές συνθήκες και συνθήκες φόρτισης, η επίτευξη τέτοιων λύσεων είναι σχεδόν αδύνατη. Εδώ βρίσκουν εφαρμογή οι προσεγγιστικές μέθοδοι επίλυσης και συνήθως εφαρμόζουν την προσέγγιση της Δυναμικής Ενέργειας, Π.

Η συνολική ενέργεια ενός ελαστικού σώματος ορίζεται ως το άθροισμα της Ενέργειας Παραμόρφωσης U και της ικανότητας παραγωγής έργου WP.

Π = U + WP ή

$$\Pi = \frac{1}{2}\int_{v} \sigma^{T} \varepsilon dV - \int_{v} u^{T} f dV - \int_{v} u^{T} T dS - \sum_{i} u_{i}^{T} P_{i}$$

Όπου V: όγκος του σώματος

S: επιφάνεια του σώματος

T: δύναμη ανά μονάδα επιφάνειας

u: παραμόρφωση σημείου

f: κατανεμημένη δύναμη ανά μονάδα όγκου

P: φορτίο

σ: τάση

ε: παραμόρφωση

Από τα παραπάνω ορίζεται η αρχή της ελάχιστης Δυναμικής Ενέργειας που λέει ότι: "*από όλα τα πεδία επιτρεπτών μετακινήσεων, αυτά τα οποία αντιστοιχούν σε ισορροπία παρουσιάζουν ακρότατα ολικής Δυναμικής Ενέργειας*".

4.2 ΒΕΛΤΙΣΤΟΣ ΣΧΕΔΙΑΣΜΟΣ ΚΑΤΑΣΚΕΥΩΝ

Μια από τις κυριότερες εκφράσεις της επιστήμης του Μηχανικού είναι ο σχεδιασμός. Από τη πρώτη στιγμή που εμφανίστηκε αυτή η έννοια, στόχος της έγινε ο "*τρόπος να γίνει κάτι*" για να καλύπτει ορισμένες ανάγκες, με τα εκάστοτε διαθέσιμα μέσα.

Οι κυριότερες φάσεις του σχεδιασμού είναι:

1. η αναγνώριση της ανάγκης για σχεδιασμό (καθορισμός προβλήματος)
2. το σχέδιο δράσης
3. η συλλογή εναλλακτικών λύσεων.

Παραδοσιακά η επιλογή της καλύτερης από τη συλλογή των εναλλακτικών λύσεων, αποτελεί το κομμάτι του Βέλτιστου Σχεδιασμού.

Αυτό το κομμάτι μπορεί να προσεγγιστεί αν απαντήσουμε στα παρακάτω ερωτήματα.

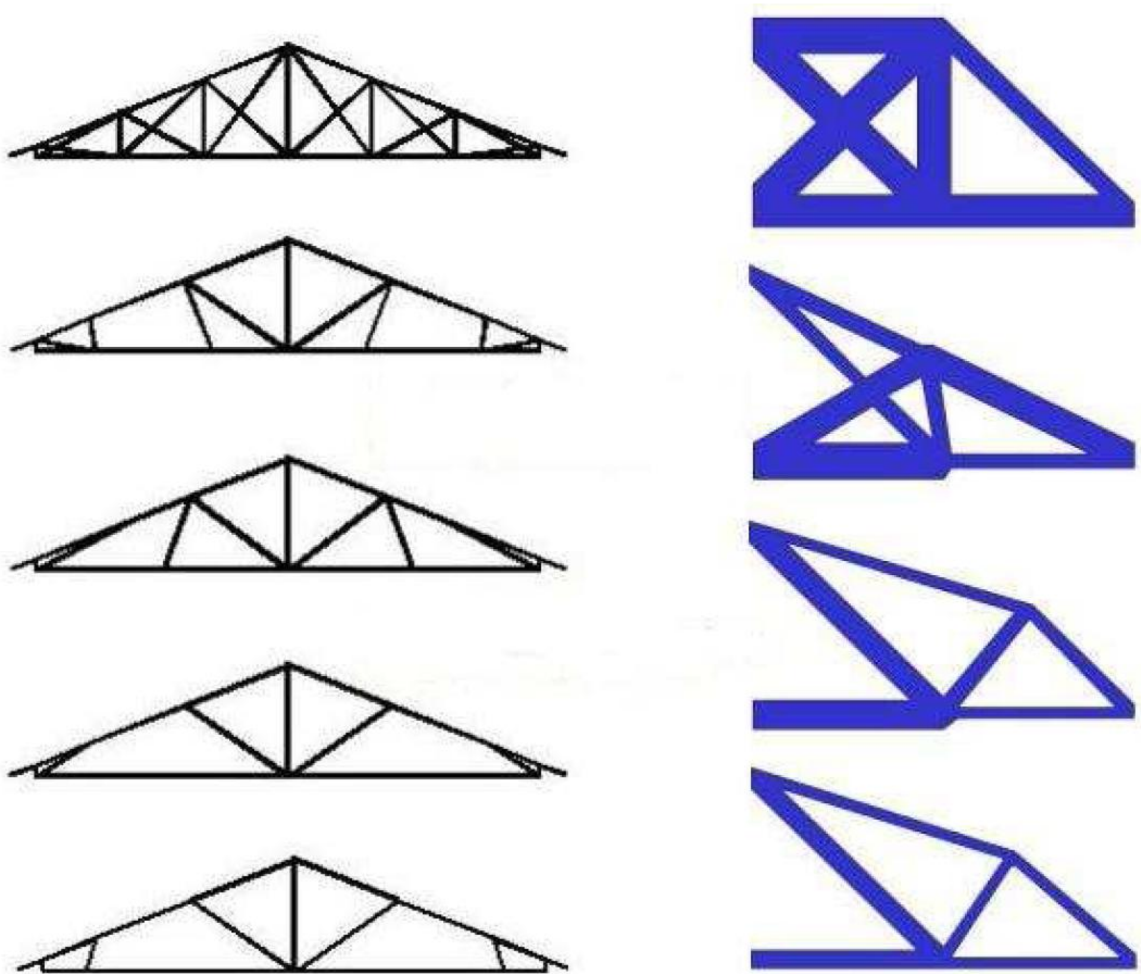
- Πως περιγράφουμε τον διαφορετικό σχεδιασμό; (παράμετροι, μεταβλητές σχεδιασμού)

- Ποια τα κριτήρια (objective criteria) μας για την επιλογή της καλύτερης/βέλτιστης λύσης;
- Ποια τα διαθέσιμα μέσα; (περιορισμοί)

4.2.1 ΤΟΠΟΛΟΓΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Η Τοπολογική Βελτιστοποίηση αποτελεί ένα από τα σημαντικότερα προβλήματα του μηχανικού και ορίζεται ως η βέλτιστη διανομή περιορισμένου υλικού σε μια συγκεκριμένη επιφάνεια. Ως παράδειγμα δίνεται η παρακάτω γέφυρα.

Αρχικά φαίνεται η γέφυρα με όλα τα στοιχεία της και στο τέλος μια απλούστερη κατασκευή με ίδια ανοχή σε φορτίσεις, τάσεις, κ.α. Τα ενδιάμεσα σχήματα μας δίνουν τις ενδιάμεσες φάσεις έως ότου προσεγγίσουμε το τελικό αποτέλεσμα. Έτσι για παράδειγμα πετυχαίνουμε ελαφρότερη κατασκευή, ή κατασκευή με λιγότερη δαπάνη υλικού τις ίδιες ανοχές, σε συγκεκριμένη φόρτιση, με την αρχική.



Σχήμα 4: *Φάσεις Τοπολογικής Βελτιστοποίησης*

Η Τοπολογική Βελτιστοποίηση κατασκευών διαφέρει από την απλή βελτιστοποίηση όσον αφορά την πολυπλοκότητα της. Υπάρχουν δύο πιθανά προβλήματα που δικαιολογούν αυτό το γεγονός. Το πρώτο είναι ότι τα μοντέλα από μόνα τους διαφοροποιούνται κατά τη διάρκεια της σχεδιαστικής διαδικασίας και δεύτερο είναι ότι ο αριθμός των συνδέσεων των στοιχείων αυξάνεται με μεγάλο ρυθμό καθώς αυξάνουμε του κόμβους σύνδεσης.

4.3 ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Ένας Αλγόριθμος βελτιστοποίησης αποτελεί μια αριθμητική μέθοδο για την εύρεση μια τιμής x^* , ώστε το αποτέλεσμα μιας αντικειμενικής συνάντησης $f(x)$ να είναι βέλτιστο (ελάχιστο ή μέγιστο). Πιθανό είναι να υπάρχουν και κάποιοι περιορισμοί όσον αφορά τις τιμές του x .

Η μαθηματική έκφραση αυτού είναι η παρακάτω:

$$\text{Minimize} \quad f(x)$$

$$\text{με περιορισμούς} \quad \begin{array}{l} h(x) = 0 \\ g(x) \leq 0 \end{array}$$

με $x \in X \subseteq R^n$,

όπου X ένα υποσύνολο του n-διάστατου χώρου R^n .

4.3.1. ΑΛΓΟΡΙΘΜΟΣ QUASI – NEWTON

Στην βελτιστοποίηση οι μέθοδοι Quasi – Newton (γνωστή και ως variable metric μέθοδος) είναι αλγόριθμοι για την εύρεση τοπικών ελάχιστων και τοπικών μέγιστων σε μια συνάρτηση. Βασίζονται στη μέθοδο του Newton, η οποία προσπαθεί να προσδιορίσει ένα στάσιμο σημείο, όπου η πρώτη παράγωγος είναι ίση με μηδέν. Αυτή η μέθοδος υποθέτει ότι μια συνάρτηση, σε μια περιοχή γύρω από το βέλτιστο, μπορεί, τοπικά, να προσεγγιστεί σαν τετραγωνική (quadratic). Έτσι με την βοήθεια της πρώτης και της δεύτερης παραγώγου, μπορεί να βρεθεί το στάσιμο σημείο.

Ας προσπαθήσουμε να περιγράψουμε την μέθοδο. Αν ένας πραγματικός αριθμός x^* είναι στάσιμο για μια συνάρτηση $f(x)$, τότε ο x^* είναι ρίζα της $f'(x)$. Το ανάπτυγμα του Taylor για την $f(x)$ θα δίνεται από τον παρακάτω τύπο,

$$f(x+\Delta x) = f(x) + f^{'}(x)\Delta x + \frac{1}{2} f^{''}(x)\Delta x^2$$

και επιτυγχάνει το ακρότατο όταν το Δ*x* επιλύει την γραμμική εξίσωση,

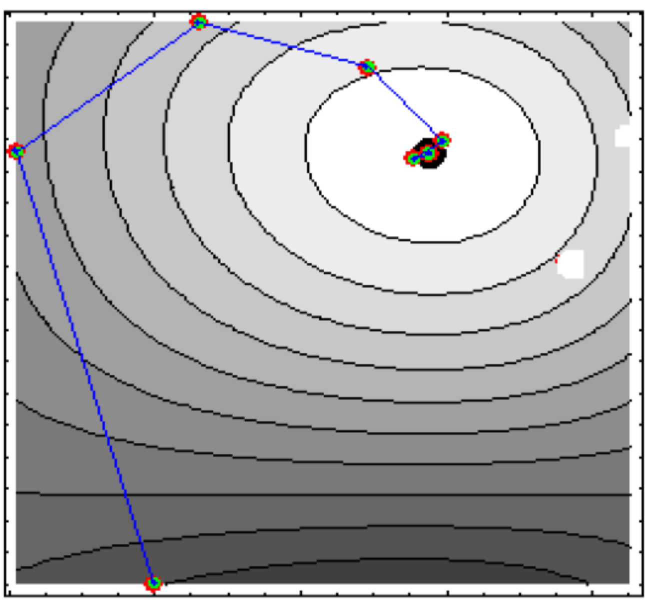
$$f^{'}(x) + f^{''}(x)\Delta x = 0$$

και η $f^{''}(x)$ είναι θετική. Γι' αυτό το λόγο, εφόσον η $f(x)$ είναι διπλά διαφορήσιμη, και η αρχική υπόθεση x_0 (τιμή από την οποία ξεκινά ο αλγόριθμος), έχει επιλεχθεί κοντά στο x^*, η ακολουθία x_n ορίζεται από τον παρακάτω τύπο,

$$x_{n+1} = x_n - \frac{f^{'}(x_n)}{f^{''}(x_n)}, n \geq 0$$

και συγκλίνει στο x^*.

Παρακάτω φαίνεται, γραφικά, πως ο αλγόριθμος αυτός συγκλίνει στο x^*.



Σχήμα 5 : *Σύγκλιση με Quasi-Newton . Πηγή: http://documents.wolfram.com/mathematica/BuiltinFunctions/AdvancedDocumentation/Optimization.html*

Αυτή η επαναληπτική ιδέα μπορεί να γενικευθεί σε πολλές διαστάσεις, αντικαθιστώντας την πρώτη παράγωγο με βαθμωτή μεταβολή (gradient), $\nabla f(x)$ και την δεύτερη παράγωγο με τον αντίστροφο του Hessian πίνακα, $Hf(x)$. Έτσι έχουμε,

$$x_{n+1} = x_n - [Hf(x_n)]^{-1} \nabla f(x_n), n \geq 0 .$$

Συνηθίζεται η μέθοδος Quasi-Newton να περιλαμβάνει ένα μικρό βήμα, μεγαλύτερο του μηδενός, αντί για γ=1,

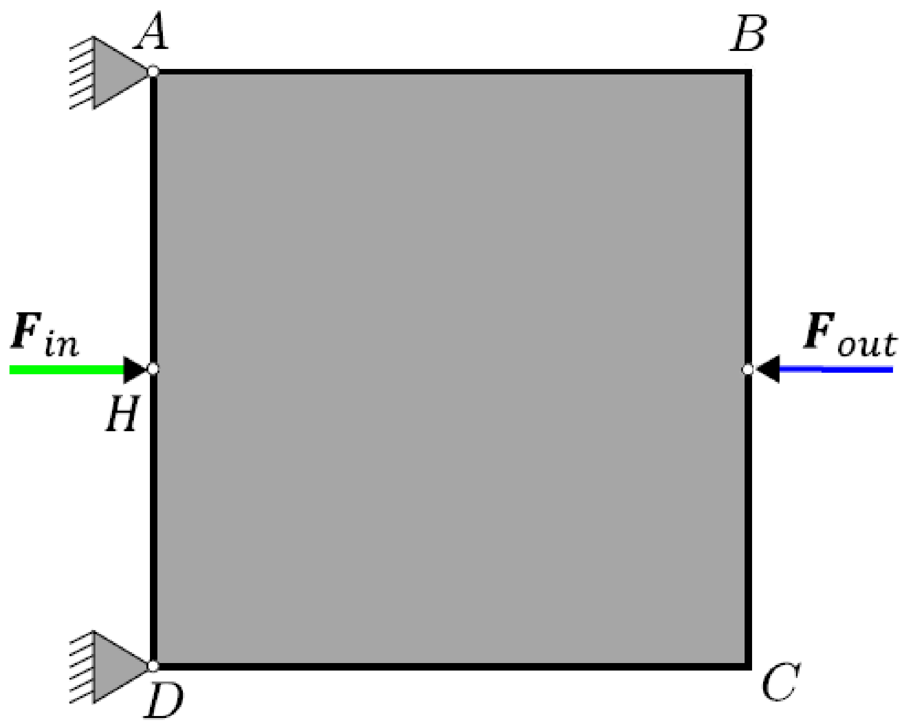
$$x_{n+1} = x_n - \gamma [Hf(x_n)]^{-1} \nabla f(x_n), n \geq 0 .$$

Αυτό γίνεται για να εξασφαλισθεί ότι ισχύουν οι συνθήκες Wolfe, για κάθε βήμα $x_n \rightarrow x_{n+1}$ της επανάληψης.

Ο πρώτος αλγόριθμος Quasi-Newton προτάθηκε από τον Davidon W. C. Το 1959, η DFP ενημερωτική φόρμουλα (DFP updating formula). Σήμερα οι γνωστότεροι και συχνότερα χρησιμοποιούμενοι αλγόριθμοι Quasi-Newton είναι η φόρμουλα SR1 και η μέθοδος BFGS.

5.ΜΗΧΑΝΙΣΜΟΣ ΜΙΑΣ ΦΟΡΤΙΣΗΣ

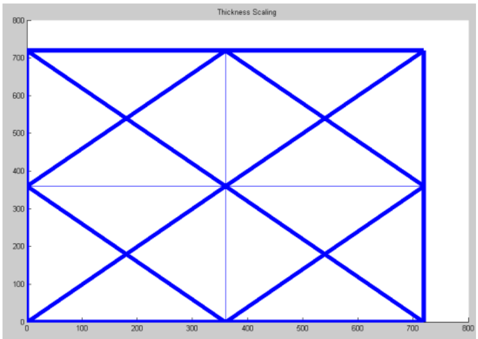
Το πρόβλημα είναι η εύρεση των διατομών των ράβδων που αποτελούν τον εύκαμπτο μηχανισμό ώστε η αντιστροφή της δύναμης να είναι μέγιστη. Στο παρακάτω σχήμα φαίνεται το τετράπλευρο σχεδιασμού καθώς και οι στηρίξεις, το σημείο της εισερχόμενης δύναμης και το σημείο που θέλουμε μέγιστη αντιστροφή.



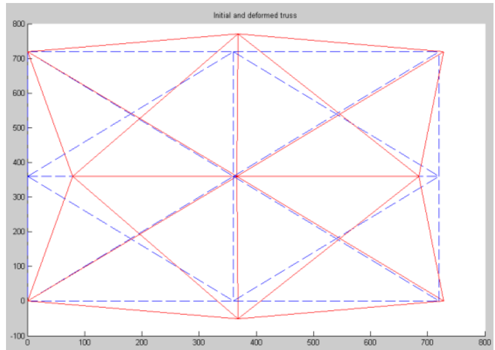
Σχήμα 6: *Τετράπλευρο χωρίο σχεδίασης εύκαμπτου μηχανισμού*

Το πρόβλημα επιλύθηκε για διαφορετικές διαμερίσεις των πλευρών του τετραπλεύρου, συγκεκριμένα για 2Χ2 , 4Χ4, 6Χ6, 8Χ8, 10Χ10, 12Χ12. Παρακάτω παρουσιάζονται τα αποτελέσματα του κώδικα για τις διάφορες διαμερίσεις.

5.1 Διαμέριση 2X2



Σχήμα 7: *Βελτιστοποιημένες διατομές ράβδων*

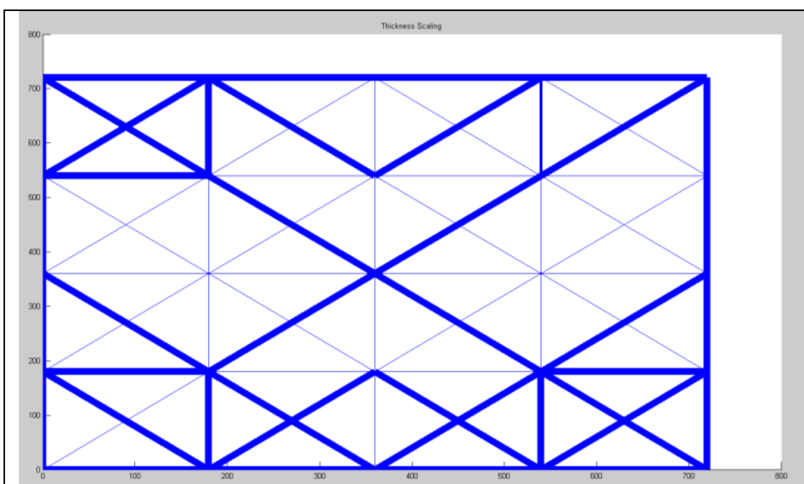


Σχήμα 8: *Μετατοπίσεις κόμβων πριν και μετά την φόρτιση*

Διατομές ράβδων			
1	1	1	1
1	1	1	1
1	1	0.1	1
1	1	0.1	1
0.1	0.1	1	1
Γεωμετρικό Πλεονέκτημα			**45,2%**

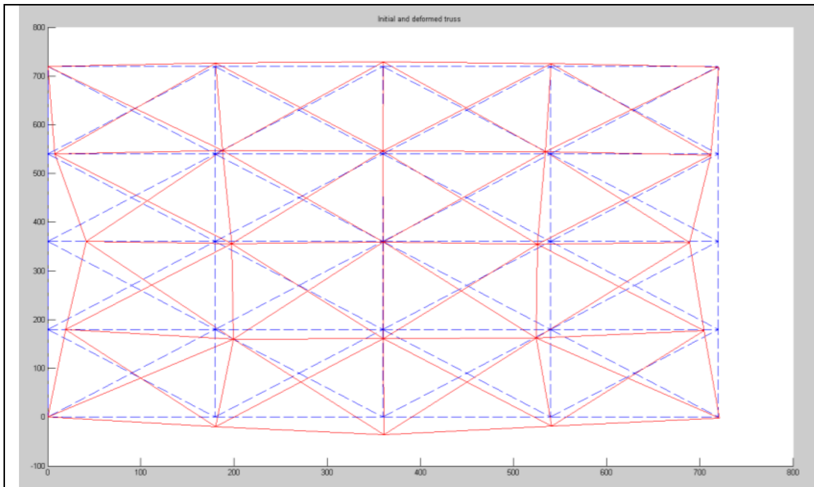
Πίνακας 1: *Βελτιστοποιημένες διατομές ράβδων εύκαμπτου μηχανισμού*

5.2 Διαμέριση 4X4



Σχήμα 9: *Βελτιστοποιημένες διατομές ράβδων*

Διατομές ράβδων		
10	0.1	10
10	0.1	10
10	10	10
0.1	0.1	10
10	0.1	10
10	10	0.1
10	0.1	10

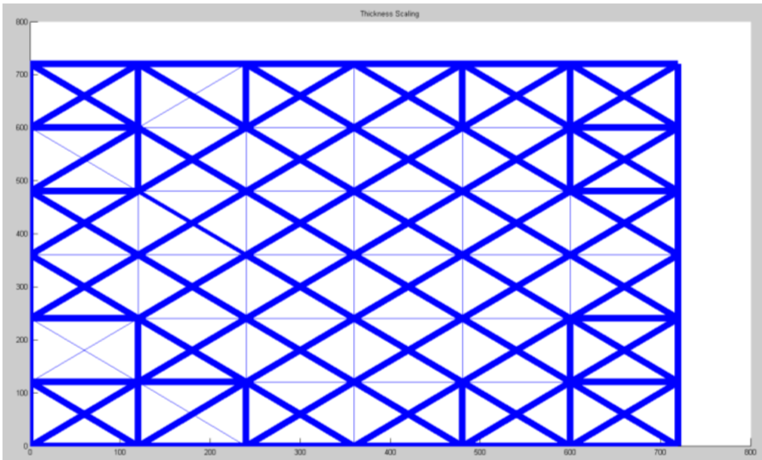


Σχήμα 10: *Μετατοπίσεις κόμβων πριν και μετά την φόρτιση*

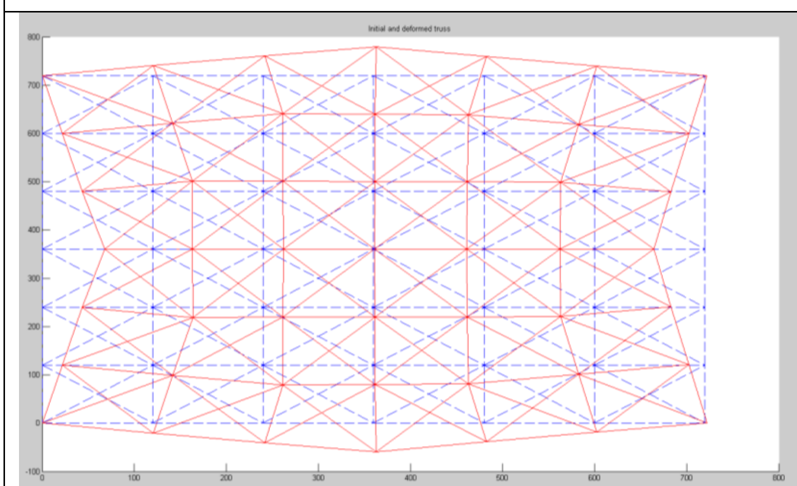
10	10	0.1
0.1	10	0.1
10	0.1	0.1
3.400745	0.1	0.1
10	0.1	3.996253
10	0.1	0.809562
10	0.1	0.1
10	10	0.1
10	0.1	10
10	0.1	10
10	0.1	10
10	0.1	10
0.1	10	10
0.1	0.1	10
0.1	0.1	10
0.1	0.1	10
10	0.1	10
Γεωμ. Πλεονέκτημα		**74,65%**

Πίνακας 2: *Βελτιστοποιημένες διατομές ράβδων*

5.3 Διαμέριση 6Χ6



Σχήμα 11: *Βελτιστοποιημένες διατομές ράβδων*

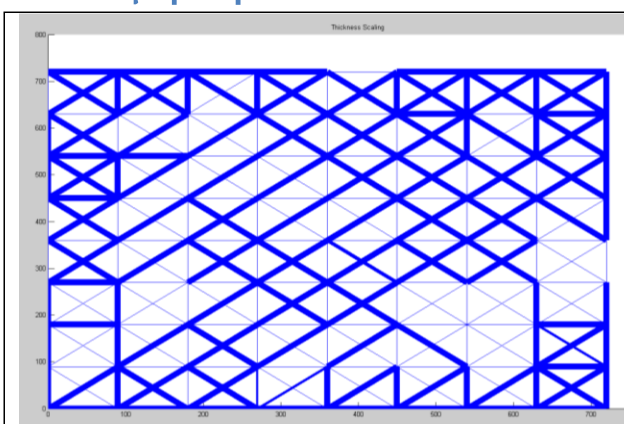


Σχήμα 12: *Μετατοπίσεις κόμβων πριν και μετά την φόρτιση*

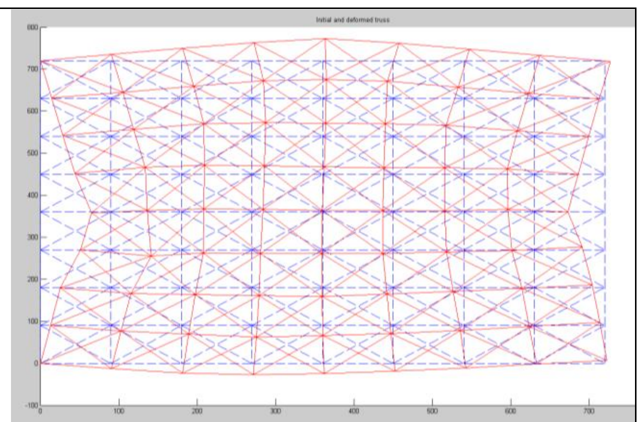
Διατομές ράβδων		
9.620198	0.1	0.1
10	0.1	0.1
9.986952	10	10
10	3.659076	10
10	0.1	0.1
10	0.1	0.1
0.111911	3.374858	10
10	10	7.783397
10	0.1	0.1
10	0.1	0.1
10	10	10
10	10	10
0.1	0.1	10
5.475084	0.1	10
10	10	10
10	10	10
10	0.1	7.904963
10	10	7.174557
10	10	10
6.189144	10	10
10	10	10
10	0.1	0.1
10	10	10
10	10	0.1
9.639068	0.1	10
5.412505	0.1	0.1
0.1	1.162794	10
0.1	10	10
9.397719	0.1	0.1
10	0.1	0.1
10	10	10
10	9.97002	10
0.1	0.1	10
0.1	0.1	0.1
10	10	10
10	10	10
0.1	0.1	10

0.1	0.1	10
1.860524	10	10
10	10	10
0.1	0.1	10
0.1	0.1	10
10	10	10
10	10	10
10	9.14045	10
9.660928	10	10
10	0.1	10
10	8.849686	10
10	10	10
10	0.1	6.40976
10	10	10
10	10	10
Γεωμ.Πλεονέκτημα		**81,47%**

5.4 Διαμέριση 8X8

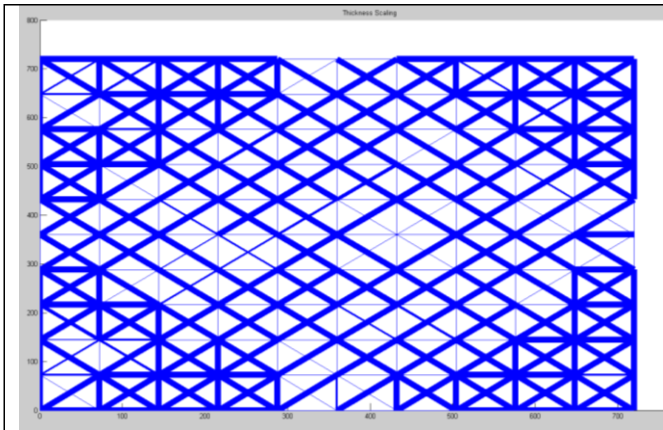


Σχήμα 13: *Βελτιστοποιημένες διατομές ράβδων*

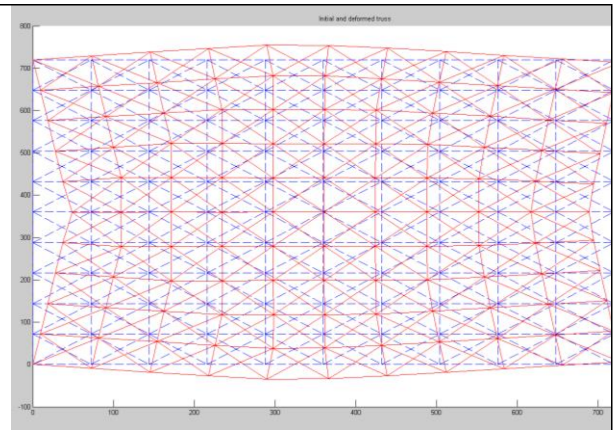


Σχήμα 14: *Μετατοπίσεις κόμβων πριν και μετά την φόρτιση*

5.5 Διαμέριση 10X10



Σχήμα 15: *Βελτιστοποιημένες διατομές ράβδων*

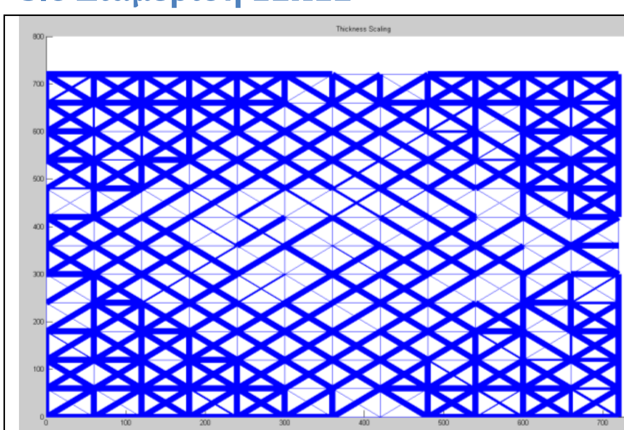


Σχήμα 16: *Μετατοπίσεις κόμβων πριν και μετά την φόρτιση*

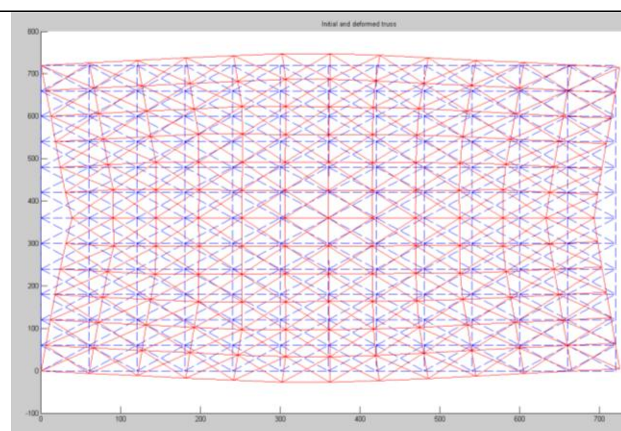
Διατομές Ράβδων								
3.983636	9.803305	0.1	9.412148	0.1	0.1	0.1	9.858518	9.693361
10	5.14988	0.1	9.619301	0.1	0.1	0.1	9.770253	1.654273
0.103418	10	0.935965	9.904492	0.239999	10	9.458643	10	9.923419
10	10	10	9.260381	9.950237	1.088013	1.839374	0.624668	3.353065
10	0.1	0.1	0.151776	0.1	0.1	9.940564	9.79925	
10	0.1	0.1	9.881365	0.1	0.1	5.178137	10	
3.060416	0.1	9.443111	10	1.978243	9.830578	9.433063	9.878731	
9.899577	9.908294	1.442668	6.579004	5.165058	3.919997	9.670486	8.482517	
10	0.1	0.1	0.1	0.1	0.1	6.602725	10	
10	0.1	2.552499	0.1	0.1	0.1	2.191664	9.279601	
9.951825	9.837595	7.081462	9.941604	6.823158	10	0.1	10	
10	10	9.509903	7.599985	0.70218	2.702107	8.681285	9.965161	
10	0.1	9.888974	0.1	0.1	0.1	0.1	10	
10	0.1	3.250015	0.1	0.1	0.1	0.232851	0.1	
10	1.768283	9.736639	9.895629	0.1	0.516178	9.851095	10	
10	9.420607	9.992405	0.154044	4.865033	9.558663	5.026227	0.1	
10	0.1	7.100065	0.1	0.1	0.1	7.613118	0.1	
0.1	7.858839	7.583982	0.1	0.1	0.1	0.1	0.1	
0.1	2.541718	7.90233	0.222529	9.959222	10	9.913467	9.983378	
10	9.789717	9.836201	0.390667	0.1	9.421393	10	8.476402	
0.473071	9.505258	10	0.1	0.1	9.313828	9.998135	0.1	
0.1	9.516462	9.879141	0.1	0.1	5.569102	0.1	0.1	
0.103352	5.766404	10	3.895224	2.878374	8.191547	10	5.984428	
9.991001	1.767403	0.1	3.909116	9.966852	1.614986	10	6.813574	
6.807743	9.330165	0.1	0.1	0.1	9.093148	0.1	8.226448	
0.1	0.328585	0.1	0.1	0.1	6.804113	0.1	3.30662	
10	9.750394	9.612023	8.921837	0.1	5.988353	9.960738	0.218659	

6.558573	9.865462	0.271756	0.129654	2.923966	9.733553	3.283454	4.270582	
9.592619	7.856304	0.1	0.1	0.1	8.10869	0.1	7.501942	
9.736891	0.183505	0.1	0.1	9.491752	0.1	0.1	8.491129	
5.752071	1.207603	10	0.1	0.1	9.933037	10	6.24612	
1.240153	9.700268	5.724881	9.998319	10	10	8.065624	5.103155	
9.981134	2.537797	0.1	0.1	4.479467	0.1	0.1	5.852394	
9.855716	0.301028	0.1	0.1	9.070962	0.1	0.1	1.890107	
6.267702	10	9.141226	9.80952	6.622902	3.827492	5.2199	8.992217	
9.990986	1.62982	2.248963	2.597631	7.762614	1.090593	3.649452	4.913984	
6.799579	6.068478	0.1	0.1	10	0.1	0.1	4.026557	
4.599343	0.1	0.1	0.1	0.1	0.1	0.1	10	
6.799301	10	2.981102	1.944501	0.1	10	9.147276	10	
9.835042	5.116854	10	5.564245	10	8.193491	2.206172	10	
3.397518	0.1	0.1	0.1	0.1	0.1	7.769639	0.1	
0.207716	0.1	0.1	9.98878	0.1	0.1	4.620506	0.1	
9.430656	9.806907	9.355409	9.945014	4.769655	10	2.259271	10	
0.240252	3.550037	9.920005	0.1	8.490562	5.26899	0.834421	10	
0.1	0.1	0.1	0.1	0.1	0.1	9.563809	9.590701	
9.983302	0.1	0.1	0.1	0.1	0.1	2.707529	3.350997	
0.664839	9.644951	2.291059	9.580837	10	10	9.973977	10	
6.723962	9.958358	10	10	2.09587	0.142842	9.287734	10	
9.988936	0.1	0.1	0.1	0.1	0.1	5.084094	10	
10	0.1	0.1	0.1	0.1	0.1	0.381295	10	
10	0.511292	10	9.799585	4.505173	1.769279	10	0.1	
9.920322	10	0.1	7.232843	5.292061	4.50895	0.38069	0.1	
Γεωμετρικό Πλεονέκτημα							**63,26%**	

5.6 Διαμέριση 12X12



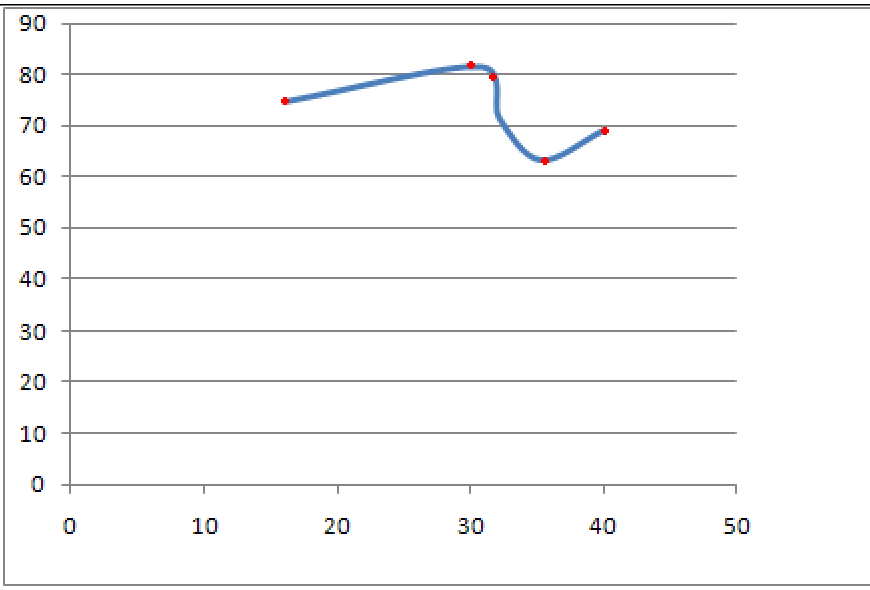
Σχήμα 17: *Βελτιστοποιημένες διατομές ράβδων*



Σχήμα 18: *Μετατοπίσεις κόμβων πριν και μετά την φόρτιση*

5.7 Συγκριτικά αποτελέσματα

Για να συγκρίνουμε τις διάφορες περιπτώσεις χρησιμοποιήσαμε τον λόγο του όγκου των ράβδων προς το εμβαδό του παραλληλόγραμμου. Έτσι κατασκευάσαμε το παρακάτω γράφημα στον κάθετο άξονα είναι το γεωμετρικό πλεονέκτημα και στον οριζόντιο το ποσοστό κάλυψης του παραλληλόγραμμου.



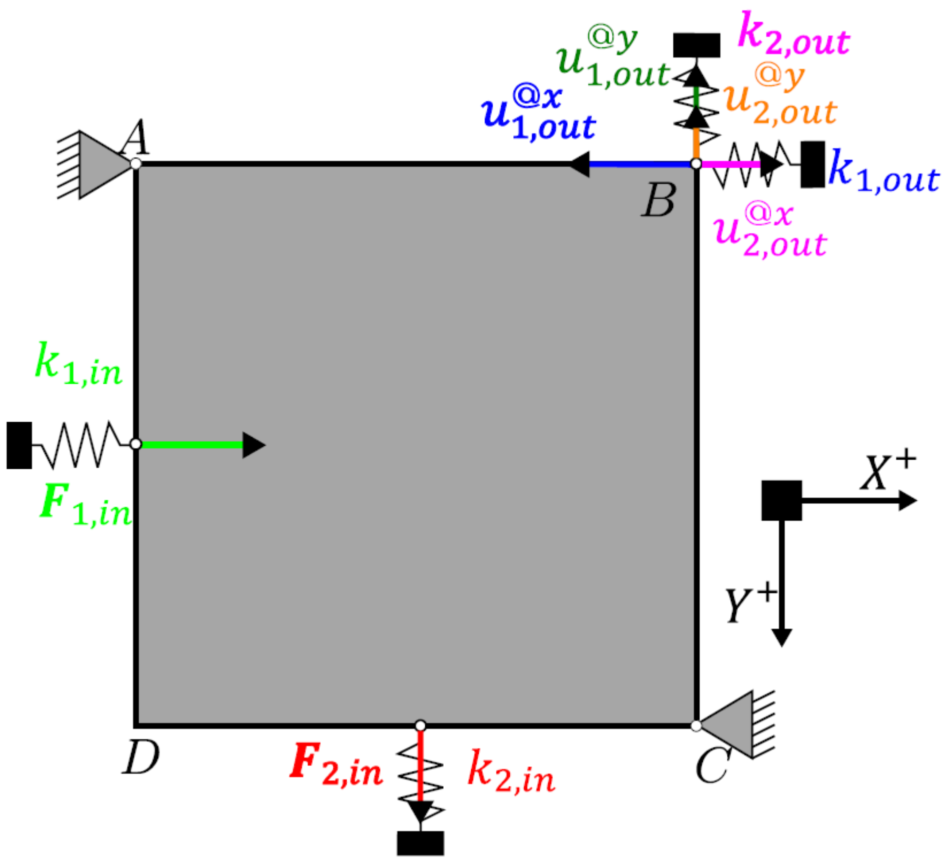
Σχήμα 19: *Συγκριτικό γράφημα ποσοστό κάλυψης-γεωμετρικό πλεονέκτημα*

4x4	16.01	74.65
6x6	30.77	81.47
8x8	32.14	71.59
10x10	35.3	63.26
12x12	40.05	69.12

Παρατηρούμε ότι καθώς πυκνώνει το πλέγμα δεν έχουμε βελτίωση του γεωμετρικού πλεονεκτήματος μετά την περίπτωση 6Χ6. Ακόμη παρατηρούμε ότι στην περίπτωση 12Χ12 έχουμε καλύτερο αποτέλεσμα από ότι στην περίπτωση 10Χ10.

6.ΠΟΛΥΜΟΡΦΙΚΟΣ ΜΗΧΑΝΙΣΜΟΣ ΔΥΟ ΦΟΡΤΙΣΕΩΝ

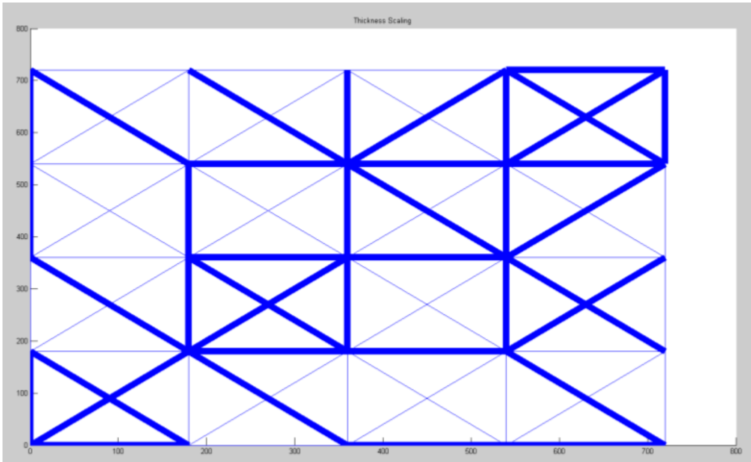
Το πρόβλημα είναι η εύρεση των διατομών του εύκαμπτου μηχανισμού, ώστε για δύο διαφορετικές φορτίσεις ο ίδιος μηχανισμός να λειτουργεί διαφορετικά. Στο παρακάτω σχήμα φαίνεται το τετράπλευρο σχεδιασμού καθώς και οι στηρίξεις, τα σημεία εφαρμογής των δυνάμεων και τα σημεία στα οποία θέλουμε μέγιστη αντιστροφή της κάθε δύναμης.



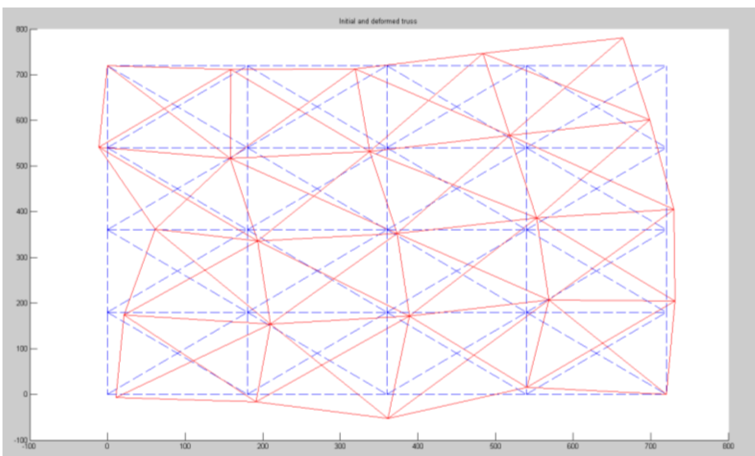
Σχήμα 20: *Τετράπλευρο χωρίο σχεδιασμού εύκαμπτου μηχανισμού*

Το πρόβλημα σε αυτή την περίπτωση έχει δύο στόχους, όταν ασκείται η δύναμη F1 θέλουμε η μετατόπιση u1x να είναι μέγιστη ενώ όταν ασκείται η δύναμη F2 θέλουμε η μετατόπιση u2y να είναι μέγιστη. Για την σύνθεση των δύο στόχων σε ένα θεωρήσαμε βάρη στους δύο στόχους με συντελεστή στάθμισης a=0.5. Ακολουθούν γραφικά αποτελέσματα καθώς και οι διατομές των ράβδων για τις διάφορες διαμερίσεις των πλευρών του τετραπλεύρου.

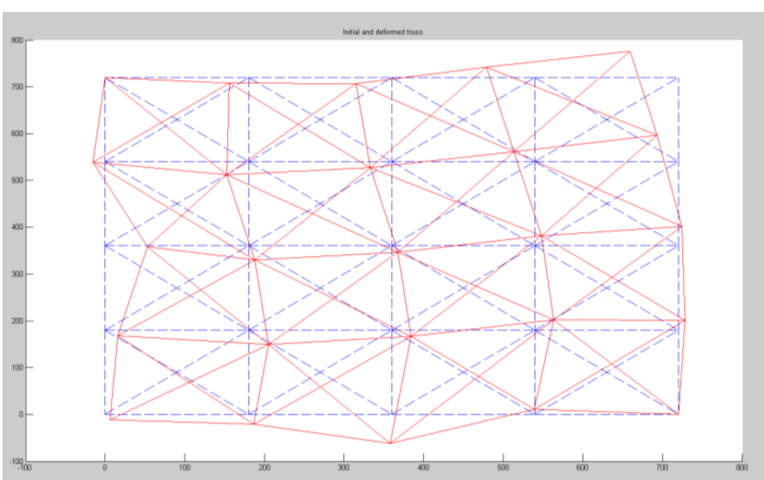
6.1 Διαμέριση 4Χ4



Σχήμα 21: *Βελτιστοποιημένες διατομές ράβδων*

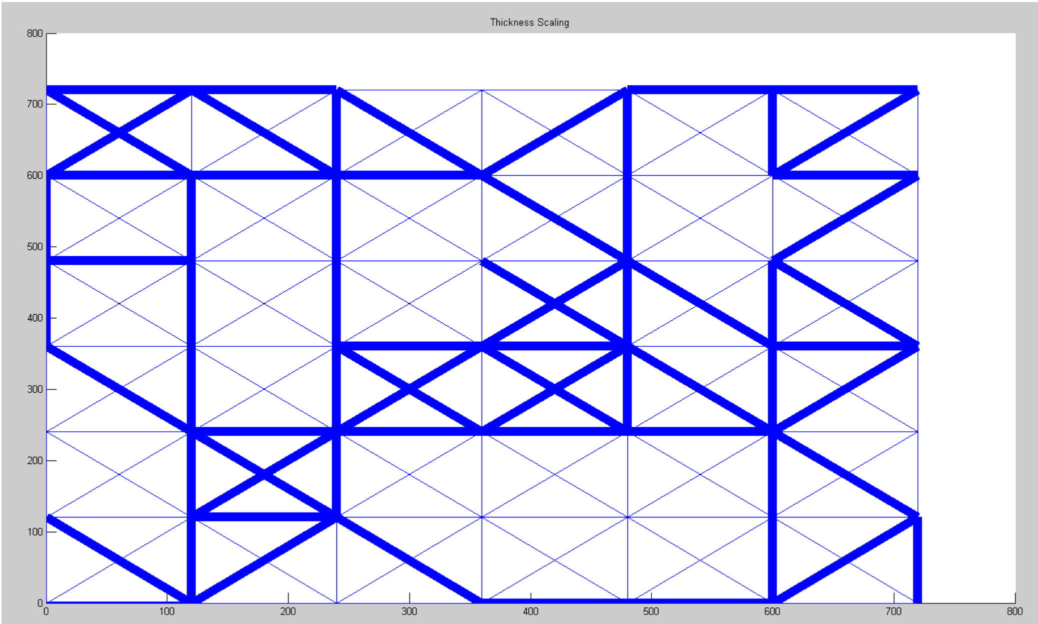


Σχήμα 22: *Φόρτιση στον x άξονα γεωμετρικό πλεονέκτημα 72,12%*

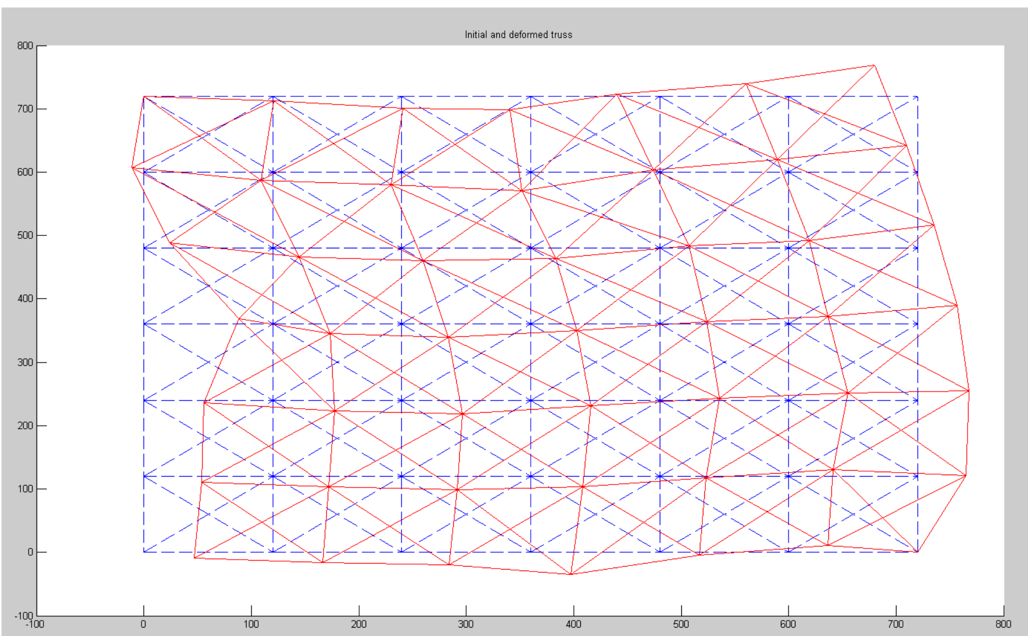


Σχήμα 23: *Φόρτιση στον y άξονα γεωμετρικό πλεονέκτημα 71,15%*

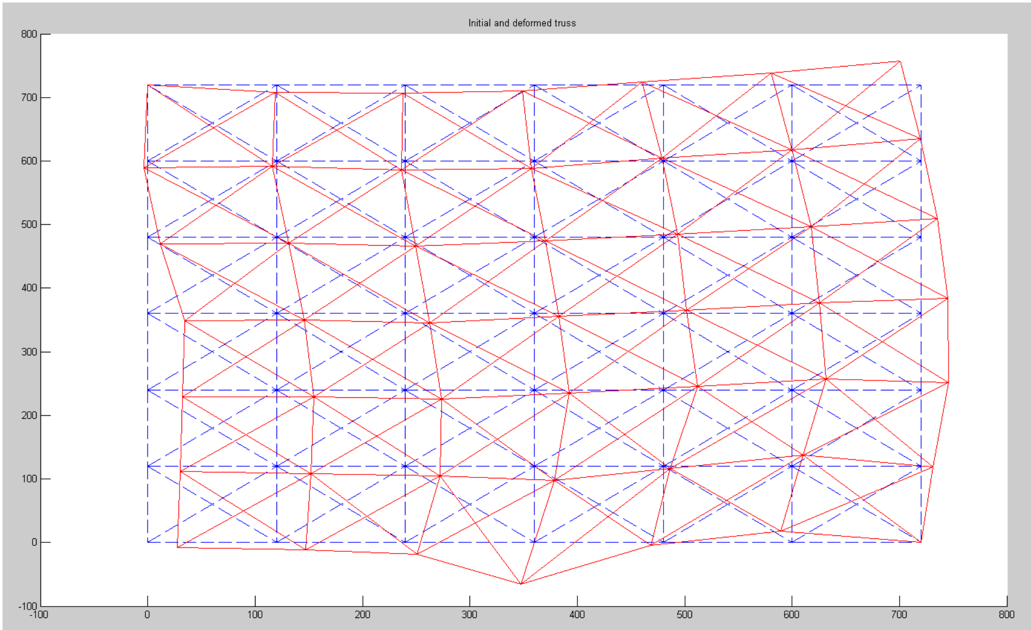
6.2 Διαμέριση 6Χ6



Σχήμα 24: *Βελτιστοποιημένες διατομές ράβδων*

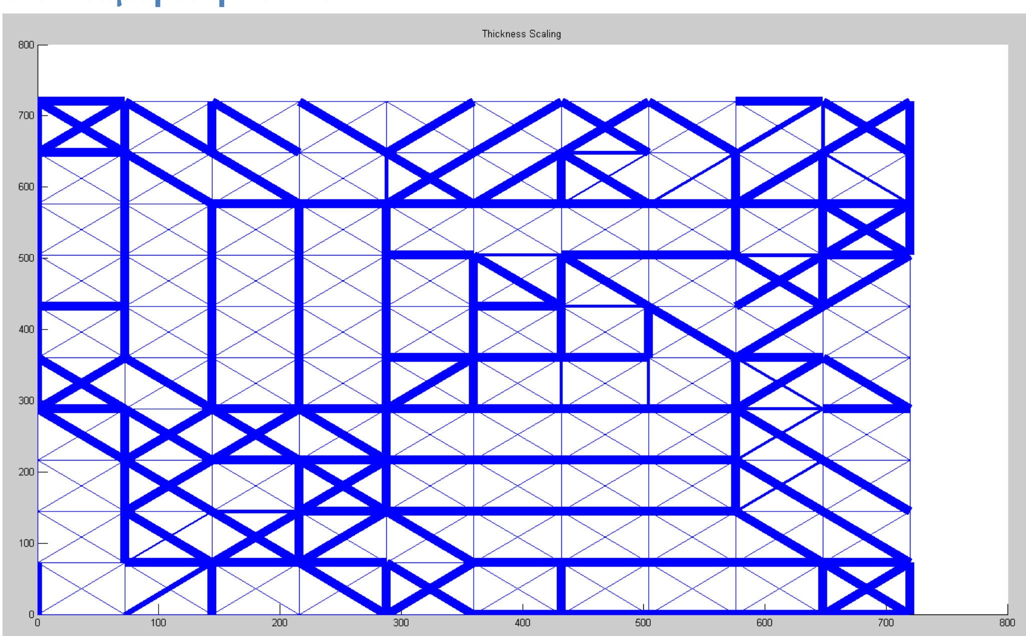


Σχήμα 25: *Φόρτιση στον x άξονα γεωμετρικό πλεονέκτημα 44,97%*

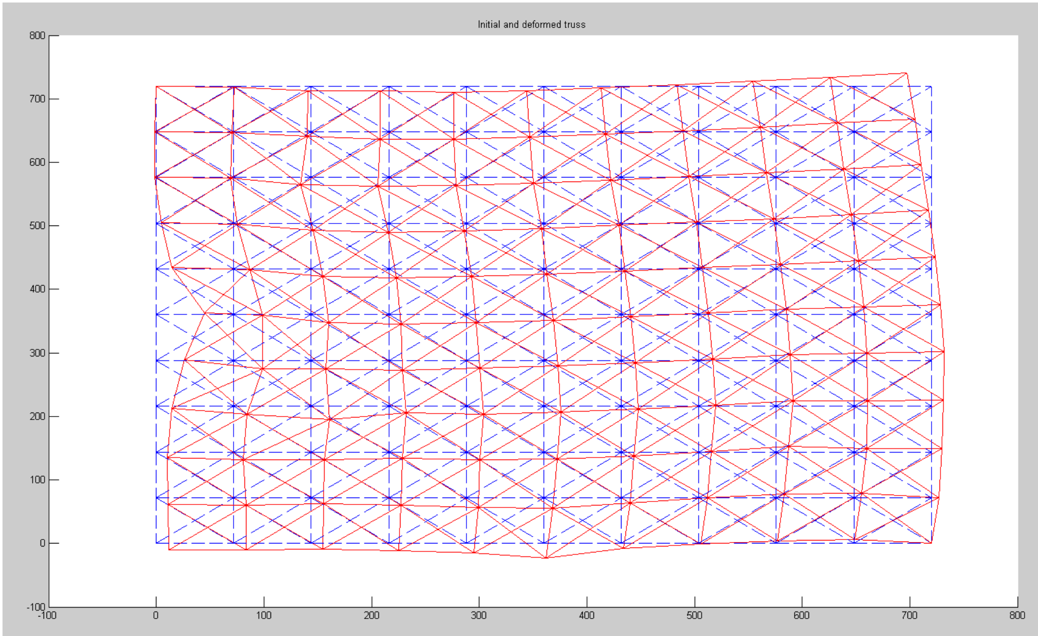


Σχήμα 26: *Φόρτιση στον y άξονα γεωμετρικό πλεονέκτημα 56,44%*

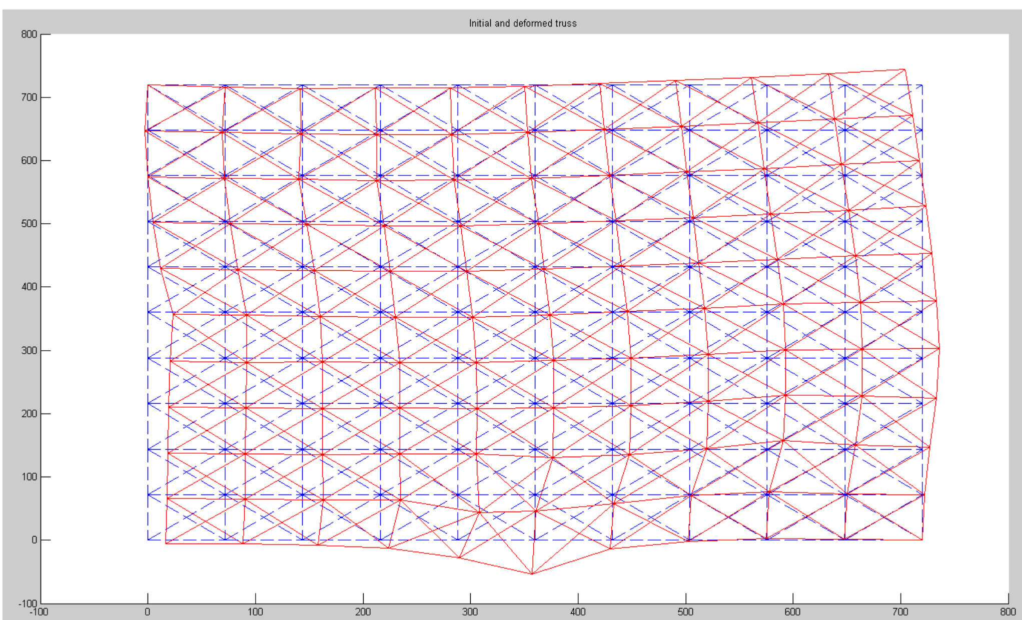
6.3 Διαμέριση 10X10



Σχήμα 27: *Βελτιστοποιημένες διατομές ράβδων*



Σχήμα 28: *Φόρτιση στον x άξονα γεωμετρικό πλεονέκτημα 49,63%*



Σχήμα 29: *Φόρτιση στον y άξονα γεωμετρικό πλεονέκτημα 44,58%*

7 Υπολογιστικό κόστος

Στους παρακάτω πίνακες φαίνεται ο χρόνος εκτέλεσης για τις διάφορες περιπτώσεις διαμερίσεων.

Μηχανισμός μίας φόρτισης		
Διακριτοποίηση	Αριθμός στοιχείων	Χρόνος Εκτέλεσης
4x4	72	1min 16sec
6x6	156	8min 31sec
8x8	272	14min 39sec
10x10	420	49min 49sec
12x12	600	72min 28sec

Πίνακας 6: *Χρόνοι εκτέλεσης για τις διάφορες διακριτοποιήσεις*

Μηχανισμός δύο φορτίσεων		
Διακριτοποίηση	Αριθμός στοιχείων	Χρόνος Εκτέλεσης
4x4	72	1min 34sec
6x6	156	4min 25sec
10x10	420	50min 6sec

Πίνακας 7: *Χρόνοι εκτέλεσης για τις διάφορες διακριτοποιήσεις*

Παράρτημα

Κώδικας επίλυσης μετακινήσεων για συγκεκριμένο διάνυσμα διατομών των ράβδων.

```
function [func,Utotal] =ex1trus2d (A)


%% example of truss2d element

% solving a two-dimensional truss

%

% Copyright: G.E. Stavroulakis (2004)

%

%% number of nodes


%% truss2d element

% solving a two-dimensional truss

% input: parameters related to the cross-sections of bars

% output: one of the cost functions


%% automatic preparation

% length and number of segments in x direction

lx = 720; nx = 4;

% length and number of segments in y direction

ly = 720; ny = 4;

%%



%% number of nodes

nnodes = (nx+1)*(ny+1);

%% nodes

xnode=zeros(nnodes,1);

ynode=zeros(nnodes,1);

iel=0;

for i = 1:nx+1

   for j = 1:ny+1

      iel=(j-1)*(nx+1)+i;

      xnode(iel)=(lx/nx)*(i-1);

      ynode(iel)=(ly/ny)*(j-1);
```

```
    end
end
%% number of elements
nelements = (nx*ny)*4+ny+nx;
%% connectivity of elements
cnct = zeros(nelements,2);
jel=0;
for ii=1:nx
    for jj=1:ny
        jel=((jj-1)*nx+(ii-1))*4;
         cnct(jel+1,1) = jj*(nx+1)+ii;
         cnct(jel+1,2) = (jj-1)*(nx+1)+ii;
         cnct(jel+2,1) = (jj-1)*(nx+1)+ii;
         cnct(jel+2,2) = (jj-1)*(nx+1)+ii+1;
         cnct(jel+3,1) = jj*(nx+1)+ii;
         cnct(jel+3,2) = (jj-1)*(nx+1)+ii+1;
         cnct(jel+4,1) = (jj-1)*(nx+1)+ii;
         cnct(jel+4,2) = jj*(nx+1)+ii+1;

    end
end
jel=nx*ny*4;
for jj = 1:ny
        jel= jel+1;
         cnct(jel,1) = (nx+1)*jj;
         cnct(jel,2) = (nx+1)*(jj+1);
end

jel=nx*ny*4+ny;
for ii = 1:nx
       jel=jel+1;
         cnct(jel,1) = ny*(nx+1)+ii;
         cnct(jel,2) = ny*(nx+1)+ii+1;
```

```
end

%% material constants

E = 10000;


%A=ones(nelements,1);

rh=0.1;

%% loading

%number of loads

nuload = 1;

% number of node, x - y loadings

loads = zeros(nuload,3);

loads(1,1) = 11;

loads(1,2) = 600;

loads(1,3) = 0;

%% boundary conditions

% number of boundary conditions

nubcs = 2;

% number of node, x - y displs (code 1 = 0, code 0 = free)

bcs = zeros(nubcs,3);

bcs(1,1) = 1;

bcs(1,2) = 1;

bcs(1,3) = 1;

 bcs(2,1) = 21;

 bcs(2,2) = 1;

 bcs(2,3) = 1;



%

%% preparation

% stiffnes matrix

Ktotal = zeros(2*nnodes,2*nnodes);

% loading vector

Ftotal = zeros(2*nnodes,1);
```

```
% space for the solution

Utotal = zeros(2*nnodes,1);

% space for local stiffness matrix

Kelm = zeros(4,4);

%

% assemply of the stiffness matrix

for i=1:nelements

   % for element i

   % first node number cnct(i,1)

   % first node coordinates xnode(cnct(i,1)), ynode(cnct(i,1))

   P1 = [ xnode(cnct(i,1)), ynode(cnct(i,1)) ];

   P2 = [ xnode(cnct(i,2)), ynode(cnct(i,2)) ];

   % local 4x4 stiffness matrix

   Kelm = truss2d(A(i,1),E,P1,P2);

   % assemply in global Ktotal stiffness matrix

   Ktotal((cnct(i,1)-1)*2+1:(cnct(i,1)-1)*2+2,(cnct(i,1)-1)*2+1:(cnct(i,1)-1)*2+2)= ...

   Ktotal((cnct(i,1)-1)*2+1:(cnct(i,1)-1)*2+2,(cnct(i,1)-1)*2+1:(cnct(i,1)-1)*2+2)+Kelm(1:2,1:2);

   Ktotal((cnct(i,1)-1)*2+1:(cnct(i,1)-1)*2+2,(cnct(i,2)-1)*2+1:(cnct(i,2)-1)*2+2)= ...

   Ktotal((cnct(i,1)-1)*2+1:(cnct(i,1)-1)*2+2,(cnct(i,2)-1)*2+1:(cnct(i,2)-1)*2+2)+Kelm(1:2,3:4);

   Ktotal((cnct(i,2)-1)*2+1:(cnct(i,2)-1)*2+2,(cnct(i,1)-1)*2+1:(cnct(i,1)-1)*2+2)= ...

   Ktotal((cnct(i,2)-1)*2+1:(cnct(i,2)-1)*2+2,(cnct(i,1)-1)*2+1:(cnct(i,1)-1)*2+2)+Kelm(3:4,1:2);

   Ktotal((cnct(i,2)-1)*2+1:(cnct(i,2)-1)*2+2,(cnct(i,2)-1)*2+1:(cnct(i,2)-1)*2+2)= ...

   Ktotal((cnct(i,2)-1)*2+1:(cnct(i,2)-1)*2+2,(cnct(i,2)-1)*2+1:(cnct(i,2)-1)*2+2)+Kelm(3:4,3:4);

end

%

% loading vector

%

for i=1:nuload

   % load at node loads(i,1) with x-y contribution equal to loads(i,2), loads(i,3)

   Ftotal(2*(loads(i,1)-1)+1)=loads(i,2);

   Ftotal(2*(loads(i,1)-1)+2)=loads(i,3);

end

%% imposing the boundary conditions
```

```
for i=1:nubcs
   % for node bcs(i,1) check x-y supports
   % if bcs(i,2)=1 then x displacement is fixed equal to zero
   % if bcs(i,3)=1 then y displacement is fixed equal to zero
   if bcs(i,2)==1
       Ktotal(2*(bcs(i,1)-1)+1,:)=0;
       Ktotal(:,2*(bcs(i,1)-1)+1)=0;
       Ktotal(2*(bcs(i,1)-1)+1,2*(bcs(i,1)-1)+1)=1;
   end
   if bcs(i,3)==1
       Ktotal(2*(bcs(i,1)-1)+2,:)=0;
       Ktotal(:,2*(bcs(i,1)-1)+2)=0;
       Ktotal(2*(bcs(i,1)-1)+2,2*(bcs(i,1)-1)+2)=1;
   end
end
%
%% solving the system of equations
Utotal=Ktotal\Ftotal;
%%
```

Κώδικας γραφικών αποτελεσμάτων

Μετατοπίσεις στους κόμβους

```
newplot
hold on
for i=1:nelements
x = [xnode(cnct(i,1)) xnode(cnct(i,2)) ];
y=[ynode(cnct(i,1)) ynode(cnct(i,2)) ];
  plot(x,y,'b--')
%    displacements of nodes
   xdispl = [Utotal(2*(cnct(i,1)-1)+1) Utotal(2*(cnct(i,2)-1)+1) ];
   ydispl = [Utotal(2*(cnct(i,1)-1)+2) Utotal(2*(cnct(i,2)-1)+2) ];
   plot(x+xdispl,y+ydispl,'r')
end
```

```
title('Initial and deformed truss')
hold off
```

Γραφική παράσταση διατομών των ράβδων

```
newplot
 hold on
 for i=1:nelements;
    x = [xnode(cnct(i,1)) xnode(cnct(i,2)) ];
 y = [ynode(cnct(i,1)) ynode(cnct(i,2)) ];

 Alpha_max = 1.5;
 Alpha_min = 0.1;
 L= Alpha_max-Alpha_min;
 Thickness_matrix =[0.5 2 3 4 6 9];

 if A(i)<=Alpha_min+L/32
   Thickness = Thickness_matrix(1,1);
 elseif A(i)<=Alpha_min+2*L/30
   Thickness = Thickness_matrix(1,2);
 elseif A(i)<=Alpha_min+2*L/6
   Thickness = Thickness_matrix(1,3);
 elseif A(i)<=Alpha_min+3*L/5
   Thickness = Thickness_matrix(1,4);
 elseif A(i)<=Alpha_min+4*L/5
   Thickness = Thickness_matrix(1,5);
   else
   Thickness = Thickness_matrix(1,6);
 end
 plot(x,y,'LineWidth',Thickness)
 end

 title('Thickness Scaling')
 hold off
```

Βιβλιογραφία

Chandrupatla, Tirupathi R. and Belegundu, Ashok D., "*Introduction to Finite Elements in Engineering*", 3rd edition , Prentice Hall, 2002

Ν. Καμινάκης , "*Σύνθεση Πολυμορφικών Εύκαμπτων Μηχανισμών με χρήση Τοπολογικής Βελτιστοποίησης και Εξελικτικών Αλγορίθμων*", Διατριβή Μ.Δ.Ε., Πολυτεχνείο Κρήτης, 2008

Ν. Σκάρος, "*Βέλτιστος Σχεδιασμός Δικτυωτών Φορέων με την βοήθεια της Μεθόδου των Πεπερασμένων Στοιχείων και Αλγορίθμων Βελτιστοποίησης*", Διατριβή Μ.Δ.Ε., Πολυτεχνείο Κρήτης, 2008

Kirsch U., "*Structural Optimization, Fundamentals and Applications*", Springer-Verlag

Mordecai A., "*Nonlinear Programming: Analysis and Methods*", Dover Publishing, 2003