



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

**Μοντέλο παραμόρφωσης δέρματος με μέθοδο πεπερασμένων
στοιχείων**

—

Εφαρμογή σε σύστημα διαδραστικής απεικόνισης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΕΤΡΟΣ Η. ΤΡΙΑΝΤΑΦΥΛΛΟΠΟΥΛΟΣ

Επιβλέπων : Κωνσταντίνος Τζαφέστας
Λέκτορας Ε.Μ.Π.

Αθήνα, Ιούνιος 2005



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

**Μοντέλο παραμόρφωσης δέρματος με μέθοδο πεπερασμένων
στοιχείων**

—

Εφαρμογή σε σύστημα διαδραστικής απεικόνισης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΕΤΡΟΣ Η. ΤΡΙΑΝΤΑΦΥΛΛΟΠΟΥΛΟΣ

Επιβλέπων : Κωνσταντίνος Τζαφέστας
Λέκτορας Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24^η Ιουνίου 2005.

.....
Κωνσταντίνος Τζαφέστας,
Λέκτορας Ε.Μ.Π.

.....
Κωνσταντίνα Νικήτα,
Αν. Καθηγήτρια Ε.Μ.Π.

.....
Σπυρίδων Τζαφέστας,
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2005

.....
ΠΕΤΡΟΣ Η. ΤΡΙΑΝΤΑΦΥΛΛΟΠΟΥΛΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Πέτρος Η. Τριανταφυλλόπουλος, 2005

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στην παρούσα διπλωματική εργασία μελετάται η συμπεριφορά της επιφάνειας του ανθρώπινου δέρματος κατά την εφαρμογή κατακόρυφης σημειακής δύναμης σε αυτό. Για τη μελέτη κατασκευάζεται μοντέλο τμήματος του δέρματος με χρήση πεπερασμένων στοιχείων. Στόχος είναι η μελέτη τεχνικών δυνάμενων να εφαρμοσθούν σε συστήματα διαδραστικής προσομοίωσης της παρακέντησης εν τω βάθει αγγείων του ανθρώπινου σώματος. Στο προσεχές μέλλον σχεδιάζεται η ενσωμάτωση των συστημάτων αυτών σε μια πλατφόρμα ιατρικής εκπαίδευσης μέσω υπολογιστή, για την άσκηση σχετικών με την παρακέντηση κλινικών δεξιοτήτων σε περιβάλλον εικονικής πραγματικότητας με απτική ανάδραση.

Αρχικά, αναλύεται η μέθοδος των πεπερασμένων στοιχείων ώστε να επιλεγεί το κατάλληλο μοντέλο για το πρόβλημά μας. Με την παραδοχή ότι υπάρχει αξονική συμμετρία επιλέγεται να χρησιμοποιηθεί μια δομή δύο διαστάσεων αποτελούμενη από δύο στρώματα διαφορετικής ελαστικότητας (υποθετικά, δέρμα και υποδόριος ιστός). Για τη μοντελοποίηση αυτή αναλύονται οι διαδικασίες που πρέπει να ακολουθηθούν και οι παραδοχές που πρέπει να ληφθούν υπ' όψιν. Επίσης αναλύεται με μαθηματικό τρόπο η προσέγγιση του προβλήματος παραμόρφωσης σε δύο διαστάσεις. Στη φάση αυτή, κατασκευής και επιλογής του βέλτιστου μοντέλου, η προσομοίωση γίνεται εξ' ολοκλήρου σε matlab.

Στη συνέχεια, αφού κατασκευαστεί το μοντέλο σε δύο διαστάσεις πρέπει να προσδιοριστούν οι παράμετροι για το σχετικό πάχος των δύο στρωμάτων και τη σχετική τους ελαστικότητα ώστε το μοντέλο να ανταποκρίνεται στην πραγματικότητα. Για να επιτευχθεί η συγκριτική αυτή αξιολόγηση, χρησιμοποιούνται δύο ομάδες πραγματικών πειραματικών τιμών που περιγράφονται σε σχετικές μελέτες διαθέσιμες στη διεθνή βιβλιογραφία. Δοκιμάζονται συγκριτικά διάφορα ζεύγη παραμέτρων με στόχο να γίνει μια ανάλυση ευαισθησίας του μοντέλου, και να εντοπιστεί αυτός ο συνδυασμός τιμών για τον οποίο υπάρχει η ελάχιστη απόκλιση από τα πειραματικά δεδομένα.

Τέλος, αφού έχει ολοκληρωθεί η μοντελοποίηση σε δύο διαστάσεις ακολουθεί η τρισδιάστατη απεικόνιση με χρήση της αξονικής συμμετρίας. Το τρισδιάστατο μοντέλο υλοποιείται εκ νέου με χρήση γλώσσας C/C++ σε γραφικό περιβάλλον OpenGL, και λειτουργεί σε πραγματικό χρόνο. Ο χρήστης μπορεί να αλλάξει τη θέση του βάθους διεύθυνσης και να δει το αποτέλεσμα σε γραφικό περιβάλλον.

Λέξεις κλειδιά

Πεπερασμένα στοιχεία, ελαστική παραμόρφωση δέρματος, απτικά συστήματα

Abstract

The scope of this thesis is the study of the behavior of the human skin surface at the application of a vertical point force on it. A model of a skin part is developed for the needs of this study, with use of finite elements. The objective of this work is the study of artificial forces when applied on interactive simulations of puncturing the deep vessels of a human body. In the future, there are plans for incorporating these systems in a computer-assisted medical education platform, for training and assessment of clinical dexterities in a virtual reality environment with haptic display.

Initially, the finite element method is analyzed so that a suitable model is selected for our problem. With the hypothesis of axial symmetry, a two dimensional structure that consists of two layers with different elasticity (supposedly, skin and subcutaneous tissue) is selected. The processes and the admissions for this modeling are analyzed and taken into consideration. A mathematical analysis of this two-dimensional deformation problem is performed, for developing and choosing an optimal model.

When the two dimensional model has been created, the parameters for the relative thickness of the two layers and their relative elasticity must be determined, so that the model corresponds to reality. In order to achieve this comparative evaluation, two sets of real experimental values are used. These values are described in relative studies available in the literature. A complete set of parameters pairs are tested in comparison, in order to perform a sensitivity analysis and identify the combination of values that entail the minimal divergence from the experimental data.

Finally, after the two-dimensional modeling has been completed, the three dimensional representation is formed based on the axial symmetry hypothesis. The three dimensional model is created from the beginning using C/C++ programming language in OpenGL graphical environment, and operates in real time. The user can change the place and depth of skin indentation and see the result in a 3D animated graphic environment.

Key words

Finite elements, elastic skin deformation, haptic systems

Περιεχόμενα

Πρόλογος	15
Κεφάλαιο 1: Εισαγωγή – Το πρόβλημα	17
1.1 Εισαγωγικά στοιχεία μοντελοποίησης	17
1.2 Μέθοδοι μοντελοποίησης	19
1.3 Άλλες προσπάθειες μοντελοποίησης με πεπερασμένα στοιχεία	22
1.4 Βιοϊατρικά δεδομένα του προβλήματος	27
1.5 Πλαίσιο εργασίας	30
Κεφάλαιο 2: Μοντελοποίηση ελαστικής παραμόρφωσης με χρήση πεπερασμένων στοιχείων	33
2.1 Βασικά στοιχεία θεωρίας ελαστικής παραμόρφωσης	33
2.2 Εισαγωγή στη μέθοδο των πεπερασμένων στοιχείων	39
2.3 Διακριτοποίηση του προβλήματος	41
2.4 Προβλήματα παραμόρφωσης σε δύο διαστάσεις	46
Κεφάλαιο 3: Εύρεση παραμέτρων (απλοποιημένου) διδιάστατου μοντέλου	51
3.1 Δημιουργία μοντέλου με χρήση των πεπερασμένων στοιχείων	51
3.2 Πειραματικά δεδομένα	57
3.3 Γραμμική παρεμβολή - Μέσο τετραγωνικό σφάλμα	60
3.4 Διόρθωση βάθους διεξόδυσης	62
3.5 Εύρεση παραμέτρων όπου το σφάλμα ελαχιστοποιείται	68
3.6 Εύρεση παραμέτρων σε μεγαλύτερη κατάτμηση	80

Κεφάλαιο 4: Υλοποίηση σε σύστημα διαδραστικής προσομοίωσης	97
4.1 Υλοποίηση σε πραγματικό χρόνο	97
4.2 Δημιουργία τρισδιάστατου μοντέλου από το διδιάστατο	98
4.3 Τρισδιάστατο μοντέλο σε γραφικό περιβάλλον πραγματικού χρόνου	100
4.4 Εφαρμογή σε παρακέντηση	102
Συμπεράσματα – Μελλοντική Εργασία	105
Παράρτημα: Ο κώδικας	109
1 Κώδικας Matlab	109
2 Κώδικας C (εφαρμογή σε OpenGL)	118
Βιβλιογραφία	133

Σχήματα

Σχήμα (1.1)	Μια καμπύλη (cubic) spline. Για να αλλάξει η μορφή της αρκεί να μετατοπίσουμε τα τέσσερα σημεία ελέγχου από τα οποία εξαρτάται.[1]	19
Σχήμα (1.2)	Τμήμα μοντέλου από ελατήρια και μάζες.[1]	20
Σχήμα (1.3)	Μοντέλο δέρματος προσώπου (Terzopoulos and Waters) [1].	20
Σχήμα (1.4)	Εισαγωγή βελόνας σε μαλακό σώμα [2]	22
Σχήμα (1.5)	Το πείραμα: Ένας ρομποτικός βραχίονας εφοδιασμένος με βελόνα, ένα τμήμα ιστού και μία κάμερα [2]	22
Σχήμα (1.6)	Σύγκριση δεδομένων που προκύπτουν από την κάμερα με το μοντέλο πεπερασμένων στοιχείων [2]	23
Σχήμα (1.7)	Εφαρμογή χειρουργικού εργαλείου σε μοντέλο ανθρώπινου οργάνου [3]	24
Σχήμα (1.8)	Υλοποίηση μοντέλου συκωτιού σε τρεις διαστάσεις [3]	24
Σχήμα (1.9)	Υλοποίηση τμήματος ποδιού σε τρεις διαστάσεις [4]	25
Σχήμα (1.10)	Προσομοίωση παραμόρφωσης επιφάνειας γάμπας [4]	25
Σχήμα (1.11)	Ανατομικά στοιχεία του δέρματος [5]	27
Σχήμα (1.12)	Στιβάδες του δέρματος [5]	28
Σχήμα (1.13)	Τομή δέρματος [5]	29
Σχήμα (1.14)	Αναπαράσταση τομής του δέρματος	31
Σχήμα (2.1)	Εφαρμογή δύναμης σε σώμα	33
Σχήμα (2.2)	Διάταση σώματος κατά την εφαρμογή δύναμης σε αυτό	34
Σχήμα (2.3)	Στένεμα σώματος κατά τη διάτασή του [6]	34
Σχήμα (2.4)	Τέντωμα κηρύθρας [6]	35
Σχήμα (2.5)	Σώμα πριν και μετά την παραμόρφωση – Διάνυσμα μετατόπισης σημείου [7]	37
Σχήμα (2.6)	Παράδειγμα σώματος από πεπερασμένα στοιχεία [8]	39
Σχήμα (2.7)	Αλλαγή ακρίβειας με πεπερασμένα στοιχεία [8]	43
Σχήμα (2.8)	Αλλαγή υλικού με πεπερασμένα στοιχεία [8]	43
Σχήμα (2.9)	Μορφή πίνακα K [8]	44
Σχήμα (2.10)	Αρίθμηση κόμβων σε μοντέλο [8]	45
Σχήμα (2.11)	Τριγωνικό πεπερασμένο στοιχείο (βαθμοί ελευθερίας) [8]	46
Σχήμα (2.12)	Τριγωνικό πεπερασμένο στοιχείο (Συντεταγμένες)	48
Σχήμα (3.1)	Μοντέλο με δύο στρώματα διαφορετικού πάχους και ελαστικότητας	51

Σχήμα (3.2)	Ελαστικές ράβδοι (παραμόρφωση μόνο ως προς το μήκος) ...	52
Σχήμα (3.3)	Σχηματική αναπαράσταση ράβδου	52
Σχήμα (3.4)	Σχηματική αναπαράσταση τριγωνικού πεπερασμένου στοιχείου	52
Σχήμα (3.5)	Ελαστικό τετράγωνο από πεπερασμένα στοιχεία	53
Σχήμα (3.6)	Δημιουργία επιφάνειας από ελαστικό υλικό με χρήση πεπερασμένων στοιχείων	53
Σχήμα (3.7)	Ορισμός σταθερών σημείων για το μοντέλο	54
Σχήμα (3.8)	Διδιάστατο μοντέλο πριν την παραμόρφωση	55
Σχήμα (3.9)	Διδιάστατο μοντέλο μεγάλης ακρίβειας	56
Σχήμα (3.10)	Μέτρηση της παραμόρφωσης της επιφάνειας του δέρματος [9]	57
Σχήμα (3.11)	Πειραματικά δεδομένα	59
Σχήμα (3.12)	Καμπύλες σύμφωνα με τα πειραματικά δεδομένα	59
Σχήμα (3.13)	Γραμμική παρεμβολή για τον υπολογισμό του κατακόρυφου σφάλματος	60
Σχήμα (3.14α)	Καμπύλη πρώτης μορφής που προκύπτει από την επιφάνεια της δομής των πεπερασμένων στοιχείων	62
Σχήμα (3.14β)	Καμπύλη δεύτερης μορφής που προκύπτει από την επιφάνεια της δομής των πεπερασμένων στοιχείων	62
Σχήμα (3.15α)	Νέο σημείο αναφοράς σε απόσταση 0,25mm από το σημείο επαφής	63
Σχήμα (3.15β)	Προσδιορισμός σφάλματος της επιφάνειας του μοντέλου ως προς το σημείο αναφοράς	64
Σχήμα (3.16)	Διόρθωση βάθους διείσδυσης	65
Σχήμα (3.17α)	Πριν τη διόρθωση του βάθους διείσδυσης	66
Σχήμα (3.17β)	Λεπτομέρεια	66
Σχήμα (3.18α)	Μετά τη διόρθωση του βάθους διείσδυσης	67
Σχήμα (3.18β)	Λεπτομέρεια	67
Σχήμα (3.19α)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 1/6	69
Σχήμα (3.19β)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 2/6	69
Σχήμα (3.19γ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 3/6	70
Σχήμα (3.19δ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 4/6	70
Σχήμα (3.19ε)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 5/6	71
Σχήμα (3.20α)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 1/6	73
Σχήμα (3.20β)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 2/6	73
Σχήμα (3.20γ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 3/6	74
Σχήμα (3.20δ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 4/6	74
Σχήμα (3.20ε)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 5/6	75
Σχήμα (3.21)	Εσφαλμένη υλοποίηση	76
Σχήμα (3.22α)	Διάγραμμα σφάλματος για λόγο ελαστικότητας 10	77
Σχήμα (3.22β)	Περιοχή αποδεκτών τιμών για λόγο ελαστικότητας 10	78
Σχήμα (3.23α)	Το βέλτιστο μοντέλο (1 ^ο σετ δεδομένων)	79
Σχήμα (3.23β)	Το βέλτιστο μοντέλο (2 ^ο σετ δεδομένων)	79

Σχήμα (3.24α)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 1/12 ...	81
Σχήμα (3.24β)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 2/12 ...	81
Σχήμα (3.24γ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 3/12 ...	82
Σχήμα (3.24δ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 4/12 ...	82
Σχήμα (3.24ε)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 5/12 ...	83
Σχήμα (3.24στ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 6/12 ..	83
Σχήμα (3.24ζ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 7/12 ...	84
Σχήμα (3.24η)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 8/12 ...	84
Σχήμα (3.24θ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 9/12 ...	85
Σχήμα (3.24ι)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 10/12 .	85
Σχήμα (3.24ια)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 11/12 .	86
Σχήμα (3.25α)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 1/12 ...	88
Σχήμα (3.25β)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 2/12 ...	88
Σχήμα (3.25γ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 3/12 ...	89
Σχήμα (3.25δ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 4/12 ...	89
Σχήμα (3.25ε)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 5/12 ...	90
Σχήμα (3.25στ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 6/12 ...	90
Σχήμα (3.25ζ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 7/12 ...	91
Σχήμα (3.25η)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 8/12 ...	91
Σχήμα (3.25θ)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 9/12 ...	92
Σχήμα (3.25ι)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 10/12 .	92
Σχήμα (3.25ια)	Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 11/12 .	93
Σχήμα (3.26)	Διάγραμμα σφάλματος για λόγο ελαστικότητας 10	94
Σχήμα (3.27α)	Το βέλτιστο μοντέλο (1 ^ο σετ δεδομένων)	95
Σχήμα (3.27β)	Το βέλτιστο μοντέλο (2 ^ο σετ δεδομένων)	95
Σχήμα (4.1)	Τρισδιάστατο μοντέλο	98
Σχήμα (4.2)	Παράδειγμα μεταβολής βάθους τυχαίου σημείου της επιφάνειας στο διδιάστατο μοντέλο	99
Σχήμα (4.3)	Προσδιορισμός βάθους τυχαίου σημείου της επιφάνειας σε τρεις διαστάσεις	99
Σχήμα (4.4α)	Τμήμα δέρματος στου οποίου το κέντρο ασκείται σημειακή δύναμη	100
Σχήμα (4.4β)	Τομή για καλύτερη απεικόνιση	100
Σχήμα (4.4γ)	Αναπαράσταση του εσωτερικού με εμφάνιση των κόμβων των πεπερασμένων στοιχείων	101
Σχήμα (4.5)	Προσομοίωση επιφάνειας δέρματος με πεπερασμένα στοιχεία. Προσδιορισμός σημείου επαφής της βελόνας [10] .	102
Σχήμα (4.6)	Παραμόρφωση δέρματος – εικόνα σε ημιδιαφανές μοντέλο [10]	103
Σχήμα (4.7)	Υλοποίηση σε πραγματικό χρόνο – Ανάδραση με χρήση ειδικής συσκευής. [10]	103

Πίνακες

Πίνακας (2.1)	Κατηγορίες πεπερασμένων στοιχείων [8]	42
Πίνακας (2.2)	Συντεταγμένες κόμβων	47
Πίνακας (3.1)	1 ^η ομάδα πειραματικών δεδομένων (set1) για βάθος 2mm	58
Πίνακας (3.2)	2 ^η ομάδα πειραματικών δεδομένων (set2) για βάθος 1mm	58
Πίνακας (3.3)	Μέσο τετραγωνικό σφάλμα για διάφορες τιμές ποσοστού πάχους και συντελεστή ελαστικότητας	68
Πίνακας (3.4)	Μέσο τετραγωνικό σφάλμα για διάφορες τιμές ποσοστού πάχους και συντελεστή ελαστικότητας	72
Πίνακας (3.5)	Μέσο τετραγωνικό σφάλμα για λόγο ελαστικότητας 10	77
Πίνακας (3.6)	Μέσο τετραγωνικό σφάλμα για διάφορες τιμές ποσοστού πάχους και συντελεστή ελαστικότητας	80
Πίνακας (3.7)	Μέσο τετραγωνικό σφάλμα για διάφορες τιμές ποσοστού πάχους και συντελεστή ελαστικότητας	87
Πίνακας (3.8)	Μέσο τετραγωνικό σφάλμα για λόγο ελαστικότητας 10	94

Πρόλογος

Η μεγάλη ανάπτυξη της τεχνολογίας, της πληροφορικής και των εφαρμογών έχει επηρεάσει πολλούς, αν όχι όλους, από τους υπόλοιπους επιστημονικούς κλάδους. Ειδικά τα τελευταία χρόνια οι αλματώδεις εξελίξεις στο χώρο αυτό έχουν συμβάλλει στην αποδοτική και αποτελεσματική επίλυση πολλών προβλημάτων σε όλους σχεδόν τους επιστημονικούς τομείς. Η πληροφορική έχει διεισδύσει σε μεγάλο βαθμό ώστε να αποτελεί πλέον αναπόσπαστο κομμάτι τόσο σε θεωρητικές – πειραματικές όσο και σε πρακτικές – υλοποιήσιμες εφαρμογές. Πλέον κανείς δε μπορεί να φανταστεί χώρους όπως για παράδειγμα η οικονομία, χωρίς να τους συνδυάσει με υπολογιστές. Η χρήση τους δε συντελεί μονάχα στην καταγραφή, αλλά προσφέρει ακρίβεια, αξιοπιστία και ταχύτητα.

Οι εξελίξεις αυτές δε θα μπορούσαν να αφήσουν αδιάφορο το χώρο της ιατρικής. Εκεί, η τεχνολογία έχει προσφέρει εφόδια που ήταν αδιανόητα στο παρελθόν και η δουλειά των ιατρών έχει πάρει άλλες διαστάσεις. Στο χώρο της διάγνωσης υπάρχουν πλέον τεχνικά μέσα, όπως οι ακτινογραφίες, το υπερηχογράφημα, το καρδιογράφημα, που επιτρέπουν την άμεση και απόλυτη πρόγνωση. Επίσης στη θεραπεία έχουν προστεθεί μέσα όπως τα laser, ή τα τελευταία χρόνια η τηλεϊατρική που έχουν αλλάξει ριζικά τις ιατρικές μεθόδους. Παρόλο που όλα τα παραπάνω δε θα ήταν υλοποιήσιμα χωρίς τη χρήση υπολογιστή, ο αμιγής προγραμματισμός δεν έχει γνωρίσει ακόμη πολλές εφαρμογές στην ιατρική. Τα άλματα της πληροφορικής ωστόσο ανοίγουν νέους ορίζοντες και νέα ερωτηματικά, προσφέρουν νέες δυνατότητες και διαφορετικούς τρόπους προσέγγισης. Ίσως να μην απέχουμε πολύ από το πραγματικό πάντρεμα των δύο κλάδων, από τη μέρα όπου ένα πρόγραμμα θα καθορίζει τη συνταγή φαρμάκων, θα προσφέρει πρώτες βοήθειες ή και ακόμη θα είναι υπεύθυνο για τη διεξαγωγή μιας χειρουργικής επέμβασης. Τότε βέβαια θα έχουμε άλλα ερωτήματα, λειτουργικά, ηθικά και νομικά. Ένας ιός στον υπολογιστή για παράδειγμα, ένα προγραμματιστικό σφάλμα, ή ακόμη και μια βλάβη σε ένα ρομποτικό βραχίονα θα επιδρά στη λειτουργία αυτών των προγραμμάτων και κατ' επέκταση στην υγεία μας. Ωστόσο, μέχρι τώρα και για λόγους τόσο αξιοπιστίας όσο και εξέλιξης, κανένας υπολογιστής και με κανένα πρόγραμμα δε γίνεται λήψη

ιατρικής απόφασης. Το μόνο που προσφέρει ο υπολογιστής είναι το εργαλείο για την εργασία του ιατρού. Αυτός έχει τον έλεγχο και την ηθική και νομική ευθύνη.

Οι αποφάσεις που λαμβάνει ένας γιατρός σε δεδομένες συνθήκες εξαρτώνται από τις γνώσεις του, τόσο θεωρητικές όσο και εμπειρικές και το συνδυασμό τους. Η απόκτηση των θεωρητικών γνώσεων επιτυγχάνεται με τη μεθόδευση και καταγραφή διάφορων εφαρμογών και των στατιστικών τους αποτελεσμάτων. Αποτελεί τη συσσωρευμένη γνώση και εμπειρία όσων προηγήθηκαν στον τομέα αυτό. Είναι προφανές ωστόσο ότι για την απόκτηση εμπειρίας είναι απαραίτητος ο πειραματισμός. Σε όλους τους κλάδους, αλλά πολύ περισσότερο στο χώρο της υγείας τα πειράματα αποτελούν ένα δύσκολο και χρονοβόρο κομμάτι. Τα περιθώρια δοκιμών σε ανθρώπινα σώματα είναι περιορισμένα, τόσο λόγω της αξίας της ανθρώπινης ζωής όσο και λόγω του αριθμού των ανθρώπων που είναι διατεθειμένοι να δοκιμάσουν έναν καινούργιο τρόπο θεραπείας. Η εξέλιξη της ιατρικής ωστόσο είναι αδύνατη χωρίς τη διεξαγωγή πειραμάτων. Επίσης, η απόκτηση εμπειρίας από τους νέους γιατρούς πραγματοποιείται μόνο με την εφαρμογή των θεωρητικών τους γνώσεων και το συνδυασμό τους για τη θεραπεία ασθενών. Πρέπει δηλαδή ο γιατρός να δοκιμάσει να εφαρμόσει κάποια μέθοδο θεραπείας ελπίζοντας να πετύχει κάποιο αποτέλεσμα. Τα περιθώρια λάθους όμως ακόμη και στα πολύ πρώιμα στάδια είναι μηδαμινά και οι δοκιμές που μπορούν να γίνουν είναι σίγουρα λίγες.

Απάντηση σε αυτό το πρόβλημα δίνει η εικονική πραγματικότητα, η προσομοίωση μίας πραγματικής ή φανταστικής κατάστασης σε εικονικό-πλασματικό περιβάλλον. Στο περιβάλλον αυτό κάθε δοκιμή είναι επιτρεπτή και το αντίκτυπο κάθε σφάλματος είναι ανύπαρκτο. Η διαδικασία είναι αναστρέψιμη και μπορεί να επαναληφθεί όσες φορές χρειαστεί και από οποιοδήποτε σημείο κάποιας προηγούμενης προσπάθειας. Η εικονική πραγματικότητα αποτελεί το πιο πρόσφορο έδαφος για τη διεξαγωγή πειραμάτων. Είναι σχεδόν ανέξοδη και παρέχει άμεση λήψη αποτελεσμάτων όσο χρονοβόρο κι αν είναι το πείραμα σε πραγματικές συνθήκες. Σε εικονικό περιβάλλον μπορεί να προσομοιωθεί εύκολα οποιαδήποτε κατάσταση. Η εξέλιξη των υπολογιστών και της πληροφορικής παρέχουν επεξεργαστική ισχύ που μπορεί να επιταχύνει τις διαδικασίες, να εκμηδενίσει τους χρόνους, να δοκιμάσει με ευκολία εναλλακτικές διαδικασίες και συνδυασμούς τους. Η μεθόδευση και η καταγραφή είναι σίγουρα ευκολότερες στο περιβάλλον αυτό, απ' ό,τι στην πραγματικότητα. Είναι φανερό εξάλλου ότι η εικονική πραγματικότητα έχει εισχωρήσει σε πολλούς τομείς, παρέχοντας θαυμαστά αποτελέσματα.

Κεφάλαιο 1

Εισαγωγή – Το πρόβλημα

1.1 Εισαγωγικά στοιχεία μοντελοποίησης

Κατά τη χρήση της εικονικής πραγματικότητας με τη βοήθεια υπολογιστή σε κάποια ιατρική εφαρμογή, εκτός από το κομμάτι του προγραμματισμού, το οποίο θα πρέπει να είναι προσαρμοσμένο στα εκάστοτε ιατρικά δεδομένα, δύο είναι τα βασικά προβλήματα που παρουσιάζονται.

Το πρώτο αφορά στην αποτελεσματική μοντελοποίηση, ώστε το πρόγραμμα να ανταποκρίνεται στην πραγματικότητα. Η μοντελοποίηση είναι μια αφαιρετική διαδικασία, και απαιτεί προσοχή στο διαχωρισμό των ουσιαστικών στοιχείων, αλλά και στο διαχωρισμό των καταστάσεων. Όπως κάθε άτομο έτσι και το κάθε ανθρώπινο σώμα διακρίνεται από τη μοναδικότητα και τη διαφορετικότητά του. Έτσι, δεν υπάρχει ιδανικό ανθρώπινο σώμα στο οποίο θα στηριχθούμε για τη μοντελοποίηση. Άμεση συνέπεια είναι να λειτουργούμε προσεγγιστικά. Όσο γενικότερο είναι το πρόβλημα που μοντελοποιούμε τόσο καλύτερη θα είναι και η προσέγγισή μας. Για παράδειγμα, σε επίπεδο κύριου φλεβικού ή νευρικού συστήματος μπορούμε να έχουμε μία σωστή γενική εικόνα χωρίς σημαντικές αποκλίσεις για τους περισσότερους ανθρώπους. Άλλο παράδειγμα είναι η θέση των οστών ή των οργάνων, την οποία μπορούμε επίσης να γνωρίζουμε. Τα τελευταία παραδείγματα ήδη έχουν μοντελοποιηθεί και υπάρχουν σε υπολογιστή με αρκετή ακρίβεια. Ωστόσο, αν για το μοντέλο μας χρειαζόμαστε την τριχοφυΐα ή την απόχρωση του δέρματος, ή ακόμη χειρότερα την εγκεφαλική λειτουργία είναι προφανές ότι δε μπορούμε να έχουμε σωστά αποτελέσματα χρησιμοποιώντας για πρότυπο έναν άνθρωπο.

Θεωρώντας ότι το μοντέλο έχει προγραμματιστεί και λειτουργεί σωστά προκύπτει ένα δεύτερο πρόβλημα. Το πρόβλημα αυτό έχει να κάνει με την οπτική (και όχι μόνο) απεικόνιση του μοντέλου, και από αυτό την εξαγωγή αποτελεσμάτων και συμπερασμάτων. Είναι προφανές ότι και εδώ η απεικόνιση είναι πολύπλοκη και η τελική εικόνα σίγουρα ελλιπής. Στο μοντέλο σίγουρα κάποια στοιχεία θα είναι

ανακριβή ή θα απουσιάζουν όπως οσμή, ή η υφή, τα οποία θα ήταν χρήσιμα στον ιατρό. Οι δυνατότητες θα είναι περιορισμένες (δε θα δίνεται στο γιατρό η δυνατότητα της ψηλάφησης), και η παρουσία του προγραμματιστή, σε πρώτο στάδιο τουλάχιστον θα είναι αναγκαία. Η εξαγωγή συμπεράσματος θα είναι φυσικά περιορισμένη για το γιατρό αφού μπροστά του δε θα έχει έναν άνθρωπο, ένα πλήρες ανθρώπινο σώμα, αλλά μία προσεγγιστική του απεικόνιση.

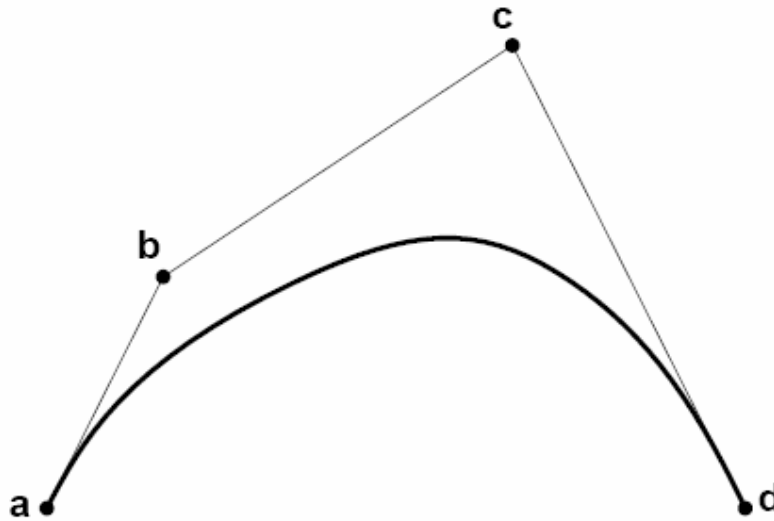
Αν και οι δυνατότητες είναι ακόμη περιορισμένες οι προσπάθειες στον τομέα αυτό έχουν θετικά αποτελέσματα, και η μοντελοποίηση μπορεί ακόμη να μην επαρκεί για εφαρμογές σε κάθε άνθρωπο, αλλά σίγουρα μπορεί να συντελέσει σε άλλους σκοπούς, όπως εκπαιδευτικούς.

Η εργασία αυτή αποτελεί τμήμα της επίλυσης ενός μεγαλύτερου προβλήματος που αφορά στην προσομοίωση μιας παρακέντησης. Στη συνολική προσπάθεια, η οποία γίνεται ολόκληρη με χρήση υπολογιστή, εκτός από απεικόνιση υπάρχει και η αίσθηση της αφής με τη χρήση κατάλληλου οργάνου με ανάδραση το οποίο παρουσιάζεται ως βελόνα στην οθόνη υπολογιστή. Έτσι ο χρήστης μπορεί να πιάσει τη βελόνα και να αισθανθεί την αντίσταση του δέρματος, καθώς εισχωρεί και μετακινείται η βελόνα στους ιστούς. Ταυτόχρονα, μπορεί να δει στην οθόνη τη βελόνα και την παραμόρφωση του σώματος. Για την υλοποίηση χρησιμοποιείται ένα υπάρχον μοντέλο του σχήματος της επιδερμίδας καθώς και των οστών του ανθρώπου στο οποίο όμως έγιναν οι κατάλληλες τροποποιήσεις για την παραμόρφωση του δέρματος, τον υπολογισμό της δύναμης της ανάδρασης και την απεικόνιση.

1.2 Μέθοδοι μοντελοποίησης

Οι υπολογιστές αποτελούν πλέον αναπόσπαστο κομμάτι στη μοντελοποίηση και την προσομοίωση. Μοντέλα αντικειμένων σε γραφικό περιβάλλον μέσω υπολογιστή εξετάζονται εδώ και τρεις δεκαετίες. Αντικείμενα με δυνατότητα παραμόρφωσης αποτελούν συνεχώς αντικείμενο έρευνας και επεξεργασίας. Τέτοια μοντέλα χρησιμοποιούνται ευρέως για την κατασκευή ανθρώπινων σωμάτων, ζώων, ρούχων, προσώπων με διάφορες εκφράσεις και πολλές άλλες μορφές.

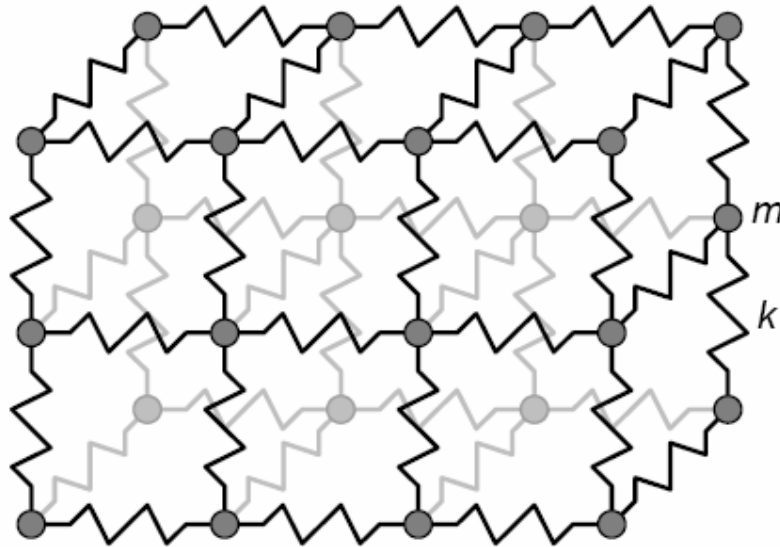
Απλά, μη-φυσικά μοντέλα (non-physical models) βασισμένα σε γεωμετρία και μαθηματικές συναρτήσεις αποτελούν και τις πρώτες προσπάθειες προς την κατεύθυνση αυτή. Οι τεχνικές που ακολουθούνται δεν απαιτούν κατά κανόνα μεγάλη υπολογιστική ισχύ, ωστόσο είναι πολύ χρονοβόρες και εξαρτώνται από τις ικανότητες του προγραμματιστή. Στο σχήμα που ακολουθεί βλέπουμε ένα απλό μοντέλο μιας καμπύλης spline, η οποία εξαρτάται από τέσσερα σημεία ελέγχου. Για τη δημιουργία μιας πολύπλοκης σύνθεσης όμως, όπως ενός ανθρώπινου προσώπου, μία διαδικασία που χρησιμοποιεί απλά γεωμετρικά σχήματα είναι απαγορευτική.



Σχήμα (1.1): Μια καμπύλη (cubic) spline. Για να αλλάξει η μορφή της αρκεί να μετατοπίσουμε τα τέσσερα σημεία ελέγχου από τα οποία εξαρτάται.[1]

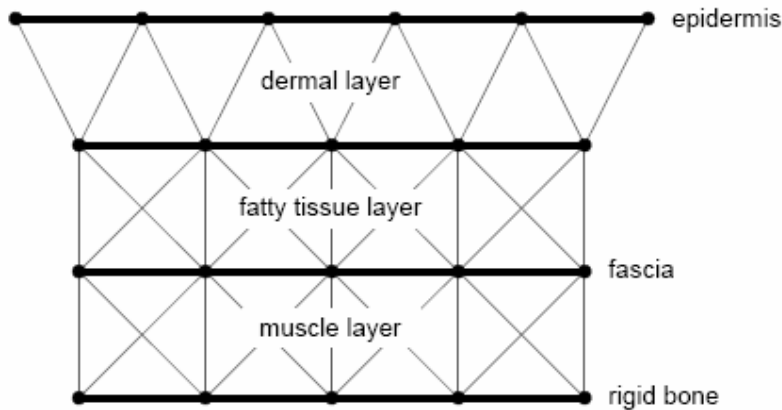
Η εξέλιξη των υπολογιστικών συστημάτων έδωσε νέα πνοή στη μοντελοποίηση. Πολύπλοκες δομές είναι πλέον υλοποιήσιμες με την επεξεργασία της παραμόρφωσης να βαρύνει πλέον τον υπολογιστή. Η διαδικασία ελέγχου και τροποποιήσεων είναι πολύ πιο απλή στην περίπτωση αυτή. Συστήματα με μάζες και ελατήρια (mass-spring systems) έχουν αντικαταστήσει με μεγάλη επιτυχία τις γεωμετρικές δομές. Σε αυτές τις δομές μία παραμόρφωση υλοποιείται εύκολα με την αλλαγή της θέσης μίας από

τις μάζες η οποία μέσω των ελατηρίων παρασύρει και τις γειτονικές της. Οι δυνάμεις των ελατηρίων είναι συχνά γραμμικές (Hookean), αλλά και μη γραμμικά ελατήρια χρησιμοποιούνται για τη μοντελοποίηση ιστών όπως του ανθρώπινου δέρματος που παρουσιάζουν ανελαστική συμπεριφορά.



Σχήμα (1.2): Τμήμα μοντέλου από ελατήρια και μάζες.[1]

Με τις μάζες να αποτελούν τους κόμβους και τα ελατήρια τις ακμές, μπορούμε να δούμε μία εφαρμογή της τελευταίας μεθόδου στην κατασκευή ιστού του δέρματος του προσώπου.



Σχήμα (1.3): Μοντέλο δέρματος προσώπου (Terzopoulos and Waters). [1]

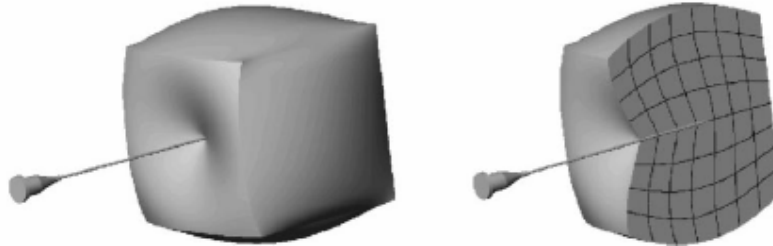
Οι δομές με μάζες και ελατήρια παρέχουν μια σχετικά εύκολη και γρήγορη υλοποίηση, ωστόσο εμφανίζουν και κάποια προβλήματα. Για μια αληθοφανή

υλοποίηση απαιτείται η εξαγωγή παραμέτρων από πραγματικά σώματα, κάτι που συχνά αποδεικνύεται αρκετά δύσκολο. Επίσης, σε ορισμένες περιπτώσεις η δομή μας μπορεί να είναι εξαιρετικά ασταθής και να προκαλεί σημαντική επιβράδυνση στην προσομοίωση.

Συνεχή μοντέλα με τη μέθοδο των πεπερασμένων στοιχείων παρέχουν μία πιο ρεαλιστική υλοποίηση σε σχέση με αυτή των μαζών-ελατηρίων. Οι κόμβοι σε αυτή την περίπτωση είναι πολύ λιγότεροι και η πολυπλοκότητα μειώνεται. Αυξάνεται ωστόσο η δυσκολία στην ανάλυση των εφαρμοζόμενων δυνάμεων. Ένα σώμα από πεπερασμένα στοιχεία προσφέρεται για υλοποίηση σε πραγματικό χρόνο, αφού πολλοί αριθμητικοί υπολογισμοί μπορούν να προηγηθούν της υλοποίησης. Ειδικά σε περιπτώσεις άκαμπτων υλικών όπως τα μέταλλα όπου η παραμόρφωση περιορίζεται σε πολύ μικρά ποσοστά της συνολικής διάστασης οι πράξεις που ακολουθούν μετά την έναρξη της υλοποίησης είναι ελάχιστες. Σε δομές με μεγάλη παραμόρφωση μπορούμε να πετύχουμε θαυμάσια αποτελέσματα, υπάρχει ωστόσο μεγαλύτερο υπολογιστικό κόστος. [1]

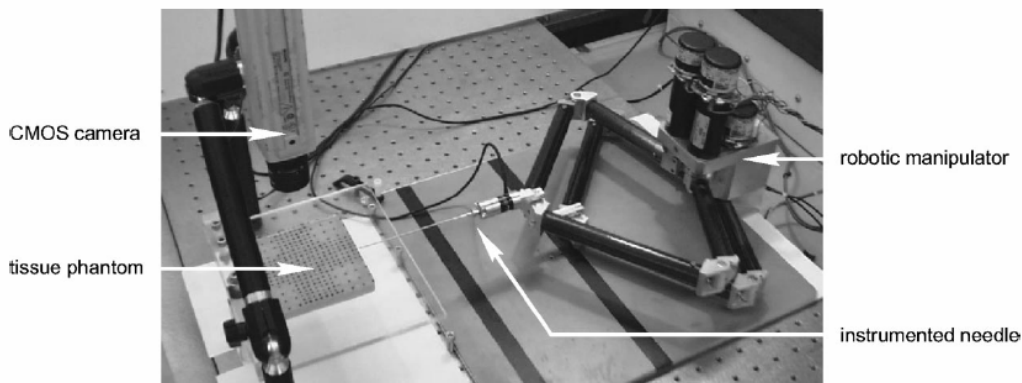
1.3 Άλλες προσπάθειες μοντελοποίησης με πεπερασμένα στοιχεία.

Η χρήση πεπερασμένων στοιχείων στη μοντελοποίηση παραμόρφωσης δέρματος δεν είναι κάτι καινούργιο. Ήδη έχουν γίνει πολλές προσπάθειες στην κατεύθυνση αυτή. Ένα καλό παράδειγμα είναι αυτό του Simon P. DiMaio, με τίτλο “Needle Insertion Modeling and Simulation” [2].



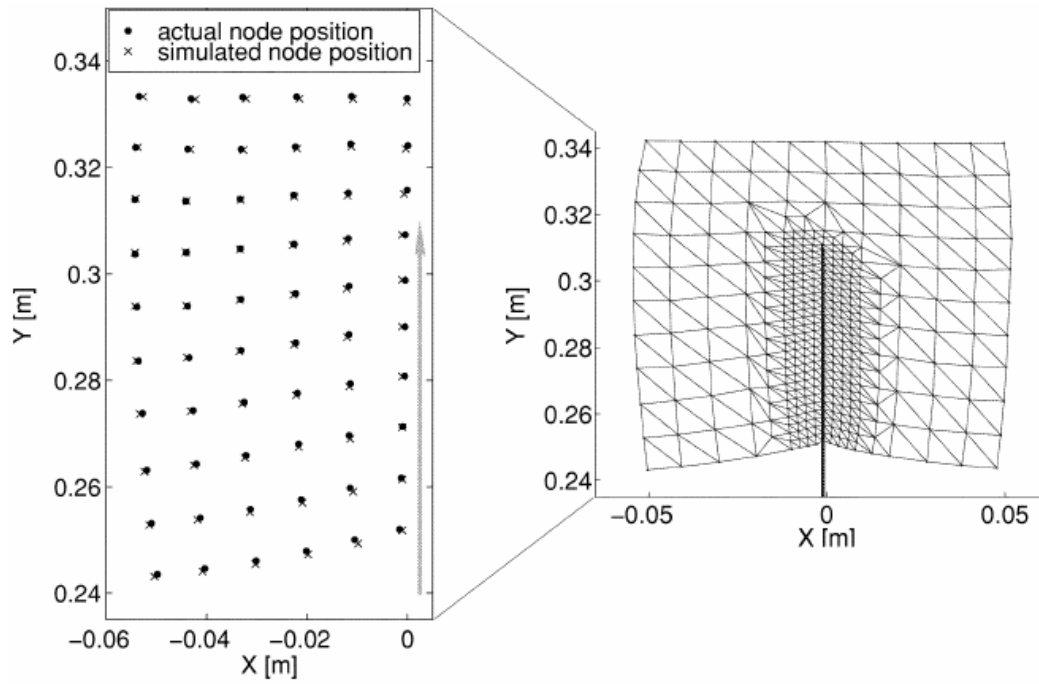
Σχήμα (1.4): Εισαγωγή βελόνας σε μαλακό σώμα [2]

Στην περίπτωση αυτή έχει κατασκευαστεί ένα σύστημα με ένα τεχνητό μοντέλο χωρίς την εφαρμογή πραγματικών δεδομένων δέρματος. Ένα παραλληλεπίπεδο τμήμα τεχνητού ιστού χρησιμοποιήθηκε για τις μετρήσεις, οι οποίες εστιάστηκαν στις δύο διαστάσεις.



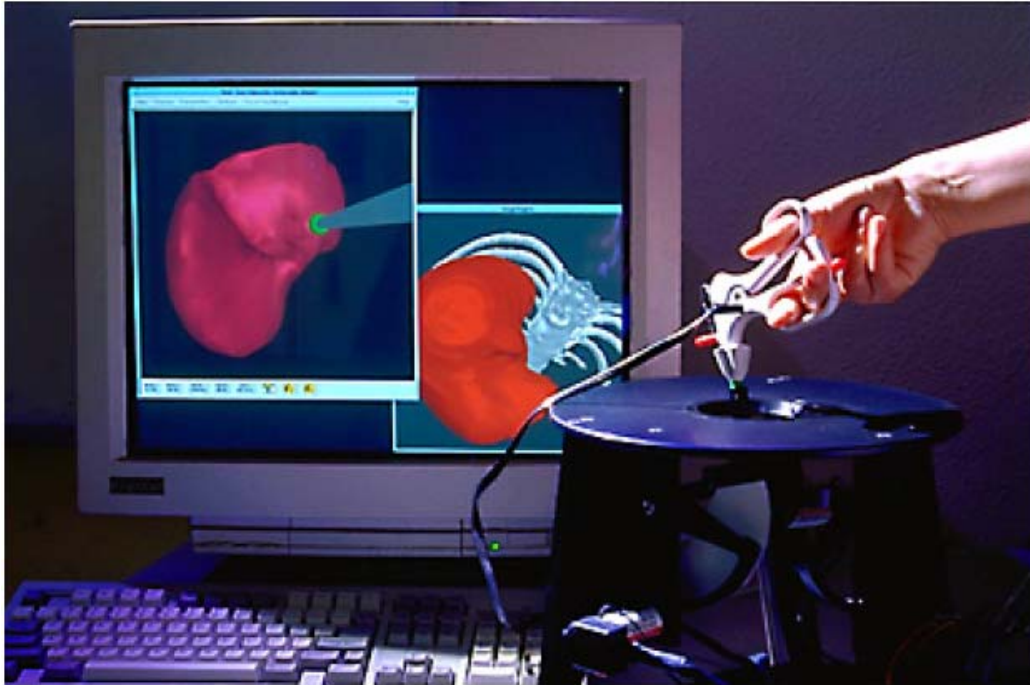
Σχήμα (1.5): Το πείραμα: Ένας ρομποτικός βραχίονας εφοδιασμένος με βελόνα, ένα τμήμα ιστού και μία κάμερα [2]

Το αποτέλεσμα ήταν η δημιουργία ενός διδιάστατου μοντέλου πεπερασμένων στοιχείων σε υπολογιστή, σύμφωνα με τα δεδομένα που παρείχε η κάμερα. Στο μοντέλο έχουμε και διεύθυνση της βελόνας στους ιστούς όπου έχει γίνει αλλαγή της ακρίβειας της δομής όπως φαίνεται στο παρακάτω σχήμα:



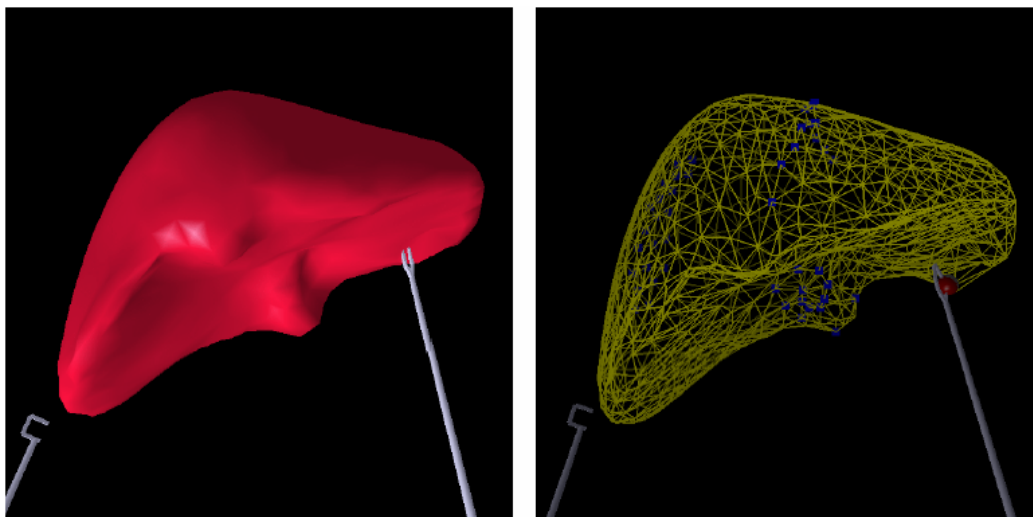
Σχήμα (1.6): Σύγκριση δεδομένων που προκύπτουν από την κάμερα με το μοντέλο πεπερασμένων στοιχείων [2]

Άλλο παράδειγμα είναι η προσπάθεια των Stéphane Cotin, Hervé Delingette και Nicholas Ayache με τίτλο “Real-time elastic deformations of soft tissues for surgery simulation” [3].



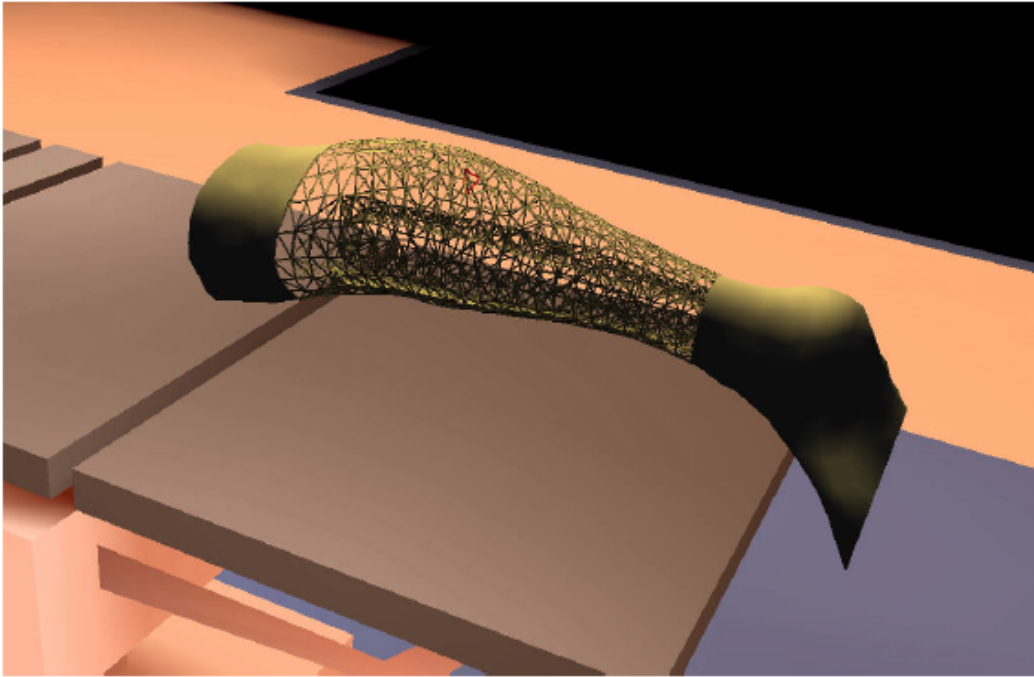
Σχήμα (1.7): Εφαρμογή χειρουργικού εργαλείου σε μοντέλο ανθρώπινου οργάνου [3]

Στην περίπτωση αυτή χρησιμοποιήθηκε χειρουργικό εργαλείο με ανάδραση μέσω υπολογιστή, με το οποίο έγινε προσομοίωση χειρουργικής επέμβασης. Το μοντέλο υλοποιήθηκε με χρήση πεπερασμένων στοιχείων σε τρεις διαστάσεις.

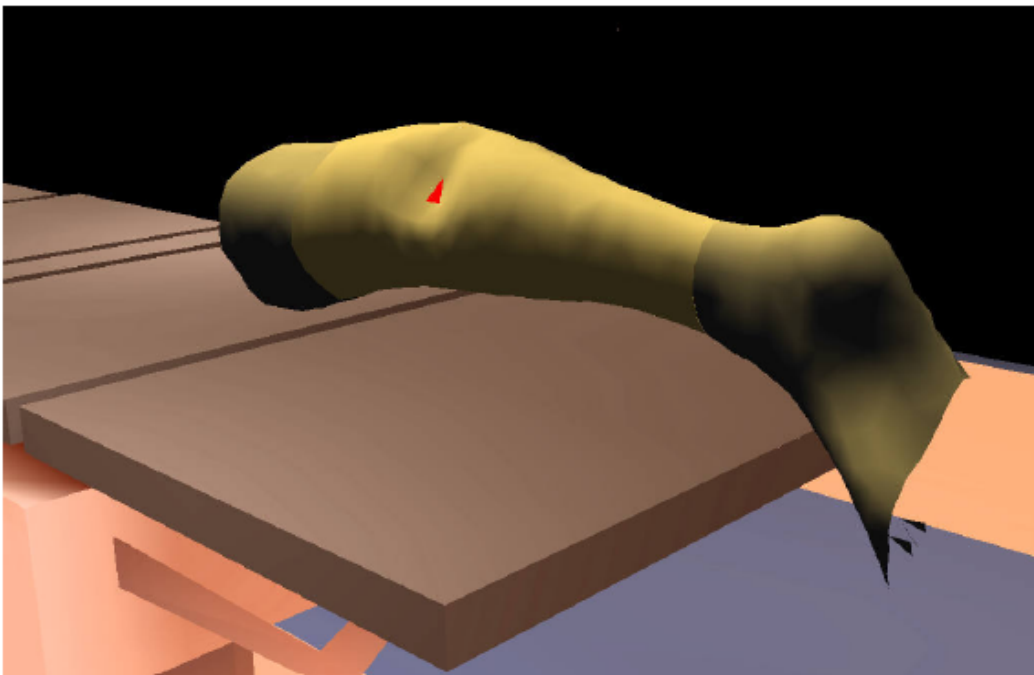


Σχήμα (1.8): Υλοποίηση μοντέλου συκωτιού σε τρεις διαστάσεις [3]

Αξίζει επίσης να αναφέρουμε την εργασία του Morten Bro-Nielsen [4] όπου μεταξύ άλλων έχει γίνει και μοντελοποίηση μελών του σώματος με πεπερασμένα στοιχεία.



Σχήμα (1.9): Υλοποίηση τμήματος ποδιού σε τρεις διαστάσεις [4]

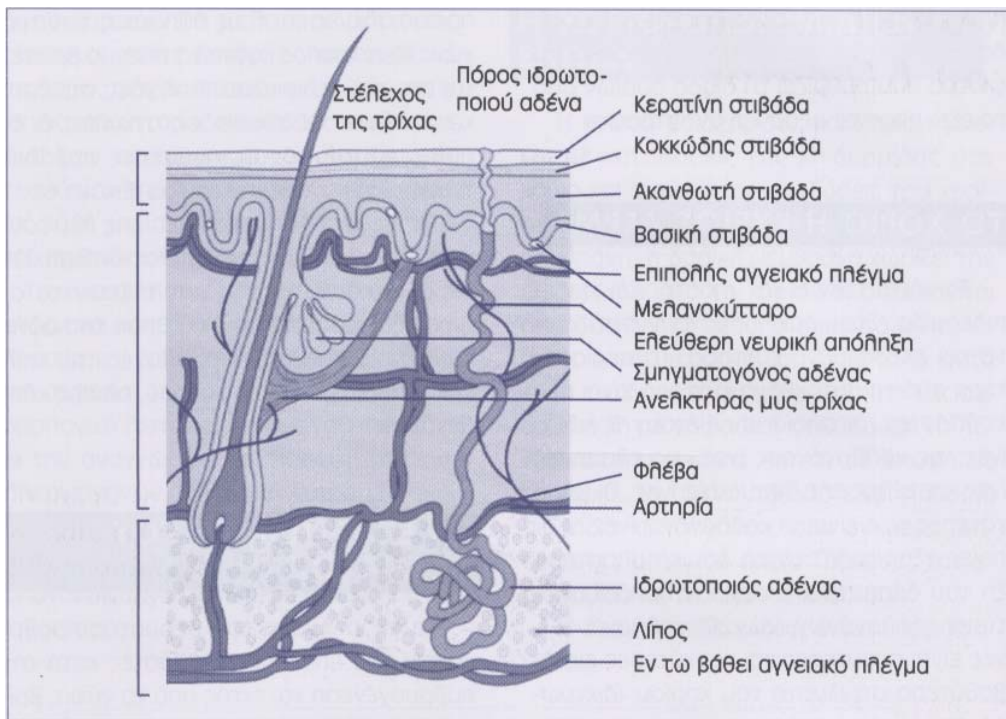


Σχήμα (1.10): Προσομοίωση παραμόρφωσης επιφάνειας γάμπας [4]

Στη δική μας περίπτωση ασχολούμαστε με μια διδιάστατη δομή τομής δέρματος από την οποία προκύπτει τελικά ένα τρισδιάστατο μοντέλο. Δίνεται έμφαση στην προσαρμογή του μοντέλου σε πραγματικά πειραματικά δεδομένα, έτσι ώστε η δομή να ανταποκρίνεται, ως προς την παραμόρφωση, σε πραγματικό τμήμα ανθρώπινου δέρματος και το αποτέλεσμα να είναι περισσότερο αληθοφανές.

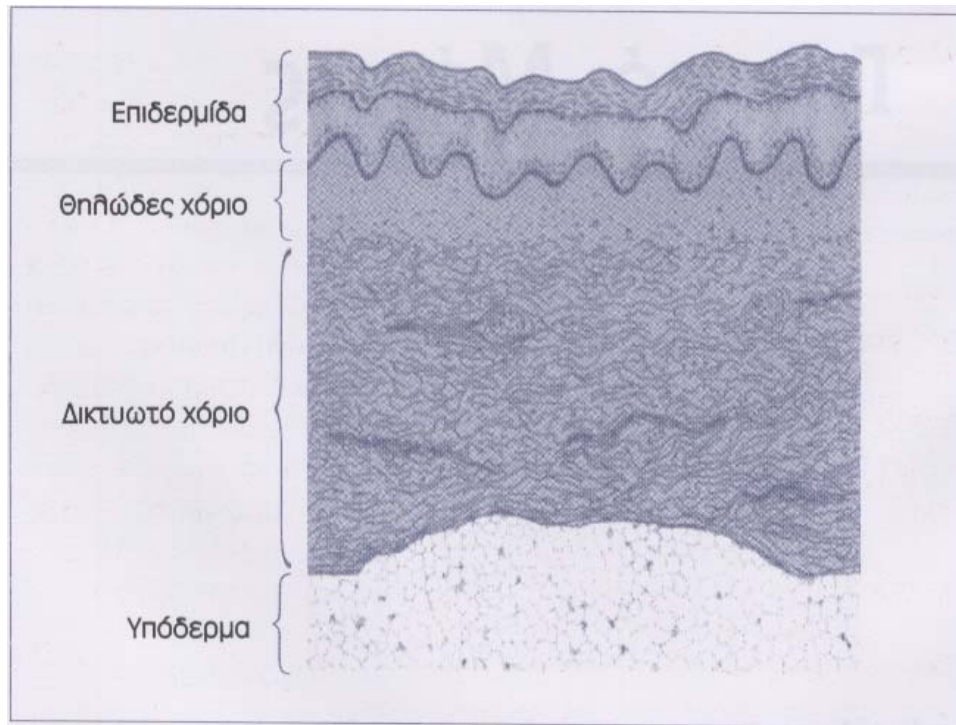
1.4 Βιοϊατρικά δεδομένα του προβλήματος

Το μοντέλο το οποίο υλοποιείται αποτελείται από μικρό τμήμα του δέρματος, η επιφάνεια του οποίου λαμβάνεται κατά προσέγγιση επίπεδη. Κάτω από αυτό δεχόμαστε ότι υπάρχει επίπεδο οστό, υπάρχει επομένως μία επίπεδη ανελαστική επιφάνεια. Ο χώρος μεταξύ των δύο επιφανειών, αυτής του δέρματος και αυτής του οστού, δεν είναι βέβαια ομογενής. Το δέρμα περιέχει διάφορα ανατομικά στοιχεία όπως τρίχες, αδένες, φλέβες και αρτηρίες τα οποία συνθέτουν μία περίπλοκη εικόνα. Τα στοιχεία αυτά διαφοροποιούνται τόσο σε σχέση με το βάθος στο οποίο βρισκόμαστε όσο και με τη θέση του υπό εξέταση τμήματος του δέρματος στο ανθρώπινο σώμα. Η τριχοφυΐα, ή η ποσότητα λίπους για παράδειγμα εμφανίζουν μεγάλες αποκλίσεις αν εξετάσουμε δύο διαφορετικά σημεία του σώματος όπως τις παλάμες σε σχέση με το στήθος ή την κοιλιά. Επίσης, κάποια ανατομικά στοιχεία μπορεί να εξαρτώνται και από το φύλο την ηλικία ή τις ιδιαιτερότητες του σώματος κάθε ατόμου. Φυσικά, η προσέγγισή μας δε μπορεί να συμπεριλάβει όλα τα στοιχεία αυτά. Η μοντελοποίηση θα αναφέρεται σε ένα τμήμα του δέρματος το οποίο θεωρείται ενδεικτικό για τα στοιχεία τα οποία μας ενδιαφέρουν (στην προκειμένη περίπτωση μας ενδιαφέρει η ελαστικότητα). Το τμήμα αυτό θα λαμβάνεται ότι είναι αντιπροσωπευτικό για οποιοδήποτε σημείο του δέρματος. Ενδεικτικά ωστόσο, μπορούμε να δούμε τα βασικά ανατομικά στοιχεία του δέρματος στο σχήμα που ακολουθεί.



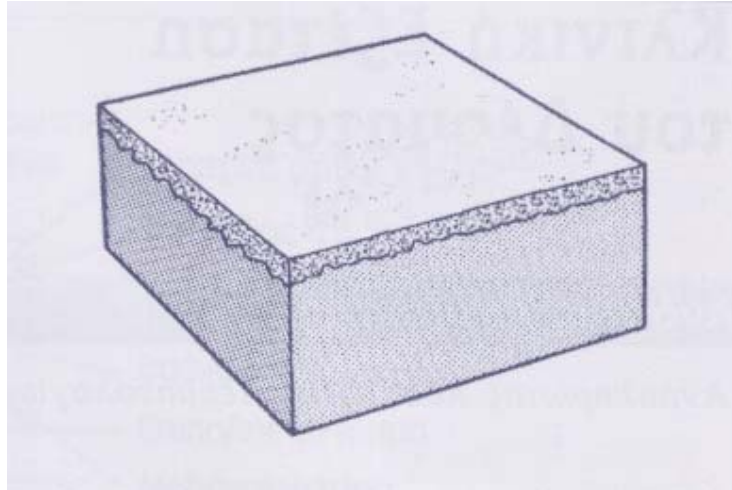
Σχήμα (1.11): Ανατομικά στοιχεία του δέρματος [5]

Εξετάζοντας ένα αντιπροσωπευτικό τμήμα του δέρματος θα διαπιστώσουμε ότι αποτελείται από πολλές διαδοχικές στιβάδες. Ξεκινώντας από την επιφάνεια και προχωρώντας προς τα κάτω συναντούμε αρχικά την επιδερμίδα, την εξωτερική, επιθηλιακή στιβάδα, ενώ κατόπιν έχουμε το κυρίως δέρμα και το υπόδερμα. Το κυρίως δέρμα, ή χόριο αποτελείται από δύο τμήματα, το θηλώδες χόριο το οποίο συνδέεται με την επιδερμίδα και το δικτυωτό χόριο το οποίο αποτελεί και το κάτω όριο του κυρίως δέρματος. Τέλος, υπάρχει ένας χαλαρός συνεκτικός ιστός, το υπόδερμα ή υποδόριος ιστός το οποίο και περιέχει άφθονο λίπος. [5]



Σχήμα (1.12): Στιβάδες του δέρματος [5]

Για την προσέγγισή μας το δέρμα θεωρείται ομογενές και αποτελείται από δύο μόνο στιβάδες. Επομένως, θεωρούμε δύο παράλληλες επίπεδες επιφάνειες, από τις οποίες η πάνω είναι εύκαμπτη και προσομοιώνει το εξωτερικό της επιδερμίδας και η κάτω είναι σταθερή, αποτελεί το οστό. Ο ενδιάμεσος χώρος αποτελεί το τμήμα του υπό εξέταση δέρματος. Προσεγγιστικά, για την προσομοίωση μπορούμε να υποθέσουμε ότι ο ενδιάμεσος χώρος αποτελείται από δύο εύκαμπτα διαδοχικά στρώματα διαφορετικού πάχους και διαφορετικής ελαστικότητας. Το επάνω ξεκινάει από την επιφάνεια του δέρματος αποτελώντας κατά προσέγγιση το κυρίως δέρμα ενώ το κάτω συνεχίζει μέχρι το άκαμπτο οστό και θεωρείται ότι προσομοιώνει το υπόδερμα. Το κατώτερο στρώμα θα έχει μεγαλύτερη ελαστικότητα από το πρώτο.



Σχήμα (1.13): Τομή δέρματος [5]

Στο μοντέλο αυτό θέλουμε να προσδιορίσουμε την παραμόρφωσή της επιδερμίδας πριν την εισαγωγή βελόνας σε κάποιο σημείο της. Έτσι, αν πάρουμε σημειακή μάζα σε ένα σημείο της επιφάνειας, το σημείο επαφής θα βρούμε την κατανομή της επιφάνειας σε συνάρτηση με το βάθος του σημείου αυτού. Η μελέτη μας περιλαμβάνει τα σημεία από την επαφή της βελόνας μέχρι ακριβώς πριν τη διάτρηση της επιδερμίδας και την εισαγωγή της βελόνας στους ιστούς.

1.5 Πλαίσιο εργασίας

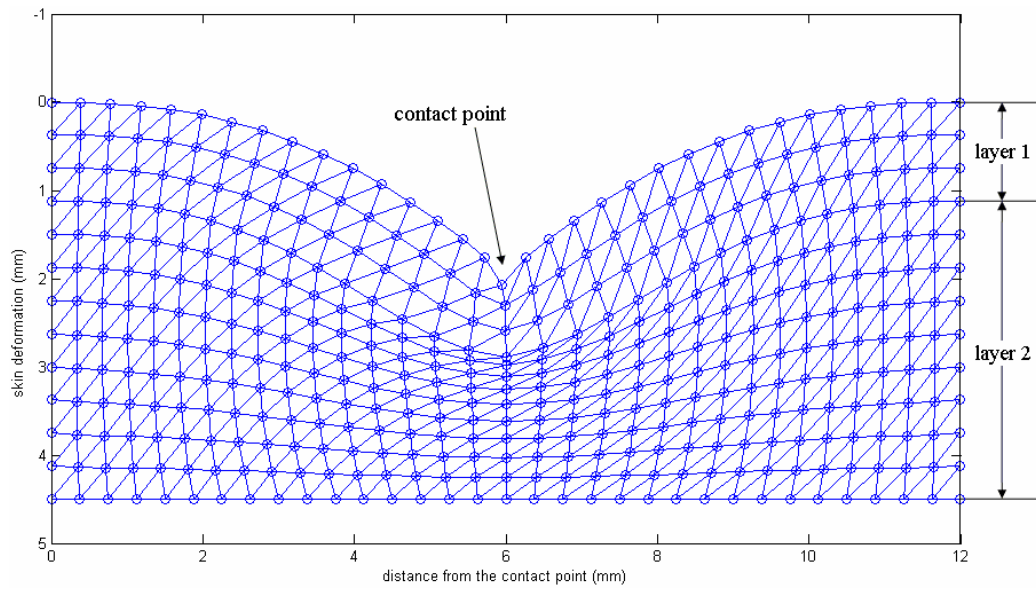
Η συγκεκριμένη εργασία αποτελείται από δύο σκέλη. Και τα δύο αναφέρονται στο στάδιο από την επαφή της βελόνας με το δέρμα, μέχρι το σημείο ακριβώς πριν τη ρήξη του ιστού του δέρματος. Στο πρώτο σκέλος υπολογίζονται διάφορες παράμετροι (όπως η ελαστικότητα) με βάση πειραματικά δεδομένα, για την προσομοίωση της παραμόρφωσης του δέρματος. Στο δεύτερο σκέλος, με χρήση των αρχικών αποτελεσμάτων γίνεται μια «real time» υλοποίηση όπου για τις διάφορες τιμές του βάθους μετατόπισης του σημείου επαφής βελόνας και δέρματος έχουμε και την αντίστοιχη απεικόνιση.

Για τη μοντελοποίηση μπορούμε να θεωρήσουμε επιπλέον ότι η διαδρομή της βελόνας είναι ευθεία με κατεύθυνση προς τα κάτω, και να υποθέσουμε επομένως ότι έχουμε αξονική συμμετρία (ως προς τον άξονα της βελόνας). Για το μοντέλο μας επομένως μπορούμε να περιοριστούμε σε διδιάστατη δομή η οποία επαναλαμβάνεται ως προς τον άξονα που ορίζει η πορεία του σημείου επαφής.

Η μέθοδος που χρησιμοποιούμε είναι αυτή των πεπερασμένων στοιχείων. Στα πεπερασμένα στοιχεία παράγοντας που αυξάνει κατά πολύ την πολυπλοκότητα είναι η εφαρμογή εξωτερικών δυνάμεων. Στην περίπτωση μας έχουμε μονάχα μία δύναμη η οποία λαμβάνεται σημειακή. Δυνάμεις βέβαια ασκούνται και από τα όρια-τοιχώματα της δομής, χωρίς ωστόσο να δυσχεραίνουν το πρόβλημά μας. Επειδή θέλουμε να πετύχουμε μία ακριβή απεικόνιση η μέθοδος που επιλέξαμε είναι η ιδανική για τις δεδομένες συνθήκες του προβλήματος.

Έτσι κατασκευάζουμε ένα σύστημα δύο διαδοχικών στρωμάτων από εύκαμπτο υλικό με το σημείο επαφής στο κέντρο της επάνω επιφάνειας του πρώτου στρώματος και άκαμπτη την κάτω επιφάνεια του δεύτερου στρώματος. Η μορφή που έχει το πρώτο στρώμα είναι ίδια με αυτή του δεύτερου, με μόνη διαφορά το πάχος και την ελαστικότητά τους.

Μία τομή του επιθυμητού αποτελέσματος φαίνεται στο παρακάτω σχήμα:



Σχήμα (1.14): Αναπαράσταση τομής του δέρματος

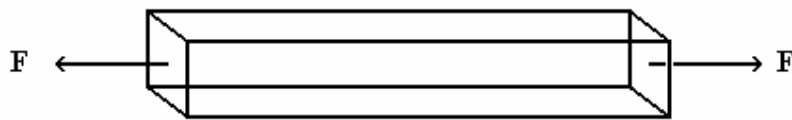
Κεφάλαιο 2

Μοντελοποίηση ελαστικής παραμόρφωσης με χρήση πεπερασμένων στοιχείων

2.1 Βασικά στοιχεία θεωρίας ελαστικής παραμόρφωσης

Πριν προχωρήσουμε σε ανάλυση της μεθόδου των πεπερασμένων στοιχείων θα πρέπει να ορίσουμε με μαθηματικό τρόπο την ελαστική παραμόρφωση ενός σώματος και τις παραμέτρους που τη συνοδεύουν. Οι παράμετροι αυτές θα πρέπει να συμπεριληφθούν στη δομή του μοντέλου μας έτσι ώστε αυτό να βρίσκεται σε συμφωνία με το ανθρώπινο δέρμα.

Ας θεωρήσουμε αρχικά ένα σώμα σε σχήμα ράβδου με μήκος L και εμβαδό διατομής A . Στο σώμα αυτό ασκούμε μία δύναμη F κατά τον άξονά του όπως φαίνεται στο σχήμα (2.1).



Σχήμα (2.1): Εφαρμογή δύναμης σε σώμα

Ορίζεται τότε η ένταση – stress (σ) σύμφωνα με τη σχέση:

$$\sigma = \frac{F}{A} \quad (2.1)$$

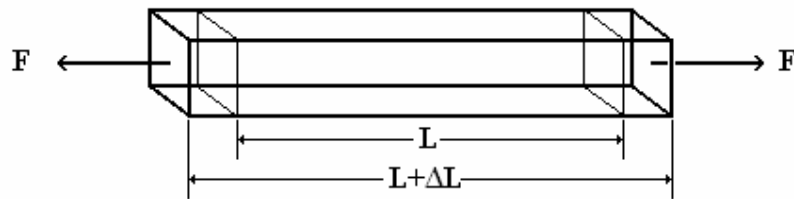
Η ένταση – πίεση – είναι δύναμη ανά μονάδα επιφάνειας με ορισμένη την κατεύθυνση τόσο της δύναμης, όσο και του διανύσματος της επιφάνειας.

Το σώμα αυτό εξ' αιτίας της εφαρμογής της δύναμης θα παρουσιάσει μία μεταβολή του μήκους του L κατά ΔL . Σύμφωνα με τη μεταβολή αυτή (που ονομάζεται παραμόρφωση) ορίζεται άλλο ένα μέγεθος που ονομάζεται διάταση – strain (ε).

Η διάταση ορίζεται ως το πηλίκο της παραμόρφωσης προς το αρχικό μήκος.

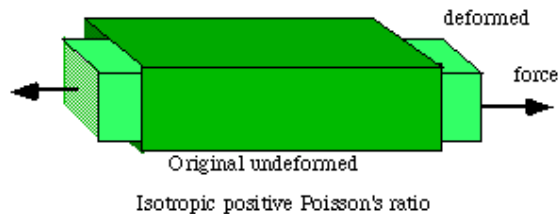
$$\varepsilon = \frac{\Delta L}{L} \quad (2.2)$$

Στο παρακάτω σχήμα βλέπουμε τη μεταβολή του φυσικού μήκους L κατά ΔL .



Σχήμα (2.2): Διάταση σώματος κατά την εφαρμογή δύναμης σε αυτό

Ουσιαστικά όλα τα συνήθη σώματα γίνονται στενότερα κατά τη διάτασή τους. Ο λόγος για τον οποίο γίνεται αυτό είναι ότι στην πλειοψηφία τους τα σώματα αντιστέκονται σε μια αλλαγή στον όγκο όπως καθορίζεται από το μαζικό συντελεστή K (bulk modulus) περισσότερο απ' ό τι αντιστέκονται σε μια αλλαγή στη μορφή, όπως καθορίζεται από το συντελεστή συνάφειας G (shear modulus). Αυτό φαίνεται και στο παρακάτω σχήμα:

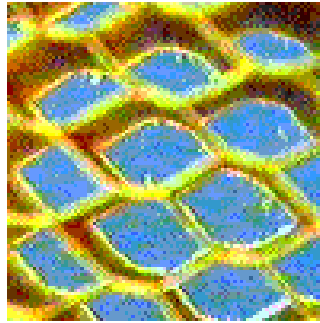


Σχήμα (2.3): Στένεμα σώματος κατά τη διάτασή του [6]

Για το λόγο αυτό ορίζεται ο συντελεστής Poisson – Poisson's ratio (ν). Ο συντελεστής αυτός προκύπτει από την αναλογία της εγκάρσιας πίεσης συστολής στη διαμήκη πίεση έκτασης στην κατεύθυνση του τεντώματος της δύναμης. Όταν έχουμε εκτατή παραμόρφωση, ο συντελεστής Poisson θεωρείται θετικός ενώ για συμπιεστική παραμόρφωση λαμβάνεται αρνητικός. Έτσι, στη μαθηματική διατύπωση του συντελεστή Poisson υπάρχει ένα αρνητικό πρόσημο, ώστε τα φυσιολογικά υλικά να έχουν θετικό συντελεστή.

$$\nu = - \frac{\varepsilon_{trans}}{\varepsilon_{longitudinal}} \quad (2.3)$$

Εξετάζοντας το πρόβλημα από τη δομική του άποψη ο λόγος για την αναλογία του συνηθισμένου θετικού συντελεστή Poisson είναι ότι οι δεσμοί των ατόμων ευθυγραμμίζονται με την παραμόρφωση. Το τέντωμα της κηρήθρας από τις κάθετες δυνάμεις, που παρουσιάζεται στο παρακάτω σχήμα δίνει μια πιο απλοϊκή επεξήγηση της έννοιας.



Σχήμα (2.4): Τέντωμα κηρήθρας [6]

Η αναλογία Poisson συσχετίζεται με τους ελαστικούς συντελεστές, το μαζικό συντελεστή K , το συντελεστή συνάφειας G και το συντελεστή ελαστικότητας του Young, E με τις παρακάτω σχέσεις. Οι ελαστικοί συντελεστές είναι μέτρα της ακαμψίας. Είναι αναλογίες της πίεσης στη διάταση. Η σχέση των συντελεστών αυτών φαίνεται στις εξισώσεις που ακολουθούν:

$$\nu = \frac{(3K - 2G)/(6K + 2G)}{2G(1 + \nu)} \quad (2.4)$$

$$E = 2G(1 + \nu) \quad (2.5)$$

Οι επιτρεπτές τιμές της αναλογίας Poisson καθορίζονται από τη θεωρία της ισοτροπικής ελαστικότητας και βρίσκονται στο διάστημα από -1 έως $1/2$. Για να είναι ένα σώμα σταθερό η ένταση θα πρέπει να είναι θετική. Από τους ελαστικούς συντελεστές προκύπτει η τελευταία περιοχή τιμών.

Για κάθε υλικό ο συντελεστής Poisson, όπως και οι ελαστικοί συντελεστές είναι διαφορετικοί. Έτσι τα λαστιχένια υλικά έχουν συντελεστή πολύ κοντά στο $1/2$. Το δέρμα το οποίο έχει και αυτό ελαστικές ιδιότητες έχει συντελεστή Poisson 0,48 ενώ η αντίστοιχη τιμή του συντελεστή ελαστικότητας του Young είναι 0,2 MPa.

Συντελεστές δέρματος

$$\text{Poisson : } \nu = 0,48 \quad (2.6)$$

$$\text{Young : } E = 0,2 \text{ (MPa)} \quad (2.7)$$

Η παραμόρφωση ενός σώματος κατά την εφαρμογή πίεσης συνεπάγεται τη διάτασή του. Η διάταξη σε ένα σημείο περιγράφεται από την εξίσωση:

$$Y = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} \quad (2.8)$$

Στην τελευταία εξίσωση οι όροι ε_{ij} αποτελούν τους συντελεστές της διάταξης. Ο πίνακας Y είναι συμμετρικός για ελαστικά σώματα, επομένως είναι $\varepsilon_{ij} = \varepsilon_{ji}$.

Επομένως, ο παράγοντας διάταξης που προκύπτει γράφεται:

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \\ \varepsilon_{xy} \end{bmatrix} \quad (2.9)$$

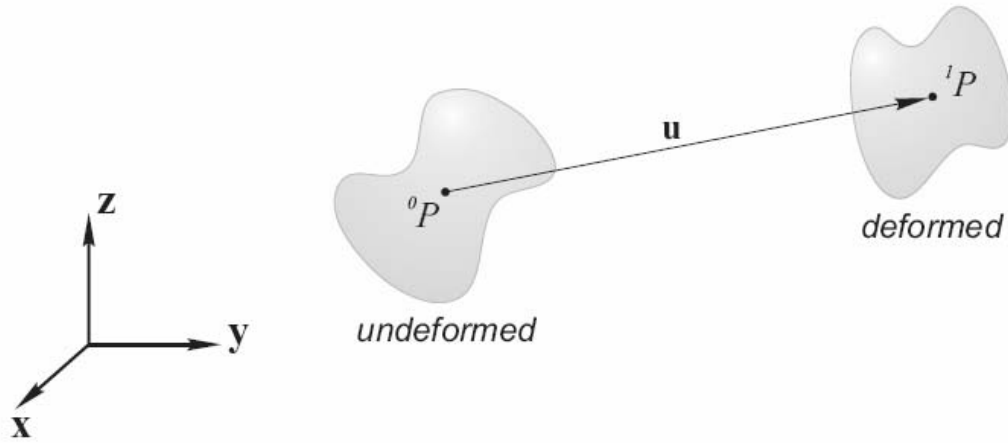
Με ανάλογο τρόπο υπολογίζεται και η πίεση σε σημείο:

$$\Psi = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (2.10)$$

Αντίστοιχα παίρνουμε:

$$\sigma = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{bmatrix} \quad (2.11)$$

Θεωρούμε το παρακάτω σχήμα. Ένα τυχαίο σημείο (P) ενός σώματος μετακινείται κατά ένα διάνυσμα \underline{u} όταν το σώμα υπόκειται σε παραμόρφωση.



Σχήμα (2.5): Σώμα πριν και μετά την παραμόρφωση – Διάνυσμα μετατόπισης σημείου [7]

Το διάνυσμα \underline{u} γράφεται:

$$\underline{u}(x, y, z) = \begin{bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{bmatrix} \quad (2.12)$$

Χρησιμοποιώντας την τελευταία σχέση έχουμε:

$$\varepsilon(\underline{u}) = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = B\underline{u} \quad (2.13)$$

Ας πάρουμε τώρα το γενικευμένο νόμο του Hooke ο οποίος διατυπώνεται ως εξής:

$$\boldsymbol{\sigma} = D\boldsymbol{\varepsilon} \quad (2.14)$$

Με D τον πίνακα ιδιοτήτων του υλικού.

Η τελευταία εξίσωση επιλύεται εύκολα αν χρησιμοποιήσουμε μία ενεργειακή διατύπωση για τη διάταση ενός ελαστικού σώματος.

$$E_{strain} = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}^T(\mathbf{x}) \boldsymbol{\sigma}(\mathbf{x}) d\mathbf{x} \quad (2.15)$$

Αντικαθιστώντας τον πίνακα ιδιοτήτων του υλικού στην τελευταία σχέση έχουμε:

$$E_{strain} = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}^T(\mathbf{x}) D \boldsymbol{\varepsilon}(\mathbf{x}) d\mathbf{x} \quad (2.16)$$

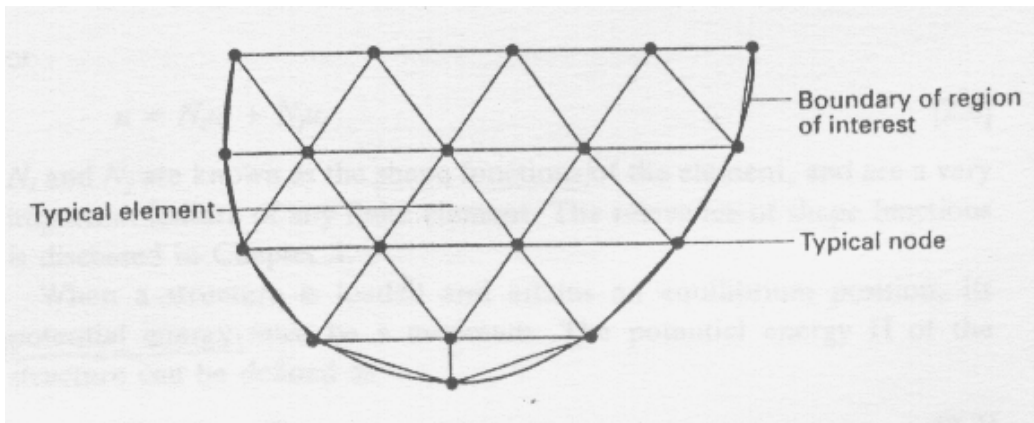
Για όλο το σώμα η συνολική ενέργεια της παραμόρφωσης είναι:

$$E(\mathbf{u}) = \frac{1}{2} \int_{\Omega} (\mathbf{B}\mathbf{u})^T D \mathbf{B}\mathbf{u} d\mathbf{x} \quad (2.17)$$

Οι εξισώσεις που περιγράψαμε είναι γενικές και χρήσιμες για τη μοντελοποίηση ελαστικής παραμόρφωσης. Ακολουθεί ανάλυση της μοντελοποίησης με πεπερασμένα στοιχεία όπου και εφαρμόζονται οι εξισώσεις αυτές.

2.2 Εισαγωγή στη μέθοδο των πεπερασμένων στοιχείων

Η μέθοδος των πεπερασμένων στοιχείων στηρίζεται στις εξής βασικές αρχές. Θεωρούμε ένα αντικείμενο-σώμα στο οποίο θέλουμε να υπολογίσουμε την κατανομή μιας μεταβλητής (όπως παραμόρφωσης ή θερμοκρασίας). Αρχικά χωρίζουμε το σώμα μας σε ένα σύστημα δομικών μονάδων, τα στοιχεία –elements– τα οποία συνδέονται μεταξύ τους σε κόμβους –nodes. Η μεταβλητή εφαρμόζεται σε κάθε στοιχείο με έναν προκαθορισμένο τρόπο. Ο αριθμός και ο τύπος των στοιχείων είναι επιλεγμένα έτσι ώστε η κατανομή της μεταβλητής σε όλο το σώμα να προσεγγίζεται ικανοποιητικά από την αναπαράσταση όλων των στοιχείων. Η διάδοση της μεταβλητής σε κάθε στοιχείο μπορεί να προσδιοριστεί από μία πολυωνυμική συνάρτηση (όπως μία γραμμική) ή μία τριγωνομετρική συνάρτηση.



Σχήμα (2.6): Παράδειγμα σώματος από πεπερασμένα στοιχεία [8]

Αφού το πρόβλημα έχει πλέον γίνει διακριτό, με την αντιστοίχησή του στο σύστημα των πεπερασμένων στοιχείων, είναι έτοιμο να επιλυθεί. Αρχικά προσδιορίζονται οι συναρτήσεις που ορίζουν το κάθε στοιχείο και στη συνέχεια συνυπολογίζονται για να δώσουν τις εξισώσεις του συστήματος. Ο υπολογισμός αυτός μπορεί να γίνει με διάφορους τρόπους αλλά η τελική μορφή είναι συγκεκριμένη για ορισμένο τύπο προβλήματος (όπως παραμόρφωσης ή θερμικό). Επομένως, εφόσον οριστούν οι συναρτήσεις που αφορούν κάθε τύπο στοιχείου, οι εξισώσεις του συστήματος υπολογίζονται άμεσα. Είναι απλά θέμα αντιστοίχισης των συντεταγμένων των κόμβων, των αρχικών συνθηκών και των ιδιοτήτων του σώματος.

Έτσι, προκύπτει ένα σύστημα το οποίο περιγράφει όλο το σώμα. Το σύστημα αυτό γράφεται στη μορφή:

$$[K] \cdot \underline{U} = \underline{F} \quad (2.18)$$

όπου $[K]$ ένας τετραγωνικός πίνακας που λέγεται «μητρώο σκληρότητας» (stiffness matrix), \underline{U} το διάνυσμα των μετατοπίσεων (ή των θερμοκρασιών) των κόμβων (nodes) και \underline{F} το διάνυσμα των εφαρμοζόμενων δυνάμεων στους αντίστοιχους κόμβους. Τα δύο διανύσματα αναπαρίστανται με πίνακες στήλη.

Η επίλυση της παραπάνω εξίσωσης αφορά στον υπολογισμό του διανύσματος \underline{U} με γνωστά τα $[K]$ και \underline{F} . Ωστόσο, η προφανής λύση με αντιστροφή του πίνακα $[K]$ δεν είναι πάντοτε εύκολη. Δεν είναι απίθανο να έχουμε πλήθος εξισώσεων της τάξης των 50000 κάτι που δυσχεραίνει την επίλυση του προβλήματος. Η μορφή ωστόσο του πίνακα $[K]$ είναι συνήθως τέτοια που επιτρέπει την εφαρμογή μεθόδων επίλυσης περισσότερο αποδοτικών τόσο σε χρόνο όσο και σε πολυπλοκότητα σε σχέση με την απευθείας αντιστροφή πίνακα.

Η μέθοδος των πεπερασμένων στοιχείων συνιστά μία άμεση και λογική επίλυση ενός προβλήματος που ακολουθεί μια καλά ορισμένη διαδικασία.



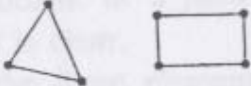

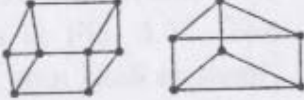
Η προσέγγιση καθώς και οι σχέσεις που χρησιμοποιούνται στο κεφάλαιο αυτό βρίσκονται σε αναλογία με το βιβλίο “Finite Element Analysis” του M J Fagan [8]

2.3 Διακριτοποίηση του προβλήματος

Το πρώτο βήμα που ακολουθεί η μέθοδος πεπερασμένων στοιχείων είναι ο χωρισμός του υπό μελέτη σώματος σε στοιχεία με τρόπο ώστε η αναπαράσταση να είναι αποδεκτή. Σε αυτό το τμήμα πρέπει να δοθεί ιδιαίτερη προσοχή, αφού η προετοιμασία του προβλήματος μπορεί να συμβάλει δραματικά στην εξοικονόμηση χρόνου και προσπάθειας τόσο στην υλοποίηση όσο και στην ανάλυση του μοντέλου. Παράγοντες όπως η συμμετρία αξίζει να συμπεριληφθούν και συνεπάγονται μεγάλη βελτίωση της αποδοτικότητας. Όπως θα δούμε παρακάτω, το μέγεθος του πίνακα $[K]$ τον οποίο θα κληθούμε να αντιστρέψουμε εξαρτάται άμεσα από το πλήθος των πεπερασμένων στοιχείων και η εξάρτηση είναι πολυωνυμικού βαθμού. Έτσι πρέπει να δοθεί ιδιαίτερη έμφαση στον ορισμό του μοντέλου και να υπάρχει διαφορετική αντιμετώπιση ανάλογα με τις ανάγκες της υλοποίησης. Όταν τα σφάλματα θέλουμε να είναι μικρά ή η μελέτη μας εστιάζεται σε μικρή περιοχή, αναγκαζόμαστε να χρησιμοποιήσουμε περισσότερα στοιχεία, αλλά όταν απαιτείται το σύστημα να λειτουργεί αποδοτικά πρέπει να περιοριστούμε σε λιγότερο ακριβείς δομές.

Όλα τα υλικά σώματα έχουν στην πραγματικότητα τρεις διαστάσεις. Ωστόσο, στις περισσότερες περιπτώσεις μπορούμε να κάνουμε κάποιες παραδοχές οι οποίες διευκολύνουν πολύ την προσέγγιση με τα πεπερασμένα στοιχεία. Έτσι για παράδειγμα αν θεωρήσουμε ότι η δύναμη που ασκείται σε ένα σώμα είναι κατά προσέγγιση σημειακή και ασκείται κατά μήκος του άξονα συμμετρίας μπορούμε να διευκολύνουμε κατά πολύ την ανάλυση. Εάν το πρόβλημα μπορεί να περιγραφεί πλήρως σε ένα επίπεδο τότε η μοντελοποίηση μπορεί να δίνει διδιάστατη αντί για τρισδιάστατη.

Τα βασικά στοιχεία, ανάλογα με τη γεωμετρία του προβλήματος φαίνονται στον παρακάτω πίνακα:

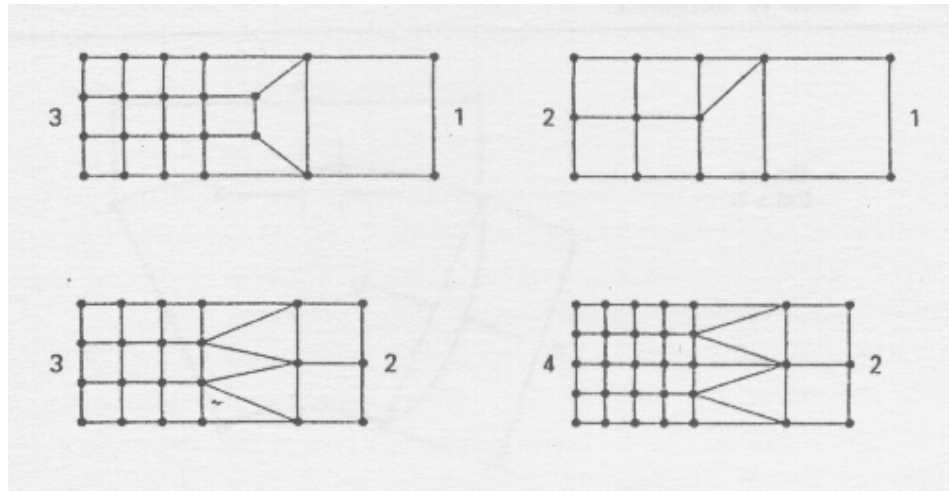
Σχήμα	Κατηγορία	Γεωμετρία
Σημείο	Μάζα (σημειακή)	
Γραμμή	Ράβδος, νήμα, δοκός, διάστημα	
Επιφάνεια	Διδιάστατο στερεό, Συμμετρικό ως προς άξονα σώμα, πλάκα	
Καμπύλη επιφάνεια	Κέλυφος	
Όγκος	Τρισδιάστατο Στερεό	

Πίνακας (2.1): Κατηγορίες πεπερασμένων στοιχείων [8]

Οι παραπάνω κατηγορίες μοντέλων ισχύουν τόσο για προβλήματα παραμόρφωσης όσο και για θερμικά. Ωστόσο, τα θερμικά προβλήματα είναι απλούστερα αφού η θερμότητα είναι μονοδιάστατο μέγεθος σε αντίθεση με τη μετατόπιση που γίνεται στο χώρο δηλαδή σε τρεις διαστάσεις. Προφανώς, όσο περισσότερες διαστάσεις περιλαμβάνει το μοντέλο τόσο πιο σύνθετη είναι η ανάλυσή του. Έτσι, όταν έχουμε μόνο ένα σημείο δε χρειάζεται να ορίσουμε συντεταγμένες, όταν θέλουμε να προσομοιώσουμε μία γραμμή έχουμε μία μόνο διάσταση, ενώ όταν πρόκειται για επιφάνεια ή όγκο θα χρειαζόμαστε δύο και τρεις διαστάσεις αντίστοιχα για την πλήρη περιγραφή της θέσης. Ειδική περίπτωση είναι αυτή της κυρτής επιφάνειας γιατί τα στοιχεία – elements εμφανίζουν και παραμόρφωση αντίστοιχη αυτής των δύο διαστάσεων και στρέψη, επομένως παρόλο που δεν πρόκειται για όγκο, οι δύο διαστάσεις δεν επαρκούν για την περιγραφή του. Εκμεταλλευόμενοι τις παραδοχές που μπορούμε να κάνουμε στο εκάστοτε πρόβλημα χρησιμοποιούμε το απλούστερο από τα παραπάνω μοντέλα.

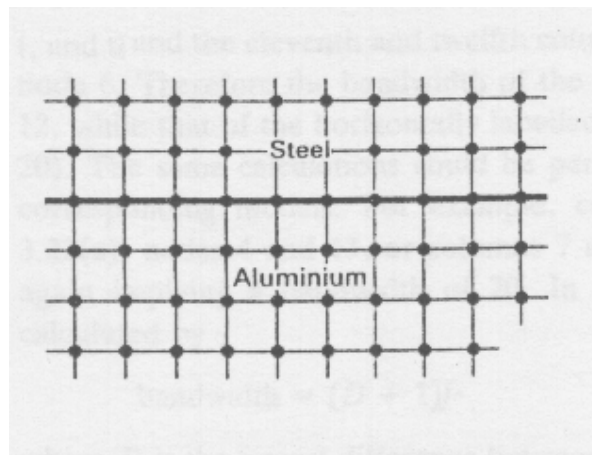
Αφού επιλέξουμε τον κατάλληλο τύπο πεπερασμένων στοιχείων, το επόμενο βήμα για την προσομοίωση ενός σώματος είναι η επιλογή του πλήθους και του μεγέθους των στοιχείων. Όσο μεγαλύτερο είναι το πλήθος, τόσο πιο μικρές είναι οι διαστάσεις και κατά συνέπεια τόσο μεγαλύτερη είναι η ακρίβεια αλλά και η πολυπλοκότητα. Στις περιπτώσεις που θέλουμε καλύτερη ακρίβεια σε ένα τμήμα ενός σώματος δε χρειάζεται να υλοποιήσουμε όλο το σώμα με γνώμονα την ακρίβεια αυτή. Μπορούμε

να διαφοροποιήσουμε την ακρίβεια χωρίς να έχουμε ασυνέχειες στο υλικό. Παρακάτω φαίνονται διάφοροι τρόποι να επιτευχθεί αυτό.



Σχήμα (2.7): Αλλαγή ακρίβειας με πεπερασμένα στοιχεία [8]

Ένα ακόμη στοιχείο στο οποίο πρέπει να επιστήσουμε την προσοχή μας είναι τα σημεία όπου οι παράμετροι του υλικού του σώματος μεταβάλλονται. Ένα πεπερασμένο στοιχείο δε μπορεί να ανήκει σε υλικά με διαφορετικές παραμέτρους. Έτσι απαιτείται μία διαχωριστική γραμμή ή επιφάνεια από κόμβους στο όριο διαφοροποίησης του υλικού. Ένα παράδειγμα φαίνεται στο παρακάτω σχήμα όπου έχουμε δύο διαφορετικές φάσεις σε ένα σώμα, μία από αλουμίνιο και μία από ασάλι.



Σχήμα (2.8): Αλλαγή υλικού με πεπερασμένα στοιχεία [8]

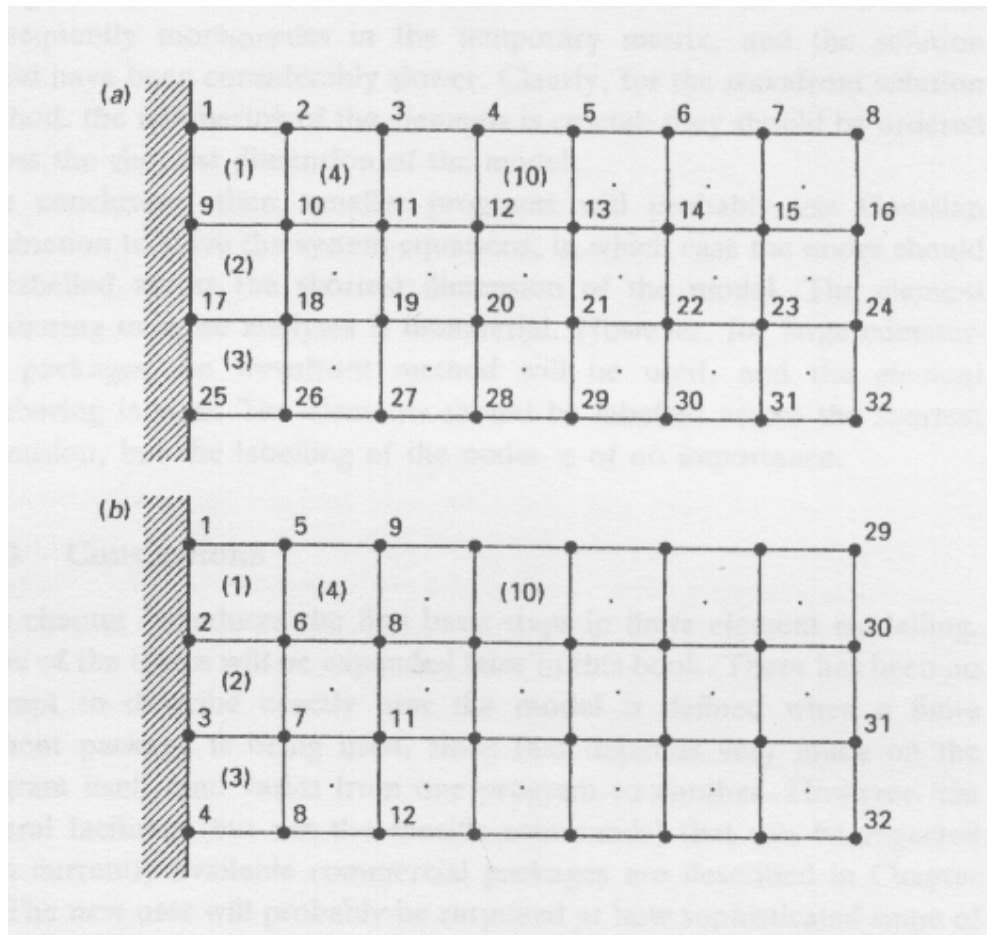
Αφού προσδιοριστούν οι εξισώσεις κάθε πεπερασμένου στοιχείου και συνδυαστούν για την εξαγωγή των εξισώσεων του συστήματος, απομένει η επίλυση του

προβλήματος. Για το σκοπό αυτό χρησιμοποιείται σε πολλές περιπτώσεις η απαλοιφή Gauss. Η μορφή του πίνακα $[K]$ είναι η παρακάτω:

$$\begin{array}{c}
 \text{Bandwidth} \\
 \left[\begin{array}{ccccccccccc}
 x & x & x & 0 & 0 & 0 & & & & & \\
 x & x & x & x & 0 & 0 & 0 & & & & \\
 x & x & x & x & x & 0 & 0 & 0 & & & \\
 0 & x & x & x & x & x & 0 & 0 & \square & \square & \square \\
 0 & 0 & x & x & x & x & x & 0 & & & \\
 0 & 0 & 0 & x & x & x & x & x & & & \\
 & 0 & 0 & 0 & x & x & x & x & & & \\
 & & & & \square & & & & & & \\
 & & & & \square & & & & & & \\
 & & & & \square & & & & & &
 \end{array} \right]
 \end{array}$$

Σχήμα (2.9): Μορφή πίνακα K [8]

Το μεγαλύτερο τμήμα του πίνακα αποτελείται από μηδενικά, εκτός από μία περιοχή με εύρος bandwidth γύρω από την κύρια διαγώνιο του πίνακα. Φυσικά, όσο μικρότερο είναι το εύρος αυτό τόσο πιο εύκολη και γρήγορη είναι η επίλυση του συστήματος. Αποδεικνύεται ότι το εύρος αυτό εξαρτάται αποκλειστικά από την αρίθμηση των στοιχείων. Το εύρος αυτό προκύπτει από το συνδυασμό των βαθμών ελευθερίας των κόμβων ενός πεπερασμένου στοιχείου. Επομένως, όσο πλησιέστεροι αριθμοί έχουν ανατεθεί στα στοιχεία αυτά τόσο μικρότερο θα είναι το εύρος.



Σχήμα (2.10): Αρίθμηση κόμβων σε μοντέλο [8]

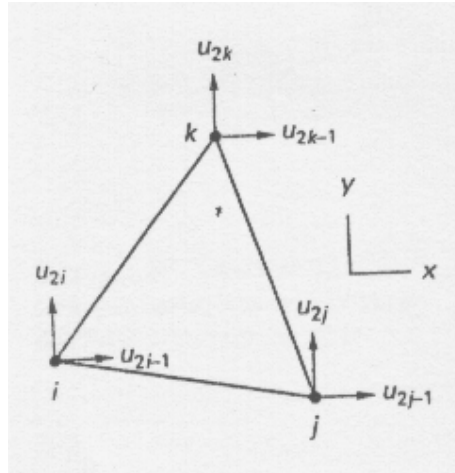
Στο παραπάνω σχήμα βλέπουμε δύο διαφορετικές μεθόδους αρίθμησης, για ένα μοντέλο παραμόρφωσης, την οριζόντια και την κατακόρυφη αρίθμηση. Στην πρώτη περίπτωση έχουμε συσχετισμό των κόμβων 1 έως 10, ενώ στη δεύτερη μόνο των κόμβων 1 έως 6. Επειδή το μοντέλο είναι διδιάστατο κάθε κόμβος αντιστοιχεί σε δύο βαθμούς ελευθερίας, απαιτούνται δύο συντεταγμένες για τον προσδιορισμό της θέσης του στο επίπεδο. Έτσι στην πρώτη περίπτωση το εύρος θα είναι 20 ενώ στη δεύτερη 12. Το ίδιο εύρος προκύπτει και από τα άλλα στοιχεία της δομής. Γενικότερα το εύρος, bandwidth μπορεί να υπολογιστεί από τη σχέση

$$bandwidth = (D+1)f \quad (2.19)$$

Με D τη μέγιστη διαφορά μεταξύ κόμβων και f το πλήθος των βαθμών ελευθερίας.

2.4 Προβλήματα παραμόρφωσης σε δύο διαστάσεις

Για το πρόβλημα που καλούμαστε να επιλύσουμε θα ακολουθήσουμε την ανάλυση σε δύο διαστάσεις εκμεταλλευόμενοι την αξονική συμμετρία του προβλήματος. Το πεπερασμένο στοιχείο στην περίπτωση αυτή είναι ένα τρίγωνο όπως αυτό του σχήματος:



Σχήμα (2.11): Τριγωνικό πεπερασμένο στοιχείο (βαθμοί ελευθερίας) [8]

Κάθε στοιχείο αποτελείται από τρεις κόμβους οι οποίοι αντιστοιχούν σε δύο βαθμούς ελευθερίας.

Εάν $\underline{u}(x,y)=[u,v]^T$ είναι το διάνυσμα της παραμόρφωσης για ένα σημείο σε ένα στοιχείο (e) τότε έχουμε:

$$\underline{u}(x,y) = \begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{N}^e(x,y) \cdot \underline{U}^e = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}^{(e)} \cdot \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{bmatrix}^{(e)} \quad (2.20)$$

Το διάνυσμα \underline{U}^e περιέχει έξι στοιχεία τα οποία αποτελούν τις μετατοπίσεις των δύο συντεταγμένων για καθέναν από τους τρεις κόμβους του τριγωνικού στοιχείου.

Στην περίπτωση που στο μοντέλο εξετάζουμε την παραμόρφωση μόνο λόγω πίεσης, δηλαδή δεν έχουμε στρέβλωση, οι πλευρές του πεπερασμένου στοιχείου λειτουργούν σαν ελατήρια και ο πίνακας ιδιοτήτων του υλικού (material property matrix) $[D^{(e)}]$ ορίζεται για κάθε στοιχείο ως εξής:

$$[D^{(e)}] = \frac{E}{1+\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \quad (2.21)$$

Στην παραπάνω σχέση E είναι ο δείκτης ελαστικότητας (elasticity modulus) του υλικού και ν ο συντελεστής Poisson. Ο πίνακας $[D^{(e)}]$ έχει σταθερή τιμή, η οποία είναι συνάρτηση μόνο του υλικού που προσομοιώνεται και της κατηγορίας του υπό εξέταση προβλήματος. Δεν υπάρχει καμία εξάρτηση από το σχήμα, το μέγεθος ή τη σύνδεση των στοιχείων, ούτε κατ' επέκταση από τις διαστάσεις ή το σχήμα ολόκληρου του σώματος.

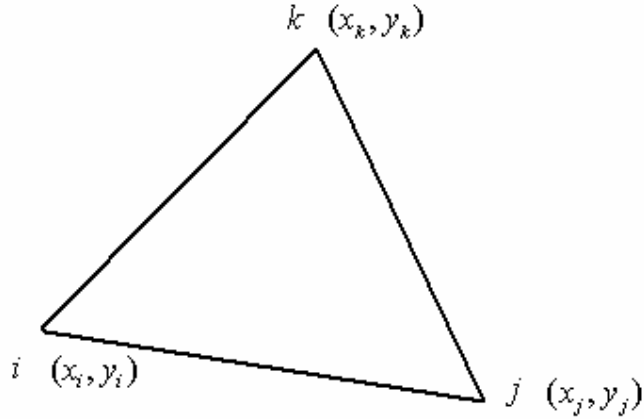
Για την εισαγωγή των παραμέτρων σχήματος και μεγέθους κάθε στοιχείου χρησιμοποιούμε τον πίνακα $[B^{(e)}]$ ο οποίος υπολογίζεται από τη σχέση:

$$[B^{(e)}] = \frac{1}{2A} \begin{bmatrix} b_i & 0 & b_j & 0 & b_k & 0 \\ 0 & c_i & 0 & c_j & 0 & c_k \\ c_i & b_i & c_j & b_j & c_k & b_k \end{bmatrix} \quad (2.22)$$

Στον παραπάνω πίνακα θεωρούμε τους κόμβους τους τριγώνου με τις αντίστοιχες συντεταγμένες τους:

ΚΟΜΒΟΣ ΣΥΝΤΕΤΑΓΜΕΝΕΣ	
i	(x_i, y_i)
j	(x_j, y_j)
k	(x_k, y_k)

Πίνακας (2.2): Συντεταγμένες κόμβων



Σχήμα (2.12): Τριγωνικό πεπερασμένο στοιχείο (Συντεταγμένες)

Ο συντελεστής A αντιπροσωπεύει το εμβαδό του τριγωνικού element και υπολογίζεται από τη σχέση:

$$A = \frac{1}{2} \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} \quad (2.23)$$

Ενώ τα $b_i, b_j, b_k, c_i, c_j, c_k$ προσδιορίζονται από τις σχέσεις:

$$\begin{aligned} a_i &= x_j y_k - x_k y_j & b_i &= y_j - y_k & c_i &= x_k - x_j \\ a_j &= x_k y_i - x_i y_k & b_j &= y_k - y_i & c_j &= x_i - x_k \\ a_k &= x_i y_j - x_j y_i & b_k &= y_i - y_j & c_k &= x_j - x_i \end{aligned} \quad (2.24)$$

Στις τελευταίες σχέσεις παρατηρούμε ότι τα $b_i, b_j, b_k, c_i, c_j, c_k$, προκύπτουν από τις σχετικές συντεταγμένες δύο κόμβων του πεπερασμένου στοιχείου. Συνεπώς, είναι ανεξάρτητα της θέσης του element στο χώρο. Το ίδιο φυσικά ισχύει και για το εμβαδό A . Το συμπέρασμα είναι ότι ο πίνακας $[B^{(e)}]$ είναι ίδιος για στοιχεία ίδιου σχήματος και μεγέθους. Με τον όρο σχήμα εννοούμε και τον προσανατολισμό, δηλαδή τα τριγωνικά elements πρέπει να είναι ακριβή είδωλα με μόνη διαφορά τις συντεταγμένες τους στο χώρο.

Σημειώνεται ότι για κάθε στοιχείο (e) του μοντέλου, η διάταση ($\underline{\varepsilon}$) υπολογίζεται από τη σχέση:

$$\underline{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \end{bmatrix} = [\mathbf{B}^e] \cdot \underline{U}^e \quad (2.25)$$

Έχοντας υπολογίσει τους πίνακες $[B^{(e)}]$ και $[D^{(e)}]$ μπορούμε να υπολογίσουμε τον πίνακα $[K^{(e)}]$ (τον stiffness matrix) του στοιχείου:

$$\begin{aligned} [K^{(e)}] &= \int_V [B^{(e)}]^T [D^{(e)}] [B^{(e)}] dV \\ &= [B^{(e)}]^T [D^{(e)}] [B^{(e)}] \int_V dV \\ &= [B^{(e)}]^T [D^{(e)}] [B^{(e)}] tA \end{aligned} \quad (2.26)$$

Στην τελευταία σχέση η μεταβλητή t αποτελεί μία σταθερά που εκφράζει το πάχος του πεπερασμένου στοιχείου.

Ο πίνακας $[K^{(e)}]$ που υπολογίσαμε αφορά μόνο ένα στοιχείο, εξ' ου και ο εκθέτης (e). Για τον υπολογισμό του συνολικού πίνακα $[K]$ (stiffness matrix) για όλο το σώμα, πρέπει να αθροίσουμε κατάλληλα τους επιμέρους πίνακες. Ο πίνακας $[K^{(e)}]$ για ένα στοιχείο αντιστοιχεί σε 6 βαθμούς ελευθερίας. Ο συνολικός πίνακας $[K]$ αντιστοιχίζεται σε όλους τους βαθμούς ελευθερίας (δύο για κάθε κόμβο). Για να συνυπολογίσουμε τους επιμέρους πίνακες των στοιχείων στον προσδιορισμό του συνολικού πίνακα χρειαζόμαστε μία επιπλέον πληροφορία ώστε οι τιμές τους να τοποθετηθούν στη σωστή θέση. Αυτό επιτυγχάνεται με έναν επιπλέον πίνακα, ο οποίος ορίζεται ως εξής:

$$U^{(e)} = \begin{Bmatrix} u_{2i-1} \\ u_{2i} \\ u_{2j-1} \\ u_{2j} \\ u_{2k-1} \\ u_{2k} \end{Bmatrix} \quad (2.27)$$

Έχοντας κατασκευάσει τον stiffness matrix, πίνακα $[K]$ του συστήματος, απομένει να υπολογίσουμε και τον πίνακα \underline{F} ώστε να μπορέσουμε να λύσουμε το σύστημα:

$$[K] \cdot \underline{U} = \underline{F} \quad (2.28)$$

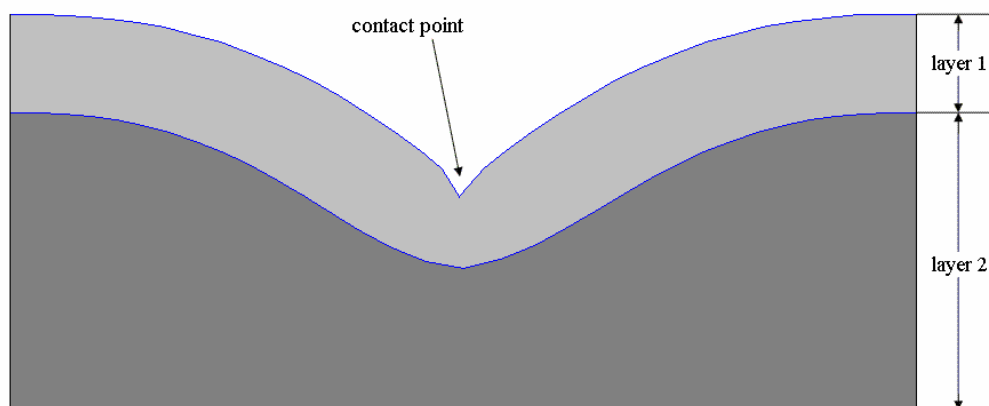
Στο τελευταίο σύστημα έχουμε άγνωστο τον πίνακα \underline{U} το διάνυσμα των μετατοπίσεων των κόμβων. Στον υπολογισμό του πίνακα \underline{F} , που αποτελεί και το διάνυσμα των ασκούμενων δυνάμεων, αρκεί να συνυπολογίσουμε τις εφαρμοζόμενες εξωτερικές δυνάμεις σε κάθε κόμβο (σε κάθε βαθμό ελευθερίας συγκεκριμένα), καθώς και τις δυνάμεις που ασκούνται από τα όρια-τοιχώματα που έχουμε θέσει.

Κεφάλαιο 3

Προσδιορισμός παραμέτρων διδιάστατου μοντέλου

3.1 Δημιουργία μοντέλου με χρήση των πεπερασμένων στοιχείων

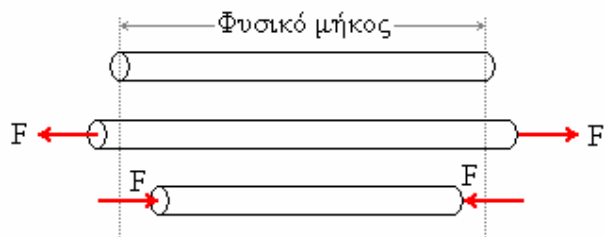
Η παραμόρφωση του δέρματος (αν θεωρήσουμε ότι τα οστά αποτελούνται από σταθερά σημεία) εξαρτάται από την ελαστικότητα δύο στρωμάτων. Αυτό της επιδερμίδας και αυτό του υποδόριου ιστού (λίπους). Επομένως, με τη χρήση των πεπερασμένων στοιχείων μπορεί να προσομοιωθεί με δύο στρώματα διαφορετικού πάχους και διαφορετικής ελαστικότητας. Η καμπύλη που προκύπτει και αποτελεί την επιφάνεια αυτής της δομής, με δεδομένο το βάθος διείσδυσης του σημειακού φορτίου, εξαρτάται από τη σχετική ελαστικότητα καθώς και το σχετικό πάχος των δύο στρωμάτων. Σε αυτό το κεφάλαιο θα προσδιορίσουμε τις περιοχές τιμών των δύο αυτών λόγων για τις οποίες η επιφάνεια αντιδρά με τρόπο παρόμοιο με αυτόν που προκύπτει από τα πειραματικά δεδομένα.



Σχήμα (3.1): Μοντέλο με δύο στρώματα διαφορετικού πάχους και ελαστικότητας.

Για το στόχο αυτό χρησιμοποιήθηκε κώδικας σε matlab όπου μπορούμε εύκολα να μετρήσουμε τα σφάλματα και να αναπαραστήσουμε σχηματικά μία τομή του μοντέλου σε δύο διαστάσεις. Το δομικό στοιχείο του διδιάστατου μοντέλου μας είναι το απλό γραμμικό τριγωνικό πεπερασμένο στοιχείο. Η φυσική ερμηνεία του στοιχείου αυτού αναλύεται παρακάτω.

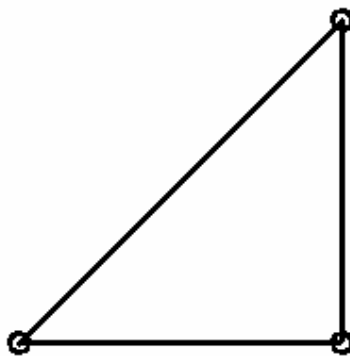
Το απλό γραμμικό τριγωνικό στοιχείο που χρησιμοποιούμε είναι ουσιαστικά ένα ορθογώνιο ισοσκελές τρίγωνο το οποίο σχηματίζεται από τρεις ελαστικές ράβδους. Η ελαστικότητα των ράβδων εμφανίζεται μόνο κατά τον άξονά τους, δεν υπάρχει δηλαδή στρέβλωση αλλά μόνο μεταβολή του μήκους κάθε ράβδου όταν στο τρίγωνο ασκηθεί κάποια δύναμη. Τα άκρα των ράβδων είναι ενωμένα σε σημεία τα οποία αποτελούν και τους κόμβους του μοντέλου.



Σχήμα (3.2): Ελαστικές ράβδους (παραμόρφωση μόνο ως προς το μήκος)

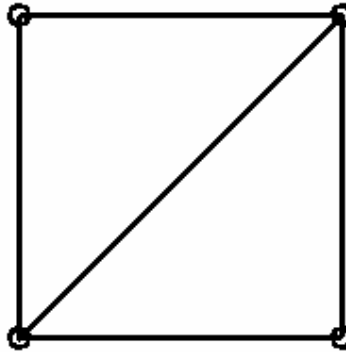


Σχήμα (3.3): Σχηματική αναπαράσταση ράβδου



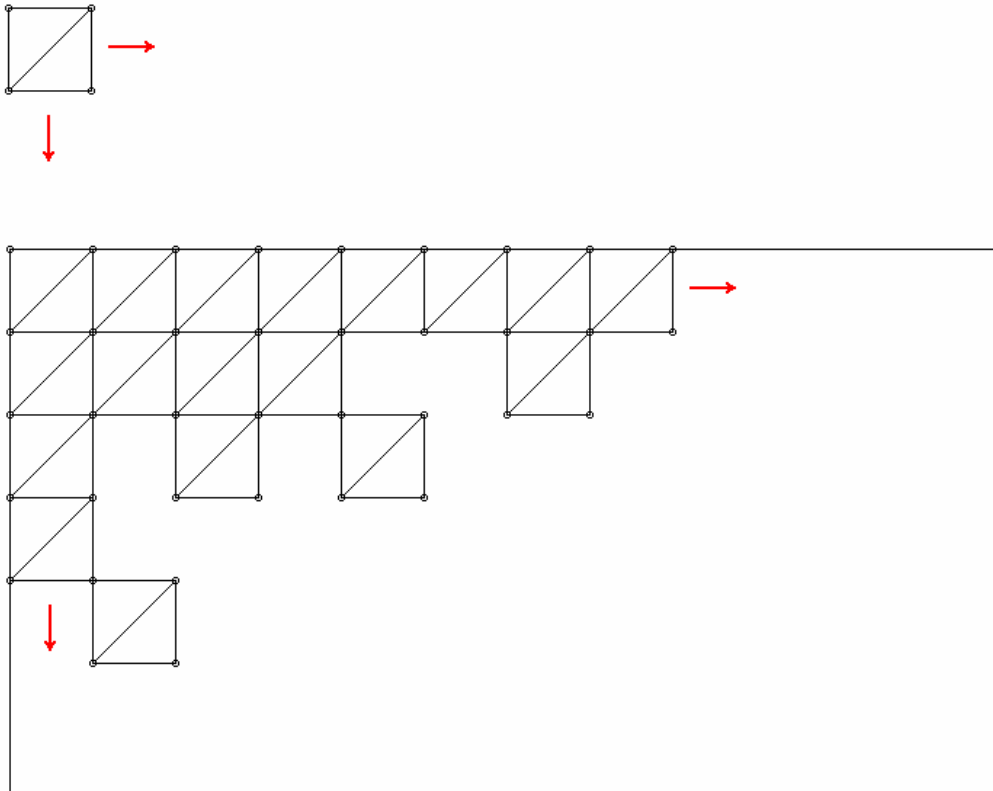
Σχήμα (3.4): Σχηματική αναπαράσταση τριγωνικού πεπερασμένου στοιχείου

Τοποθετώντας δύο τέτοια τρίγωνα μαζί παίρνουμε ένα τετράγωνο που αποτελείται από τέσσερις κόμβους.



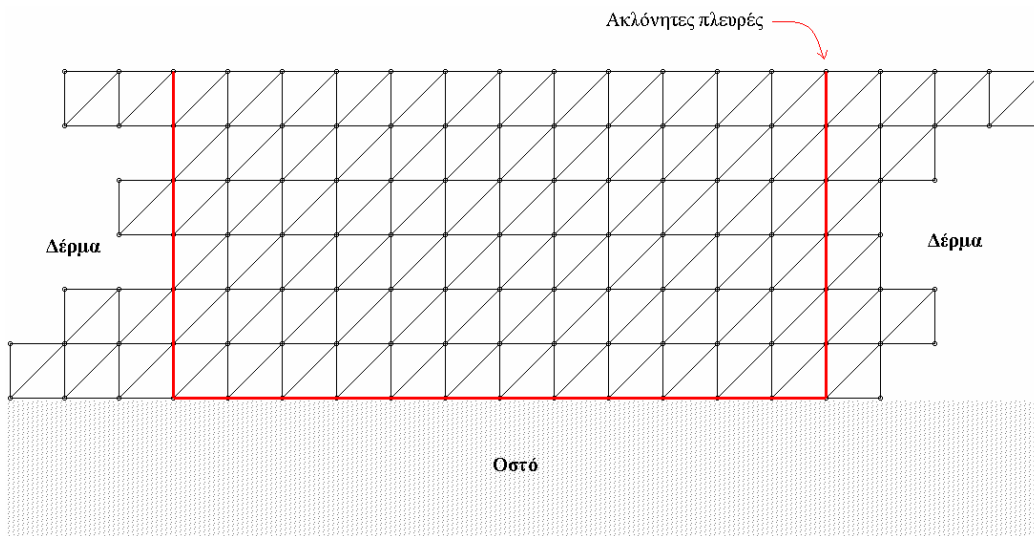
Σχήμα (3.5): Ελαστικό τετράγωνο από πεπερασμένα στοιχεία

Επαναλαμβάνοντας πολλές φορές τη δομή αυτή και προς τις δύο κατευθύνσεις μπορούμε να κατασκευάσουμε ένα παραλληλόγραμμο το οποίο θα αποτελεί και μια επίπεδη τομή ενός τμήματος του δέρματος.



Σχήμα (3.6): Δημιουργία επιφάνειας από ελαστικό υλικό με χρήση πεπερασμένων στοιχείων

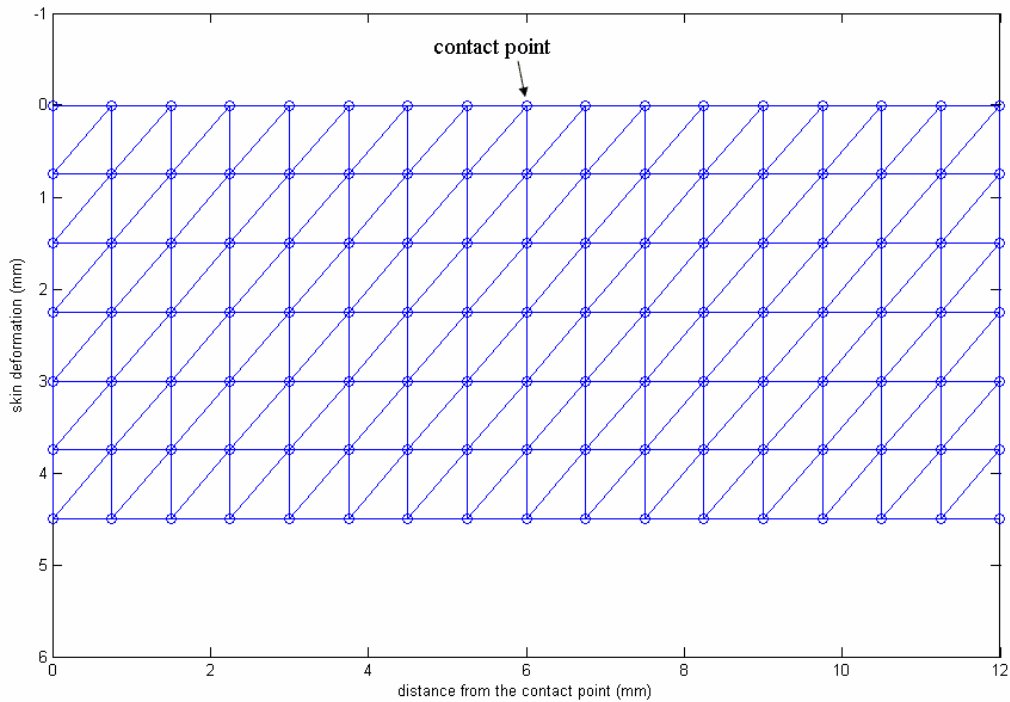
Θεωρώντας ότι κάτω από την επιφάνεια του δέρματος βρίσκεται οστό και ότι δεξιά και αριστερά από τη δομή μας το δέρμα συνεχίζεται μπορούμε να εκλάβουμε ότι η κάτω έδρα του παραλληλογράμμου καθώς και οι δύο πλευρές δε μετακινούνται. Έτσι, θέτουμε τους αντίστοιχους κόμβους ως ακλόνητα σημεία.



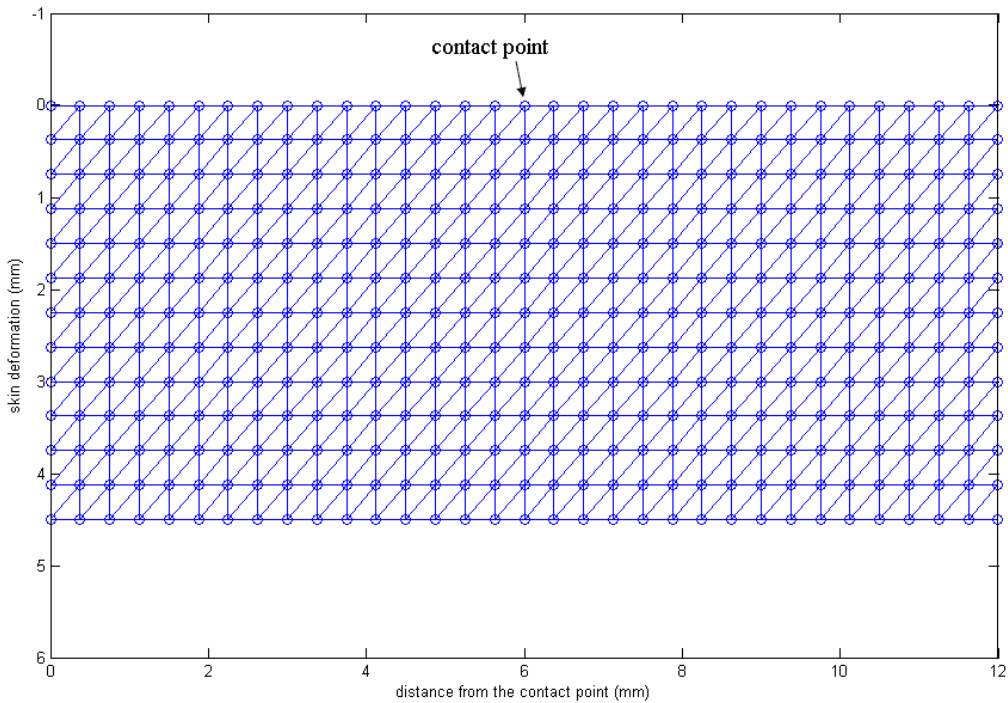
Σχήμα (3.7): Ορισμός σταθερών σημείων για το μοντέλο

Τέλος αρκεί να ορίσουμε το μέγεθος του παραλληλογράμμου και το πλήθος των πεπερασμένων στοιχείων που θα χρησιμοποιήσουμε. Για να υπάρχει αναλογία με τις πειραματικές τιμές θεωρούμε ότι το πάχος του δέρματος από την επιφάνεια μέχρι το οστό είναι 4,5mm ενώ ένα καλό μήκος είναι τα 12mm με το σημείο επαφής στο κέντρο. Μήκος μεγαλύτερο από αυτό δεν είναι απαραίτητο αφού μετά την ακτίνα των 6mm θεωρούμε ότι τόσο η επιφάνεια όσο και το εσωτερικό του δέρματος δε δέχεται καμία παραμόρφωση. Τα δύο μήκη αυτά θα πρέπει να τα μοιράσουμε σε ελαστικά τετράγωνα, οπότε την αναλογία αυτή σε χιλιοστά θα πρέπει να τη διατηρήσουμε και για την αναλογία των τετραγώνων οριζόντια και κατακόρυφα. Έτσι εάν κατακόρυφα χρησιμοποιήσουμε 3 τετράγωνα τότε οριζόντια θα χρησιμοποιήσουμε 8 τετράγωνα. Αντίστοιχα, μπορούμε να χρησιμοποιήσουμε 6 τετράγωνα κατακόρυφα και 16 οριζόντια ή 12 κατακόρυφα, 32 οριζόντια, ανάλογα με την ακρίβεια που θέλουμε να επιτύχουμε. Φυσικά, όσο περισσότερα τα τετράγωνα μας τόσο καλύτερη και η ακρίβεια, αλλά τόσο μεγαλύτερη και η πολυπλοκότητα του προβλήματος. Επειδή το μοντέλο αποτελείται από δύο διαστάσεις, εάν διπλασιάσουμε την ακρίβεια, δηλαδή το πλήθος των τετραγώνων ως προς κάθε διάσταση το πλήθος των πεπερασμένων στοιχείων τετραπλασιάζεται. Το ίδιο και οι αντίστοιχοι πίνακες που το αναπαριστούν. Το πρόβλημα τότε γίνεται εξαιρετικά πολύπλοκο.

Μία καλή επιλογή πλήθους πεπερασμένων στοιχείων για το μοντέλο μας προκύπτει με 16 τετράγωνα οριζόντια και 6 τετράγωνα κατακόρυφα. Με τη δομή αυτή θα έχουμε $16 \times 6 = 96$ ελαστικά τετράγωνα τα οποία θα αποτελούνται από $17 \times 7 = 119$ κόμβους. Κάθε κόμβος με τη σειρά του ορίζει 2 βαθμούς ελευθερίας, οπότε για την περιγραφή του συστήματος ο συνολικός πίνακας K (stiffness matrix) θα έχει μέγεθος $(2 \times 119) \times (2 \times 119) = 56644$ θέσεων. Για να έχουμε καλή ακρίβεια στους βαθμούς ελευθερίας θα πρέπει να χρησιμοποιήσουμε τουλάχιστον 2bytes ανά βαθμό ελευθερίας. Έτσι ο πίνακας K έχει μέγεθος τουλάχιστον 113Kbytes. Διπλασιάζοντας την ακρίβεια ο αντίστοιχος πίνακας πλησιάζει τα 0,5Mbytes και η υλοποίηση καθίσταται πολύ χρονοβόρα. Ωστόσο, δεν έχουμε καμία ουσιαστική βελτίωση στα αποτελέσματα όπως θα δείξουμε παρακάτω. Το μοντέλο που χρησιμοποιούμε φαίνεται στο σχήμα που ακολουθεί.



Σχήμα (3.8): Διδιάστατο μοντέλο πριν την παραμόρφωση



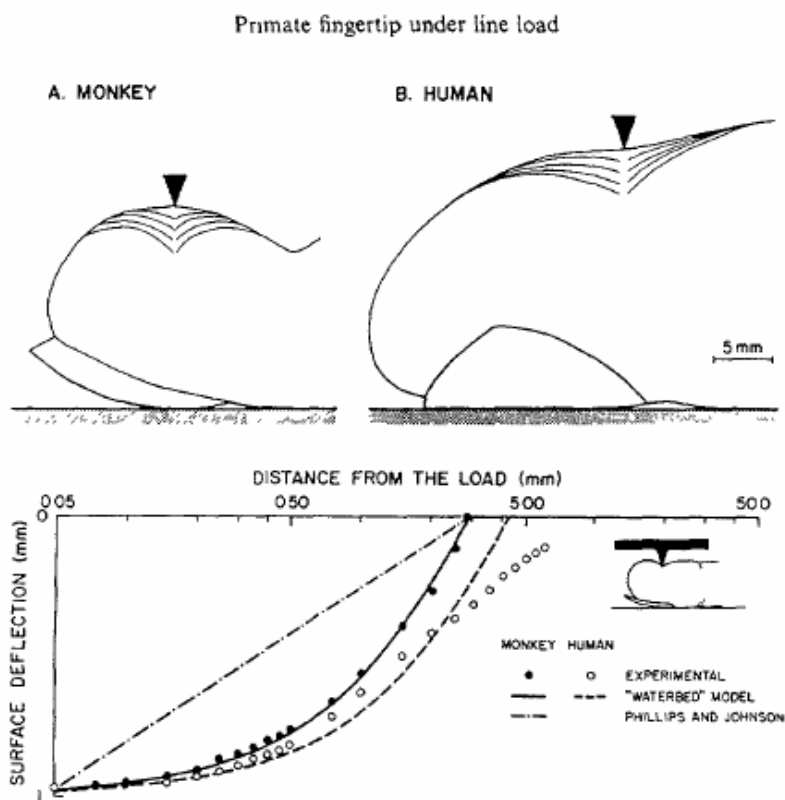
Σχήμα (3.9): Διδιάστατο μοντέλο μεγάλης ακρίβειας

Στο σύστημα αυτό, το σημείο επαφής (contact point) είναι στο κέντρο της επάνω επιφάνειας ενώ ο λόγος πάχους που μπορούν να έχουν τα δύο στρώματα διαφορετικής ελαστικότητας είναι $1/5$, $2/4$, $3/3$, $4/2$, $5/1$ ανάλογα με το πώς θα κατανείμουμε τα 6 τετράγωνα που έχουμε κατακόρυφα. Εάν χρησιμοποιήσουμε μοντέλο μεγαλύτερης ακρίβειας θα έχουμε περισσότερες επιλογές ως προς την κατανομή των δύο στρωμάτων. Σε κάθε περίπτωση όμως η αλλαγή υλικού θα πρέπει να γίνεται σε διαχωριστική επιφάνεια, όπως ορίζεται από τους κόμβους του μοντέλου, αφού ένα πεπερασμένο στοιχείο (ένα τρίγωνο) δε μπορεί να αποτελείται παρά από ένα υλικό. Οι συντεταγμένες των κόμβων στην κάτω και στις πλευρικές επιφάνειες είναι σταθερές αφού τα αντίστοιχα σημεία είναι ακλόνητα. Αφού δημιουργήσουμε τους αντίστοιχους πίνακες μπορούμε να επιλύσουμε το σύστημα θέτοντας το βάθος διείσδυσης του σημείου επαφής, ακολουθώντας τη διαδικασία που ορίσαμε στο δεύτερο κεφάλαιο. Σκοπός μας είναι να βρούμε τον ιδανικό λόγο πάχους και ελαστικότητας των δύο στρωμάτων. Επομένως, στον κώδικά μας δοκιμάζουμε επαναληπτικά διάφορες τιμές των λόγων αυτών και μετρούμε το εκάστοτε σφάλμα ώστε να το ελαχιστοποιήσουμε.

3.2 Πειραματικά δεδομένα

Τα πειραματικά δεδομένα που έχουμε στη διάθεσή μας, πάνω στα οποία καλούμαστε να προσαρμόσουμε το μοντέλο μας έχουν ληφθεί από μετρήσεις οι οποίες περιγράφονται σε μελέτες δημοσιευμένες στη διεθνή βιβλιογραφία [9], και αποτελούνται από δύο ομάδες τιμών.

Οι τιμές αυτές, όπως περιγράφεται στο [9], προέκυψαν με την παρακάτω διαδικασία. Επιλέχθηκε αρχικά ένα σημείο επαφής ως κέντρο για το οποίο μπορούμε να θεωρήσουμε ότι για μία ακτίνα 6 χιλιοστών το δέρμα είναι περίπου επίπεδο. Το σημείο αυτό φαίνεται στο παρακάτω σχήμα.



Σχήμα (3.10): Μέτρηση της παραμόρφωσης της επιφάνειας του δέρματος [9]

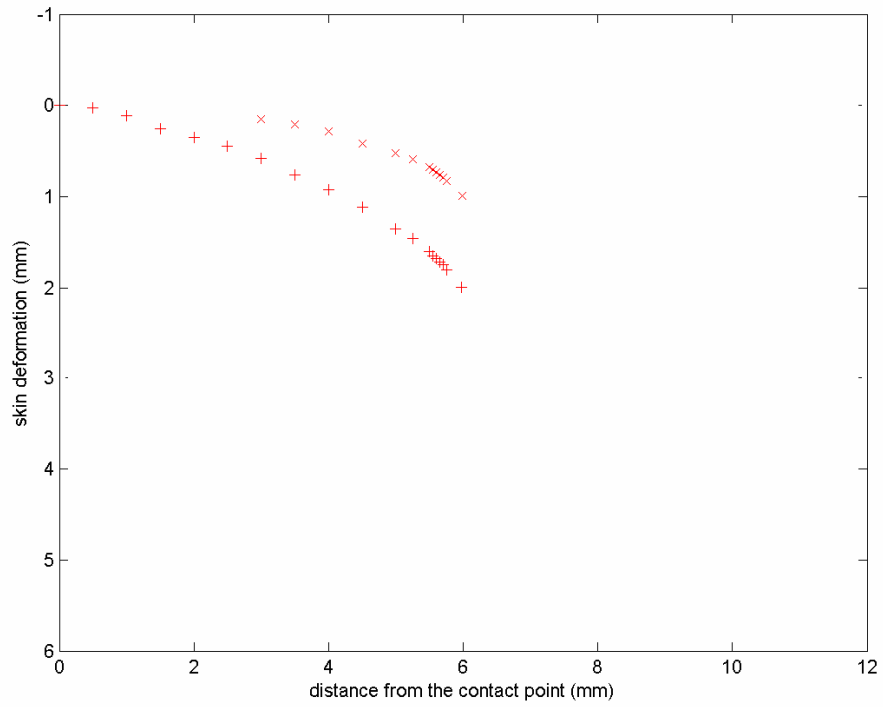
Στη συνέχεια επιλέχθηκε ένα βάθος διείσδυσης ενός σημειακού φορτίου το οποίο εφαρμόστηκε στο σημείο αυτό. (Το φορτίο αυτό βέβαια είναι κατά προσέγγιση σημειακό). Κατόπιν, και για το βάθος αυτό καταγράφηκε το βάθος διείσδυσης του δέρματος σε διάφορες αποστάσεις από το κέντρο (ακτινικά). Από τις μετρήσεις προέκυψε μία καμπύλη που παρουσιάζει την παραμόρφωση της επιφάνειας του δέρματος για το συγκεκριμένο βάθος διείσδυσης του σημείου επαφής. Εκτελώντας τη διαδικασία αυτή δύο φορές λαμβάνουμε δύο ομάδες τιμών για βάθος 2,0mm και 1,0mm αντίστοιχα. Οι τιμές αυτές φαίνονται στους παρακάτω πίνακες.

1 ^η ΟΜΑΔΑ ΤΙΜΩΝ (Βάθος διείσδυσης σημείου επαφής 2mm)	
Απόσταση από το σημείο επαφής (mm)	Βάθος διείσδυσης (mm)
0,25	1,80
0,30	1,75
0,35	1,72
0,40	1,68
0,45	1,65
0,50	1,60
0,75	1,46
1,00	1,35
1,50	1,12
2,00	0,93
2,50	0,76
3,00	0,58
3,50	0,45
4,00	0,35
4,50	0,25
5,00	0,11
5,50	0,03
6,00	0,00

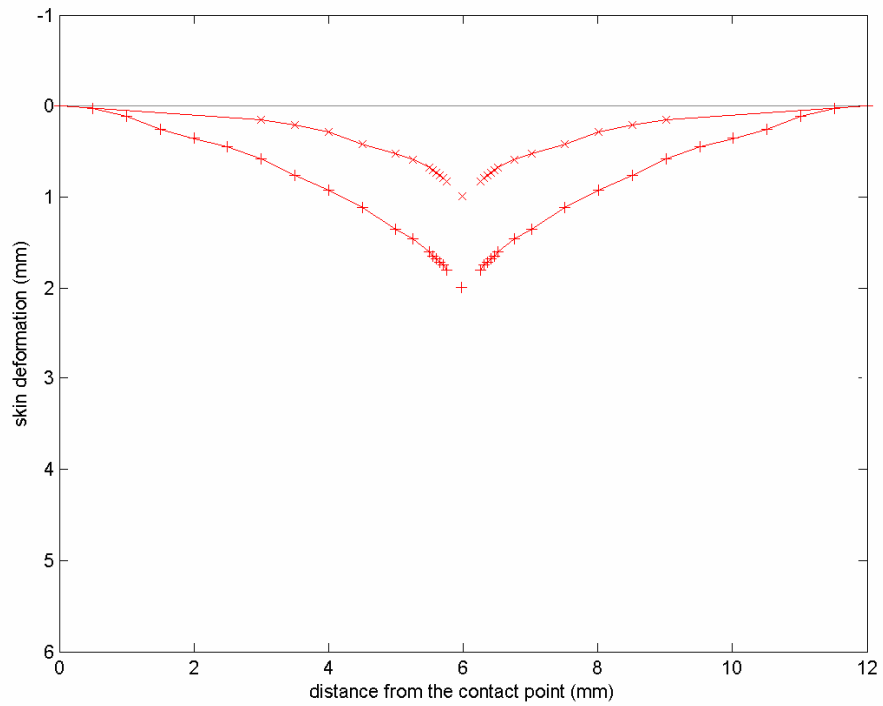
Πίνακας (3.1): 1^η ομάδα πειραματικών δεδομένων (set1) για βάθος 2mm

2 ^η ΟΜΑΔΑ ΤΙΜΩΝ (Βάθος διείσδυσης σημείου επαφής 1mm)	
Απόσταση από το σημείο επαφής (mm)	Βάθος διείσδυσης (mm)
0,25	0,83
0,30	0,79
0,35	0,76
0,40	0,73
0,45	0,70
0,50	0,68
0,75	0,59
1,00	0,52
1,50	0,42
2,00	0,28
2,50	0,21
3,00	0,15

Πίνακας (3.2): 2^η ομάδα πειραματικών δεδομένων (set2) για βάθος 1mm



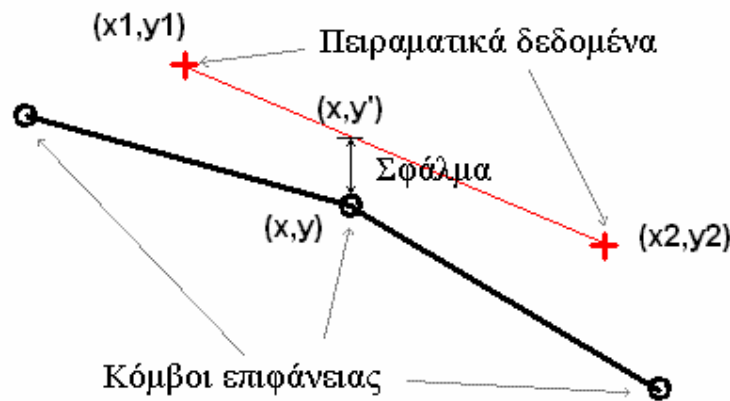
Σχήμα (3.11): Πειραματικά δεδομένα



Σχήμα (3.12): Καμπύλες σύμφωνα με τα πειραματικά δεδομένα

3.3 Γραμμική παρεμβολή - Μέσο τετραγωνικό σφάλμα

Αφού ορίσαμε τον τρόπο κατασκευής του μοντέλου για δεδομένο βάθος διείσδυσης του σημείου επαφής μπορούμε να προσδιορίσουμε το σφάλμα μεταξύ της κατανομής της επιφάνειας του μοντέλου και των πειραματικών δεδομένων. Για να υπολογίσουμε το σφάλμα αυτό αρκεί να μετρήσουμε πρώτα την απόκλιση των κόμβων της επιφάνειας από τη θεωρητική τους θέση, σύμφωνα με τα πειραματικά δεδομένα. Αυτό μπορούμε εύκολα να το πετύχουμε με χρήση της γραμμικής παρεμβολής. Λαμβάνοντας ότι η κίνηση κατά τον οριζόντιο άξονα είναι πολύ μικρότερη σε σχέση με αυτή στον κατακόρυφο, μπορούμε να θεωρήσουμε ότι το σφάλμα βρίσκεται μόνο κατά τον κατακόρυφο άξονα.



Σχήμα (3.13): Γραμμική παρεμβολή για τον υπολογισμό του κατακόρυφου σφάλματος

Έτσι, για να προσδιορίσουμε το σφάλμα που εμφανίζει ένας κόμβος (x, y) της επιφάνειας του μοντέλου με δεδομένες τις θέσεις (x_1, y_1) και (x_2, y_2) δύο γειτονικών πειραματικών σημείων θα πρέπει να υπολογίσουμε τη θεωρητική θέση του σημείου αυτού (x, y') . Για τη θέση αυτή χρειαζόμαστε μόνο τη νέα τεταγμένη (y') την οποία θα υπολογίσουμε εύκολα αφού τα σημεία (x_1, y_1) , (x, y') και (x_2, y_2) είναι συνευθειακά. Έτσι έχουμε:

$$\frac{y' - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} \quad (3.1)$$

Από την τελευταία σχέση προκύπτει:

$$y' = y_1 + \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1) \quad (3.2)$$

Το σφάλμα επομένως για ένα κόμβο είναι:

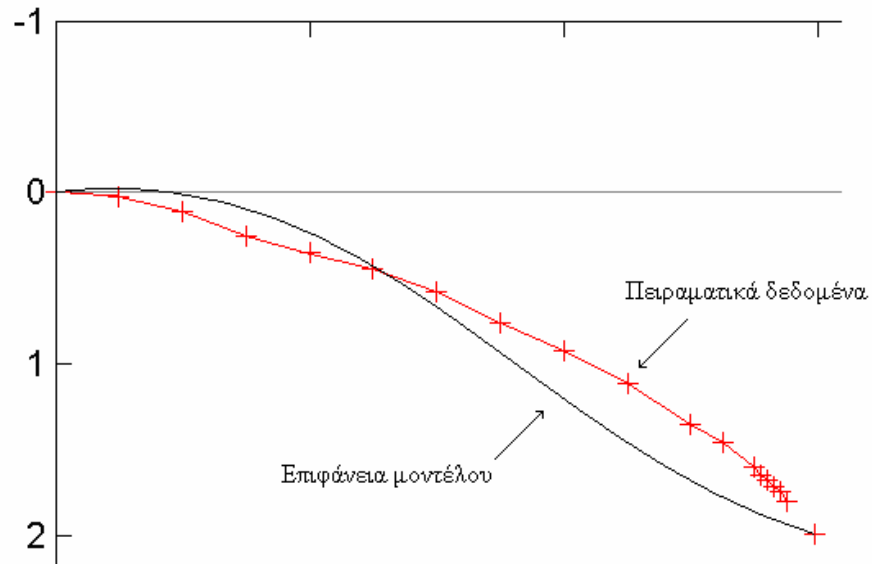
$$error = |y - y'| = \left| y - \left(y_1 + \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1) \right) \right| \quad (3.3)$$

Για να προσαρμόσουμε το μοντέλο ώστε να ανταποκρίνεται στα πειραματικά δεδομένα προσπαθούμε ώστε η απόκλιση της κατανομής που ακολουθεί η επιφάνεια να είναι η ελάχιστη δυνατή. Το σφάλμα που μετράμε ορίζεται ως το μέσο τετραγωνικό σφάλμα (σε χιλιοστά) της απόστασης διείσδυσης των σημείων της επιφάνειας σε σχέση με τα πειραματικά στοιχεία. Από το σύνολο των σφαλμάτων που εμφανίζουν οι κόμβοι πλήθους (n) της επιφάνειας του μοντέλου υπολογίζουμε το μέσο τετραγωνικό σφάλμα σύμφωνα με τη σχέση:

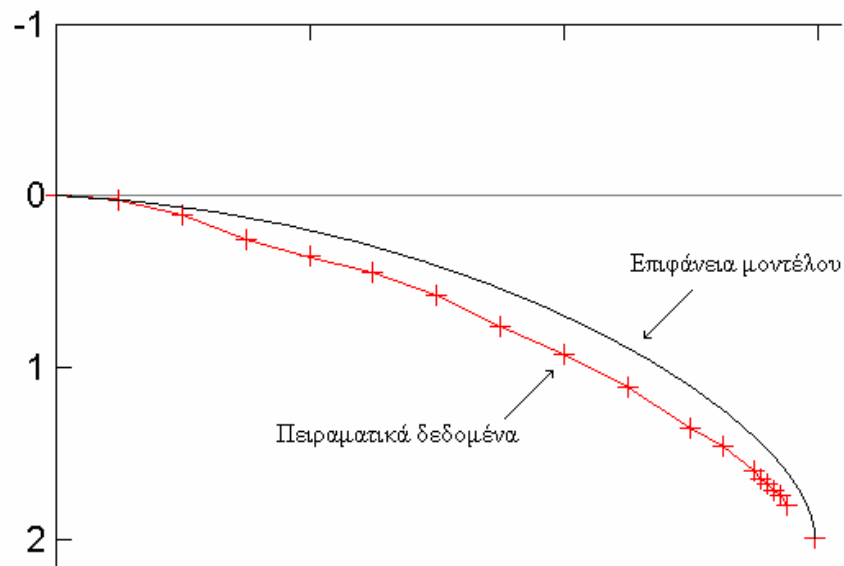
$$E = \frac{\sum_{i=1}^n error_i^2}{n} \quad (3.4)$$

3.4 Διόρθωση βάθους διείσδυσης

Εξετάζοντας τις επιφάνειες που προκύπτουν από τους ανώτερους κόμβους των πεπερασμένων στοιχείων για το εκάστοτε μοντέλο βλέπουμε ότι έχουμε δύο μορφών καμπύλες.



Σχήμα (3.14α): Καμπύλη πρώτης μορφής που προκύπτει από την επιφάνεια της δομής των πεπερασμένων στοιχείων.

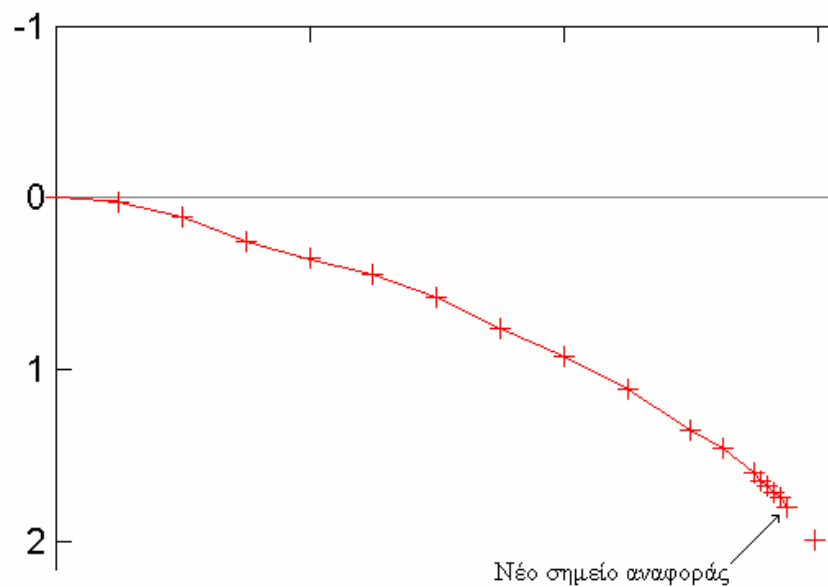


Σχήμα (3.14β): Καμπύλη δεύτερης μορφής που προκύπτει από την επιφάνεια της δομής των πεπερασμένων στοιχείων.

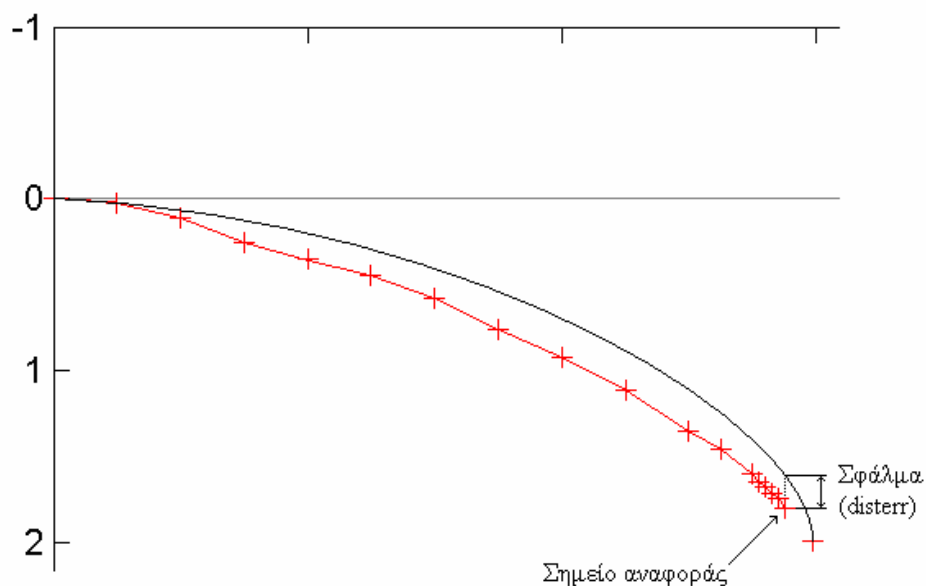
Παρατηρώντας τα αποτελέσματα των μετρήσεων έχουμε τις καμπύλες της μορφής του σχήματος 3.13α να εμφανίζουν μικρότερο μέσο τετραγωνικό σφάλμα από καμπύλες της μορφής του σχήματος 3.13β. Ωστόσο, οι τελευταίες είναι προτιμότερες, αφού το αποτέλεσμα είναι πολύ πιο ρεαλιστικό. Για το λόγο αυτό θα αλλάξουμε την προσέγγιση ώστε το μέσο τετραγωνικό σφάλμα στις καμπύλες της δεύτερης περίπτωσης να μειωθεί.

Στις μετρήσεις που προέκυψαν πειραματικά μπορούμε να διαπιστώσουμε ότι κοντά στο κέντρο όπου βρίσκεται το σημείο επαφής δεν έχουμε τιμές. Αυτό συμβαίνει επειδή πολύ κοντά στο σημείο επαφής δε μπορούμε να πάρουμε μετρήσεις. Έτσι σε απόσταση μικρότερη των 0,25mm από το κέντρο τα πειραματικά δεδομένα δε μας παρέχουν σωστή πληροφορία.

Για το λόγο αυτό θα προσαρμόσουμε το μοντέλο στο πρώτο έγκυρο σημείο των πειραματικών δεδομένων. Επομένως, ως σημείο αναφοράς δε θα έχουμε το σημείο επαφής, αλλά θα επιλέξουμε νέο σημείο αναφοράς το σημείο σε απόσταση 0,25mm. Για δεδομένες παραμέτρους του μοντέλου θα προσαρμόσουμε την επιφάνεια των πεπερασμένων στοιχείων ώστε να περνά από το σημείο αυτό. Για να το πετύχουμε αυτό αρκεί να μετατοπίσουμε το σημείο διείδυσης.

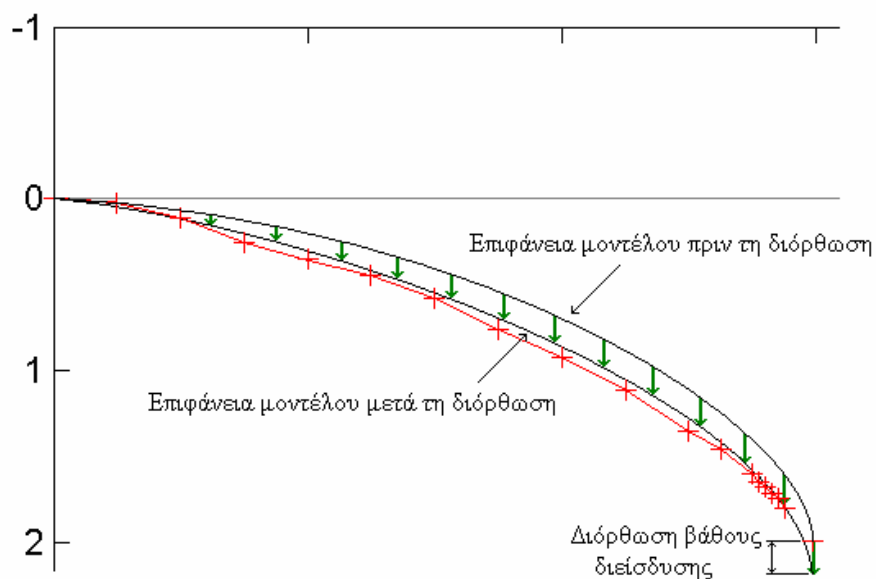


Σχήμα (3.15α): Νέο σημείο αναφοράς σε απόσταση 0,25mm από το σημείο επαφής



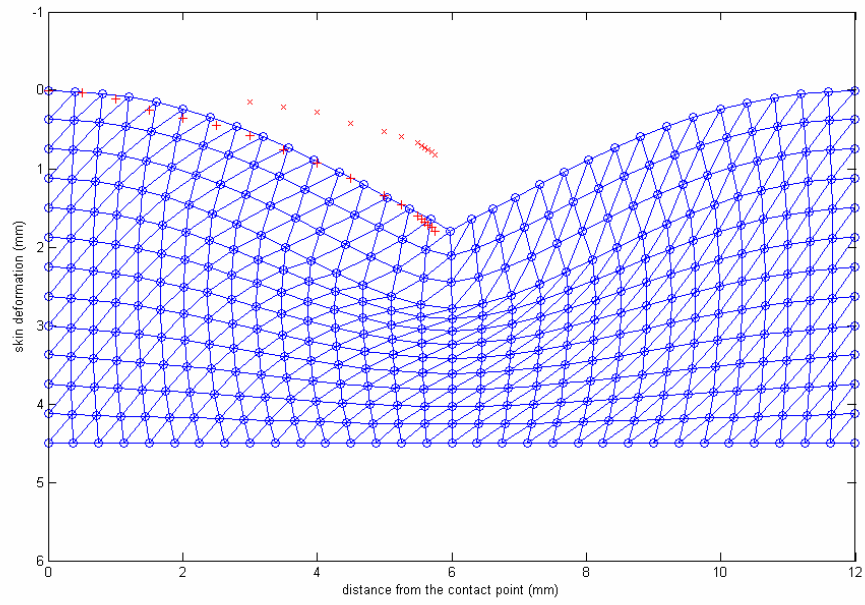
Σχήμα (3.15β): Προσδιορισμός σφάλματος της επιφάνειας του μοντέλου ως προς το σημείο αναφοράς

Έτσι, προσδιορίζουμε την τιμή μιας μεταβλητής σφάλματος ($disterr$) της παραμόρφωσης του κοντινότερου σημείου στο σημείο επαφής. Σύμφωνα με την τιμή αυτή αυξάνουμε ή μειώνουμε ελάχιστα τη διείδυση του σημειακού φορτίου ώστε να συμπέσει περίπου το πρώτο σημείο για το οποίο έχουμε πειραματική τιμή με την επιφάνεια της προσομοίωσης. Με τον τρόπο αυτό φροντίζουμε η καμπύλη της προσομοίωσης να συμπίπτει με τα δύο ακραία σημεία για τα οποία έχουμε πειραματικά δεδομένα. Η καμπύλη μας δηλαδή περνάει με αυτό τον τρόπο από το σημείο των 6,0mm από το κέντρο (το οποίο λαμβάνεται ακλόνητο) καθώς και από αυτό των 0,25mm (το οποίο είναι και το κοντινότερο στο σημείο επαφής για το οποίο έχουμε πειραματική τιμή βάθους διείδυσης). Το αποτέλεσμα είναι πολύ καλύτερο. Αυτό φαίνεται εξάλλου και από την τιμή του μέσου τετραγωνικού σφάλματος η οποία μειώνεται κατά πολύ για τις καμπύλες της επιφάνειας των πεπερασμένων στοιχείων που προσεγγίζουν τα πειραματικά δεδομένα.

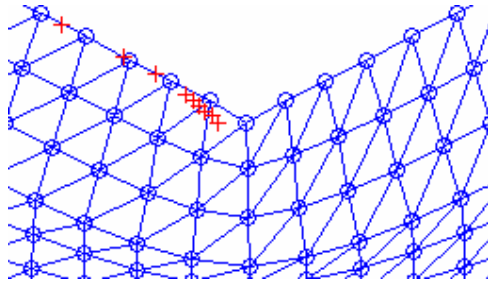


Σχήμα (3.16): Διόρθωση βάθους διείσδυσης

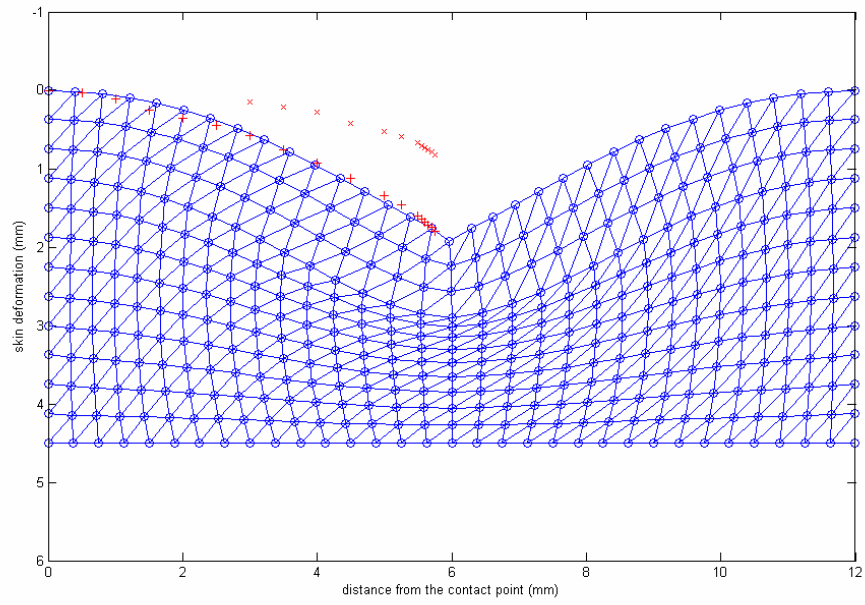
Στα παρακάτω σχήματα μπορούμε να δούμε μια καμπύλη πριν τη διόρθωση του βάθους διείσδυσης και μετά τη διόρθωσή του. Οι κόκκινοι σταυροί αποτελούν τα πειραματικά δεδομένα ενώ με μπλε είναι χρωματισμένο το μοντέλο μας. Στο πρώτο σχήμα είναι φανερό ότι ο πρώτος σταυρός, σε μικρή απόσταση από το σημείο επαφής δε συμπίπτει με την επιφάνεια του μοντέλου. Στο δεύτερο σχήμα εξαλείφεται το πρόβλημα αυτό.



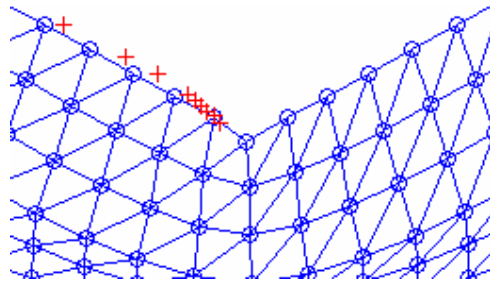
Σχήμα (3.17α): Πριν τη διόρθωση του βάθους διείσδυσης



Σχήμα (3.17β): Λεπτομέρεια



Σχήμα (3.18α): Μετά τη διόρθωση του βάθους διείσδυσης



Σχήμα (3.18β): Λεπτομέρεια

3.5 Εύρεση παραμέτρων όπου το σφάλμα ελαχιστοποιείται

Ακολουθώντας το παραπάνω σκεπτικό θα εφαρμόσουμε επαναληπτικά για τις διάφορες τιμές του συντελεστή ελαστικότητας (elasticity factor) καθώς και του πάχους του πρώτου στρώματος ως προς το συνολικό ώστε να προσαρμόσουμε το μοντέλο μας στα πειραματικά δεδομένα. Επομένως, με παραμέτρους το λόγο ελαστικότητας και το ποσοστό πάχους του ανώτερου στρώματος θα αναζητήσουμε το ελάχιστο μέσο τετραγωνικό σφάλμα και για τις δύο ομάδες τιμών (set1 και set2).

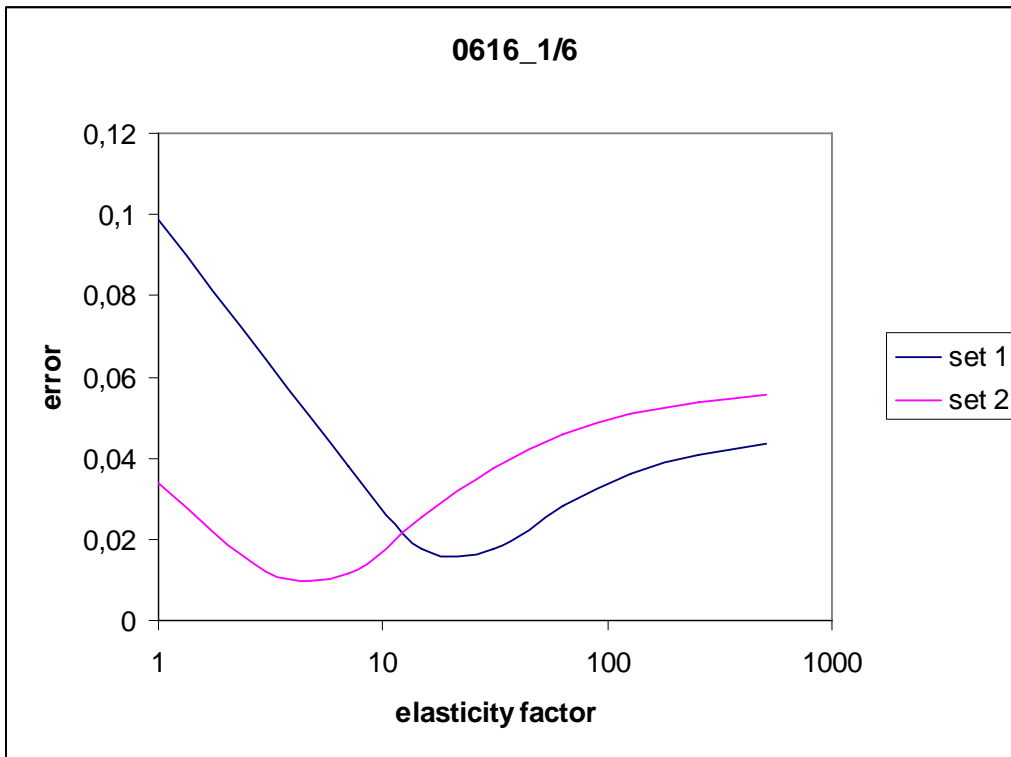
Στους πίνακες που ακολουθούν έχουμε θεωρήσει ότι το μοντέλο μας αποτελείται από 6 στρώματα ελαστικών τετραγώνων κατακόρυφα (6 layers) και 16 οριζόντια. Για το μοντέλο αυτό οι τιμές που μπορεί να πάρει το πάχος του πρώτου στρώματος είναι συγκεκριμένες (από 1 έως 5) αφού στην υλοποίηση με πεπερασμένα στοιχεία η αλλαγή ελαστικότητας απαιτεί διαχωριστική επιφάνεια, δε μπορεί δηλαδή να γίνει στο μέσο ενός στοιχείου. Για το πηλίκο ελαστικότητας ωστόσο δεν έχουμε κανένα περιορισμό, με εξαίρεση ότι το ανώτερο στρώμα πρέπει να είναι πιο ανελαστικό από το εσωτερικό, σε αντιστοιχία με το πραγματικό δέρμα. Για το λόγο αυτό θα πρέπει να εξετάσουμε ένα μεγάλο εύρος τιμών του elasticity factor. Αυξάνοντας γεωμετρικά το λόγο ελαστικότητας (el_f) για τιμές από 1 έως 511 προκύπτει ο πίνακας 3.3.

set 1	el f	1	3	7	15	31	63	127	255	511
layer 1										
1		0,098849	0,064378	0,037897	0,017701	0,017456	0,028318	0,036297	0,040957	0,043469
2		0,098828	0,059213	0,026587	0,008461	0,018544	0,027661	0,032844	0,035593	0,037008
3		0,098819	0,063454	0,030322	0,006466	0,011731	0,020515	0,025324	0,027827	0,029105
4		0,098803	0,072696	0,043768	0,019236	0,005814	0,010954	0,016082	0,018899	0,020361
5		0,098782	0,084517	0,064572	0,042276	0,023226	0,011584	0,009017	0,010718	0,012272

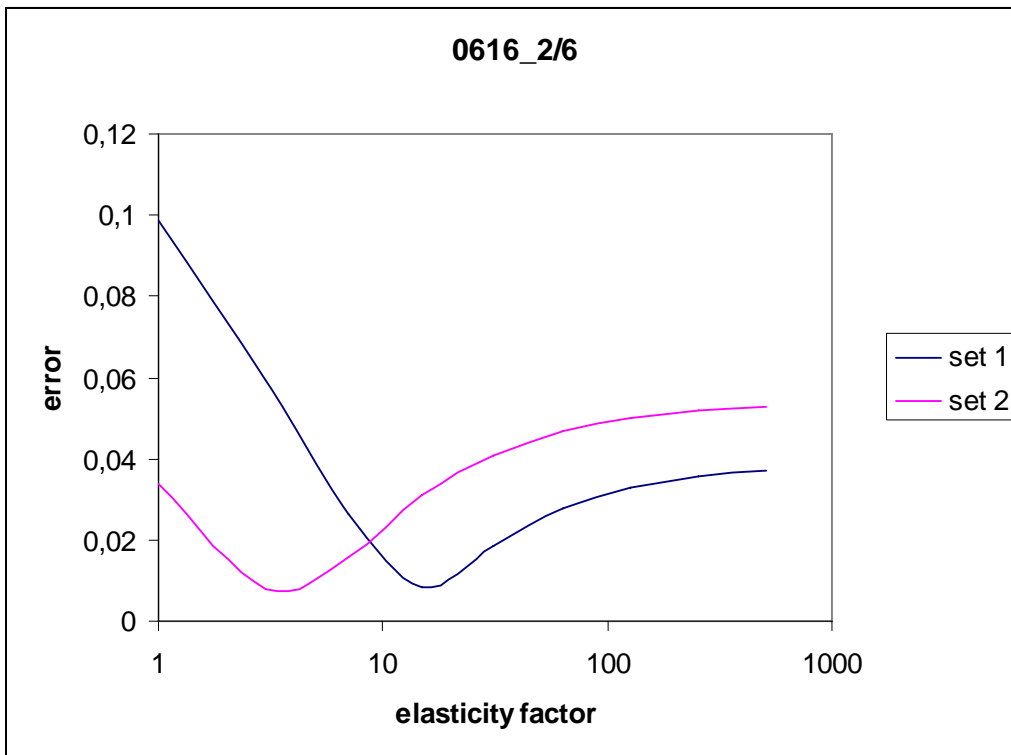
set 2	el f	1	3	7	15	31	63	127	255	511
layer 1										
1		0,033983	0,011838	0,011805	0,025701	0,037616	0,045974	0,051095	0,053964	0,055487
2		0,034015	0,007922	0,015769	0,030933	0,040982	0,046913	0,050163	0,051869	0,052743
3		0,03401	0,010776	0,012447	0,027313	0,037003	0,042579	0,045582	0,047143	0,047939
4		0,034003	0,016895	0,006535	0,01889	0,029116	0,035358	0,038819	0,040643	0,041581
5		0,033993	0,024568	0,012745	0,00765	0,016854	0,024877	0,029899	0,032716	0,034209

Πίνακας (3.3): Μέσο τετραγωνικό σφάλμα για διάφορες τιμές ποσοστού πάχους και συντελεστή ελαστικότητας

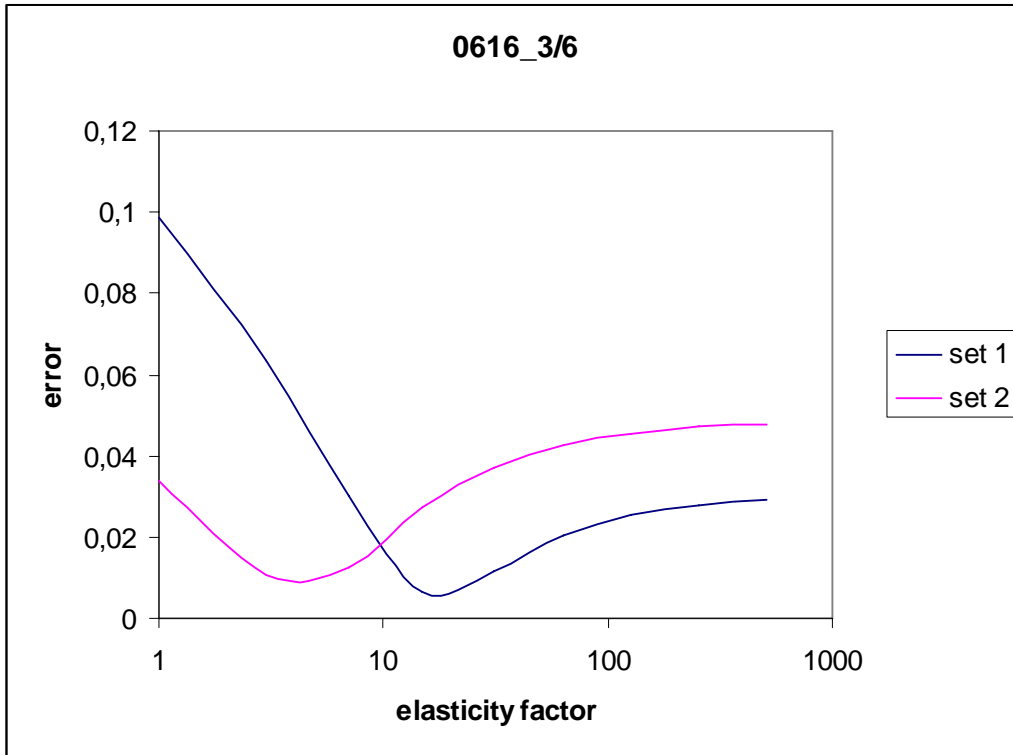
Σε λογαριθμική κλίμακα παρουσιάζουμε τα αποτελέσματα αυτά για τις διάφορες τιμές του πάχους του πρώτου στρώματος. Για κάθε ομάδα τιμών τα αντίστοιχα αποτελέσματα είναι διαφορετικά. Για το λόγο αυτό θα βρούμε την τομή όπου το σφάλμα είναι και για τις δύο ομάδες ελάχιστο.



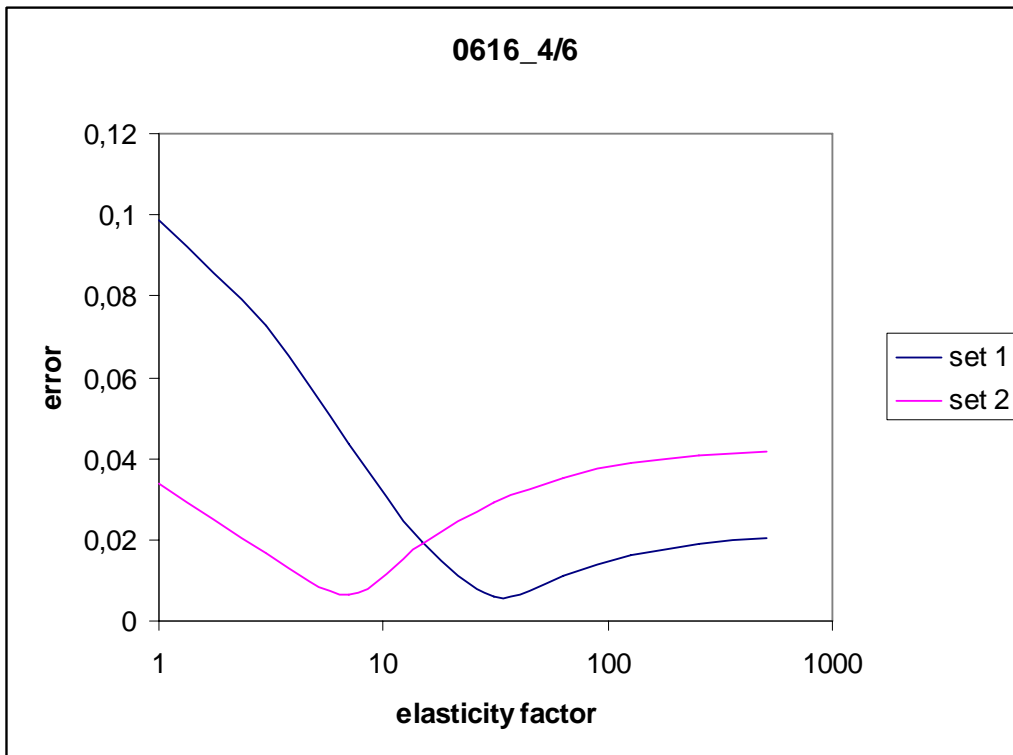
Σχήμα (3.19α): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 1/6



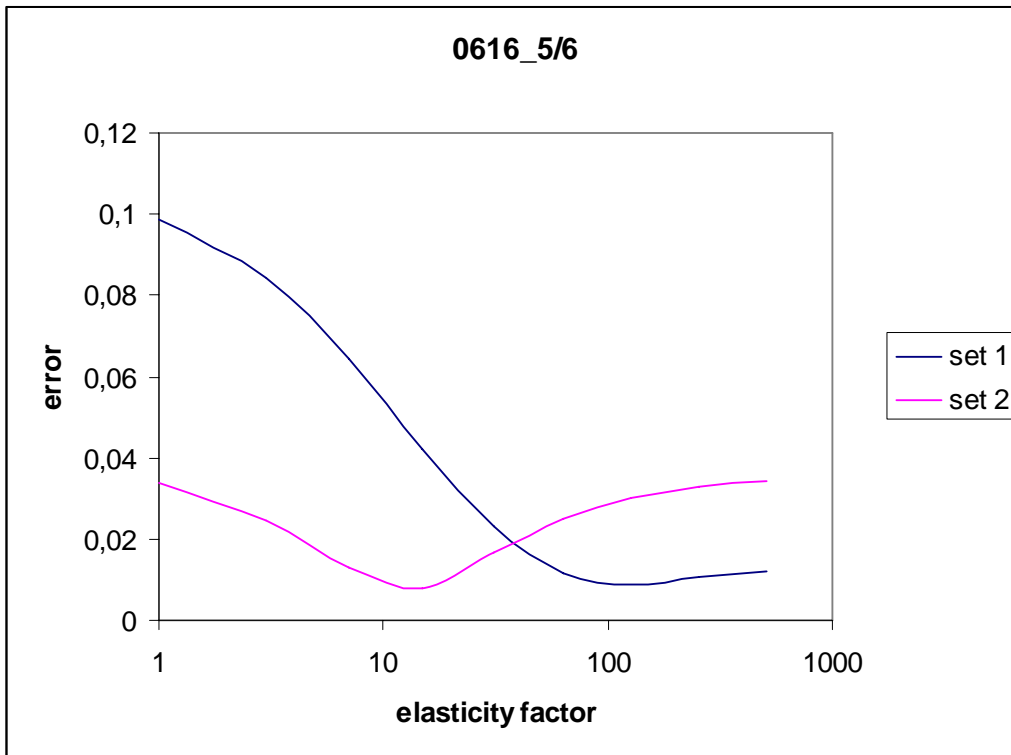
Σχήμα (3.19β): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 2/6



Σχήμα (3.19γ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 3/6



Σχήμα (3.19δ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 4/6



Σχήμα (3.19ε): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 5/6

Για τιμές πολύ μεγάλες για το λόγο ελαστικότητας τα αποτελέσματα δεν εμφανίζουν ενδιαφέρον αφού το δεύτερο στρώμα είναι πρακτικά ανύπαρκτο. Επομένως δε χρειάζεται να αναζητήσουμε λύση σε μεγαλύτερες τιμές του συντελεστή ελαστικότητας.

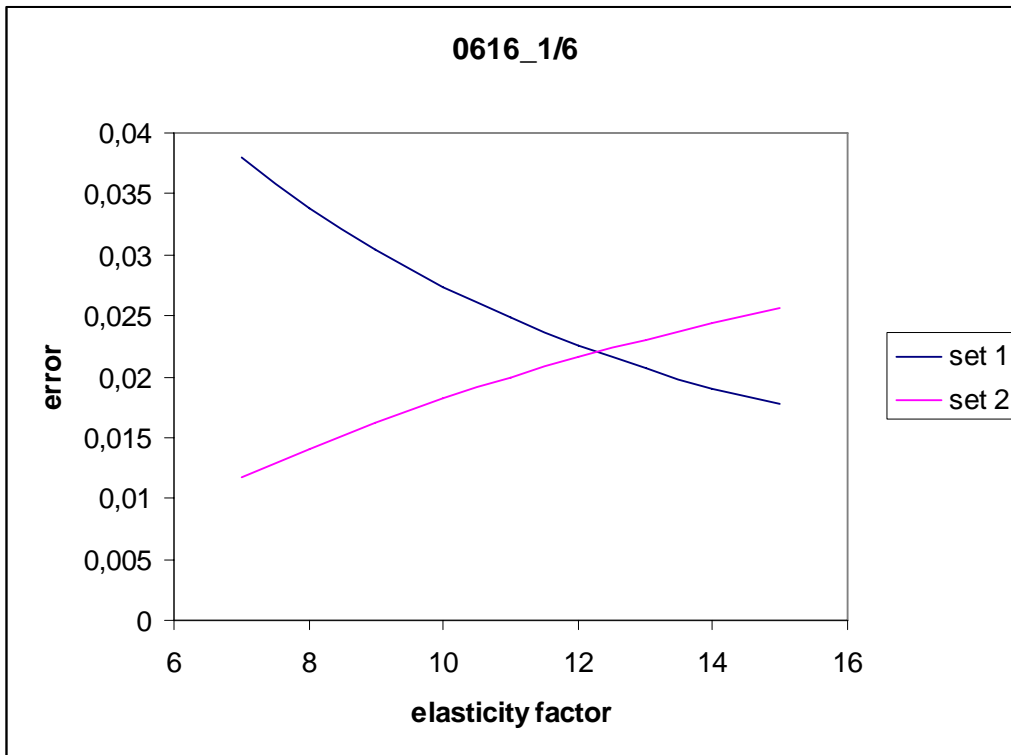
Για τις δύο ομάδες τιμών (set 1 και set 2) έχουμε διαφορετικές τιμές των παραμέτρων για τις οποίες το σφάλμα είναι ελάχιστο. Ωστόσο, παρατηρώντας τον πίνακα 3.3 ενδιαφέρον και στις δύο περιπτώσεις παρουσιάζει η περιοχή μεταξύ των τιμών 7 και 15 για το λόγο ελαστικότητας. Η περιοχή αυτή των τιμών δεν έχει αναλυθεί στον πίνακα 3.3. Για το λόγο αυτό και για τις τιμές αυτές επαναλαμβάνουμε την ίδια διαδικασία, προσθέτοντας περισσότερες ενδιάμεσες τιμές, γραμμικά αυτή τη φορά (πίνακας 3.4).

set 1	el f	7	8	9	10	11	12	13	14	15
layer 1										
1		0,037938	0,033869	0,030402	0,027407	0,024816	0,02258	0,020666	0,019046	0,017699
2		0,026606	0,022019	0,018251	0,015165	0,012689	0,010792	0,009468	0,008705	0,008454
3		0,030337	0,025451	0,021335	0,017824	0,014805	0,012202	0,009961	0,008052	0,006472
4		0,04377	0,039107	0,035079	0,031569	0,028486	0,025762	0,02334	0,021178	0,01924
5		0,064572	0,060802	0,05739	0,054288	0,051458	0,048868	0,04649	0,044299	0,042275

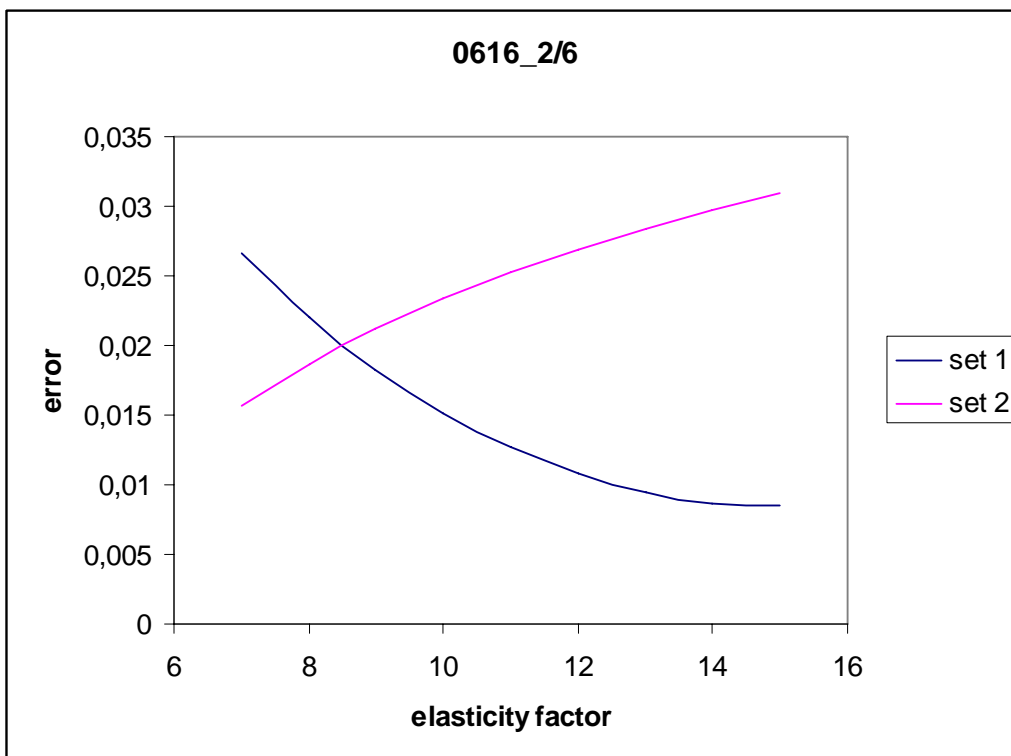
set 2	el f	7	8	9	10	11	12	13	14	15
layer 1										
1		0,011758	0,01406	0,016194	0,018149	0,019934	0,021563	0,023055	0,024425	0,025687
2		0,015711	0,018669	0,021178	0,02335	0,025246	0,026917	0,028401	0,029727	0,030919
3		0,012382	0,015268	0,01773	0,019866	0,021733	0,023376	0,024832	0,026131	0,027296
4		0,006504	0,00811	0,009968	0,011785	0,013476	0,015021	0,016427	0,017707	0,018873
5		0,012779	0,010826	0,009275	0,008107	0,007337	0,006961	0,006938	0,007194	0,007644

Πίνακας (3.4): Μέσο τετραγωνικό σφάλμα για διάφορες τιμές ποσοστού πάχους και συντελεστή ελαστικότητας

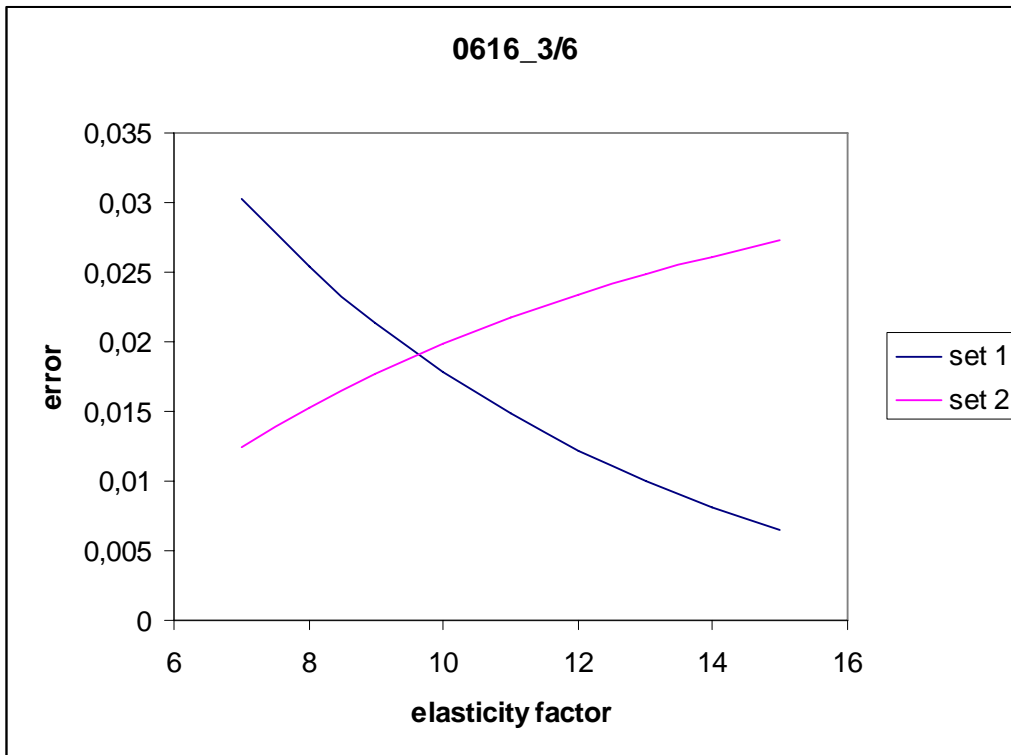
Με την ίδια διαδικασία προκύπτουν τα αντίστοιχα διαγράμματα. Τώρα οι τιμές του λόγου ελαστικότητας είναι γραμμικές και δεν είναι απαραίτητη η λογαριθμική κλίμακα.



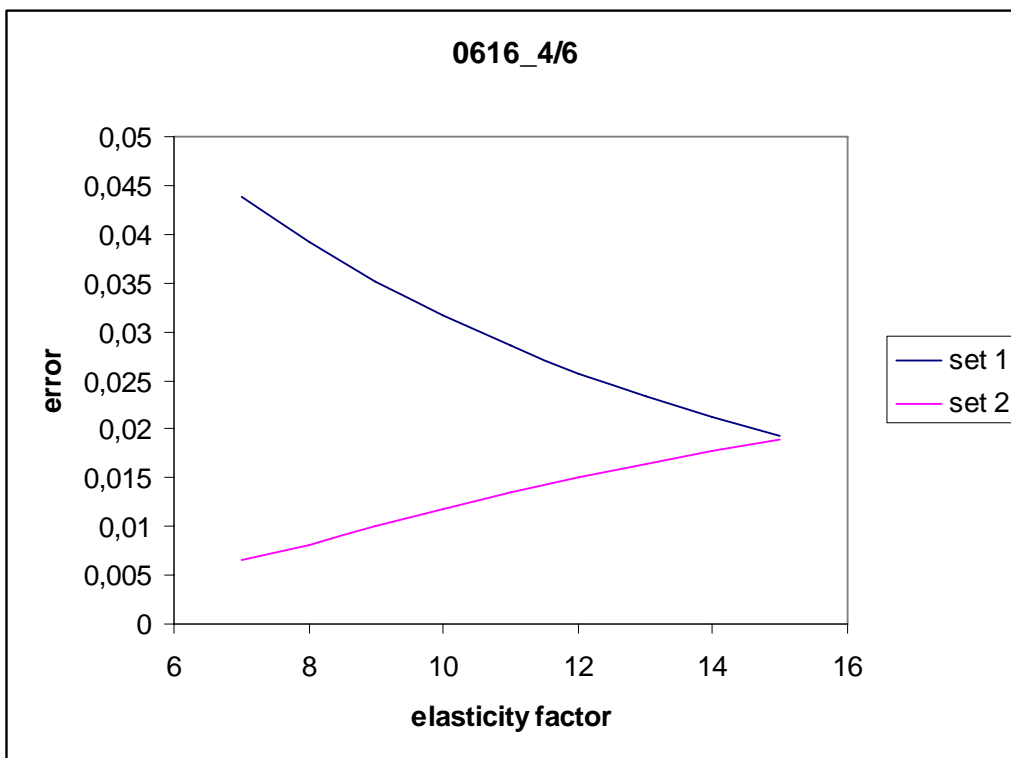
Σχήμα (3.20α): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 1/6



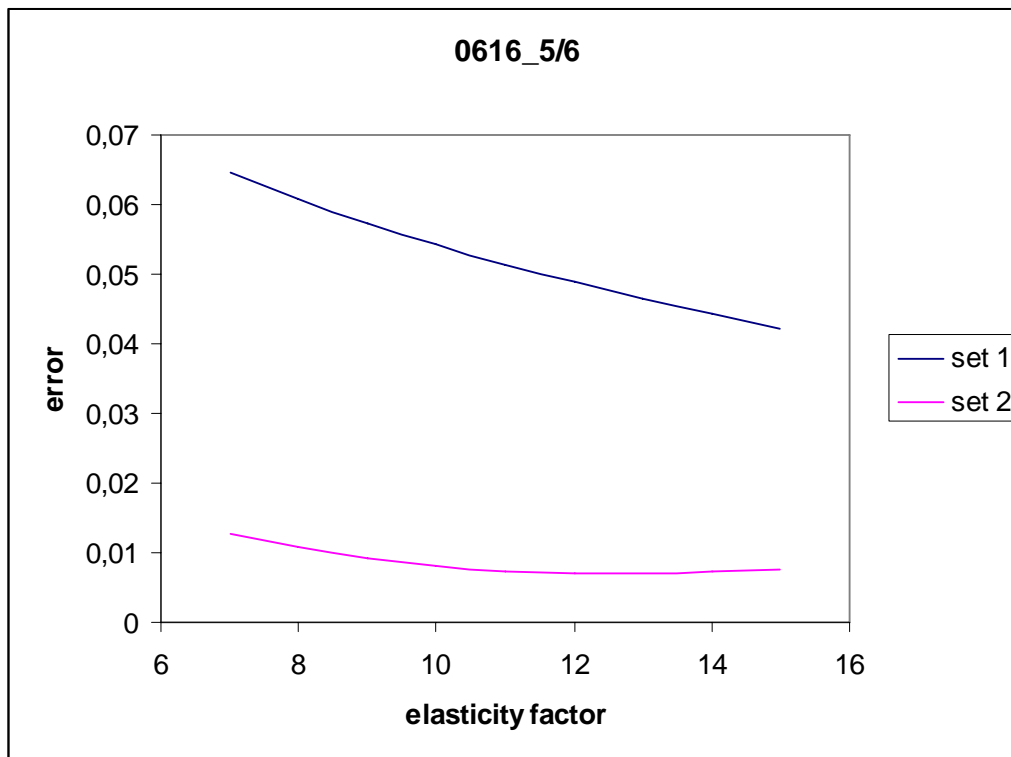
Σχήμα (3.20β): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 2/6



Σχήμα (3.20γ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 3/6



Σχήμα (3.20δ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 4/6

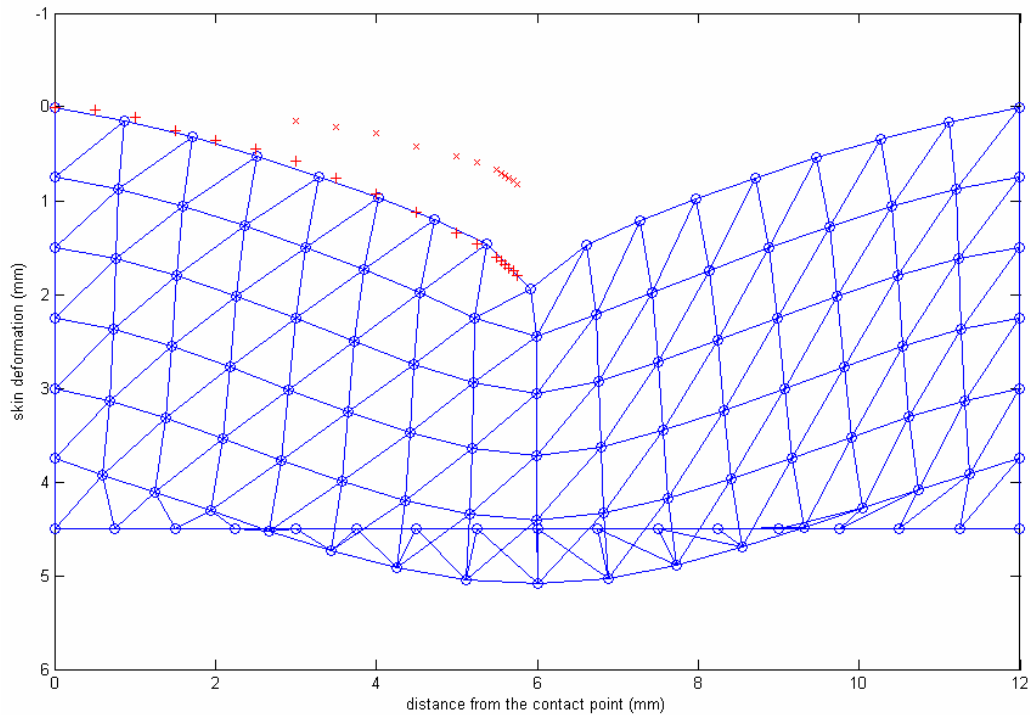


Σχήμα (3.20ε): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 5/6

Από τους πίνακες και τα διαγράμματα προκύπτουν πολλά ικανοποιητικά ζεύγη λόγου πάχους – λόγου ελαστικότητας για τα οποία το μοντέλο μας ως προς το μέσο τετραγωνικό σφάλμα ανταποκρίνεται ικανοποιητικά στα πειραματικά δεδομένα. Ωστόσο, θα πρέπει να απορρίψουμε κάποιες ομάδες αποτελεσμάτων.

Όσο αυξάνεται ο λόγος ελαστικότητας των δύο στρωμάτων, το μεγαλύτερο ποσοστό της παραμόρφωσης μεταδίδεται στο κατώτερο στρώμα. Με άλλα λόγια το πάχος του ανώτερου στρώματος είναι πρακτικά σταθερό.

Πράγματι, αν θεωρήσουμε ότι το κατώτερο στρώμα είναι πολύ πιο ελαστικό από το ανώτερο, μία πίεση θα προκαλέσει μεταβολή κυρίως στο δεύτερο στρώμα. Το πάχος του πρώτου στρώματος παραμένει σχεδόν σταθερό ενώ σχεδόν όλο το βάθος διείσδυσης μετατοπίζεται στο δεύτερο στρώμα. Αυξάνοντας το βάθος του σημείου της επιφάνειας όπου ασκείται η πίεση, θα βρεθούμε σε μία κατάσταση όπου το πάχος του κατώτερου στρώματος σχεδόν μηδενίζεται. Από το σημείο αυτό και μετά το μοντέλο είναι ακατάλληλο όπως φαίνεται στο παρακάτω σχήμα:



Σχήμα (3.21): Εσφαλμένη υλοποίηση

Παρόλο που η επιφάνεια προσομοιώνεται ικανοποιητικά τα πεπερασμένα στοιχεία περνούν κάτω από την επιφάνεια που ορίζει το οστό. Επομένως, έχουμε περιορισμό στο μέγιστο βάθος του σημείου επαφής. Η τιμή του θα είναι περίπου ίση με το πάχος του κατώτερου στρώματος.

Συνεπώς, μπορούμε να συμπεράνουμε ότι για να ανταποκρίνεται το μοντέλο μας σε όσο το δυνατόν μεγαλύτερο βάθος για το σημείο επαφής θα πρέπει το ανώτερο στρώμα να είναι όσο γίνεται λεπτότερο. Άρα, θα προτιμήσουμε το μοντέλο εκείνο στο οποίο το πάχος του πρώτου στρώματος είναι ελάχιστο. Επιπρόσθετα, αυτό θα κάνει το μοντέλο να ανταποκρίνεται περισσότερο στην πραγματικότητα, όχι μόνο στην επιφάνεια αλλά και εσωτερικά. Στο δέρμα έχουμε μια μικρή περιοχή κοντά στην επιφάνεια της επιδερμίδας περισσότερο ανελαστική, ενώ το μεγαλύτερο τμήμα στο εσωτερικό εμφανίζει αυξημένη ελαστικότητα.

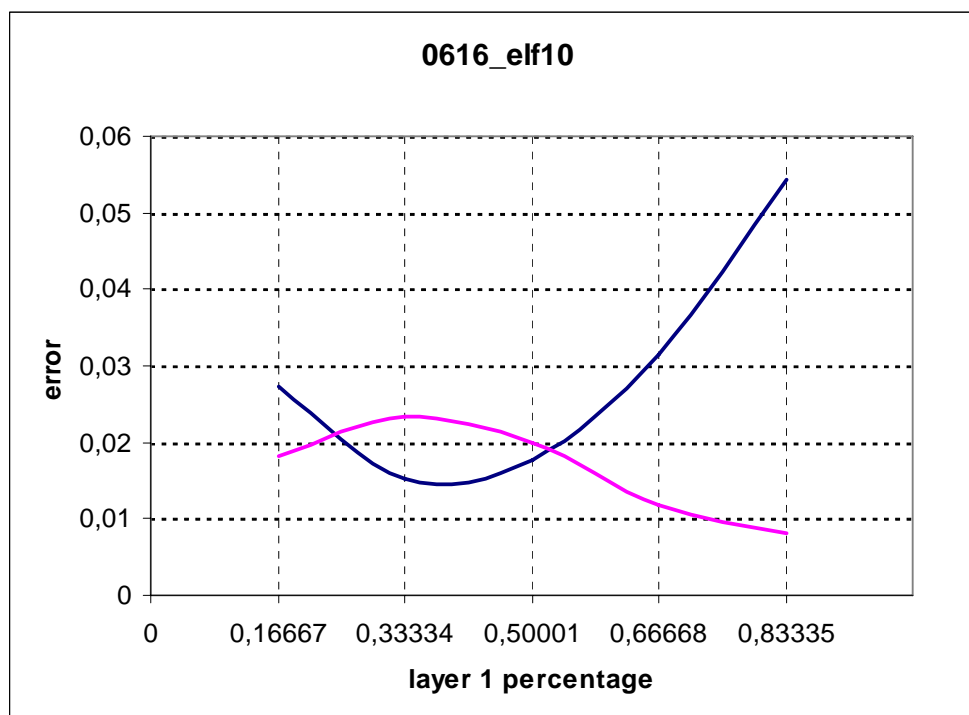
Σύμφωνα με τα παραπάνω μπορούμε να θεωρήσουμε ότι η περίπτωση όπου το πάχος του πρώτου στρώματος είναι τα $5/6$ του συνολικού μπορεί να μη ληφθεί υπ' όψιν για τη διεξαγωγή αποτελεσμάτων. Έτσι το τελευταίο διάγραμμα μπορεί να παραλειφθεί. Σύμφωνα με τα υπόλοιπα δεδομένα προκύπτει ότι μια καλή προσέγγιση για το διδιάστατο μοντέλο προκύπτει όταν η σχέση ελαστικότητας μεταξύ των δύο στρωμάτων (layer1 και layer2) είναι 10. Η επιλογή αυτή δε δίνει το καλύτερο δυνατό αποτέλεσμα για κανένα από τα δύο σετ πειραματικών τιμών (set1 και set2) αλλά είναι η βέλτιστη για το συνδυασμό τους.

Με λόγο ελαστικότητας ίσο με 10 προκύπτει επομένως ο παρακάτω πίνακας για τις διάφορες τιμές του πάχους του πρώτου στρώματος, για τις δύο ομάδες δεδομένων.

	elasticity factor 10 between two layers				
layer1 %	0,166667	0,333333	0,5	0,666667	0,833333
set 1 error	0,027407	0,015165	0,017824	0,031569	0,054288
set 2 error	0,018149	0,02335	0,019866	0,011785	0,008107

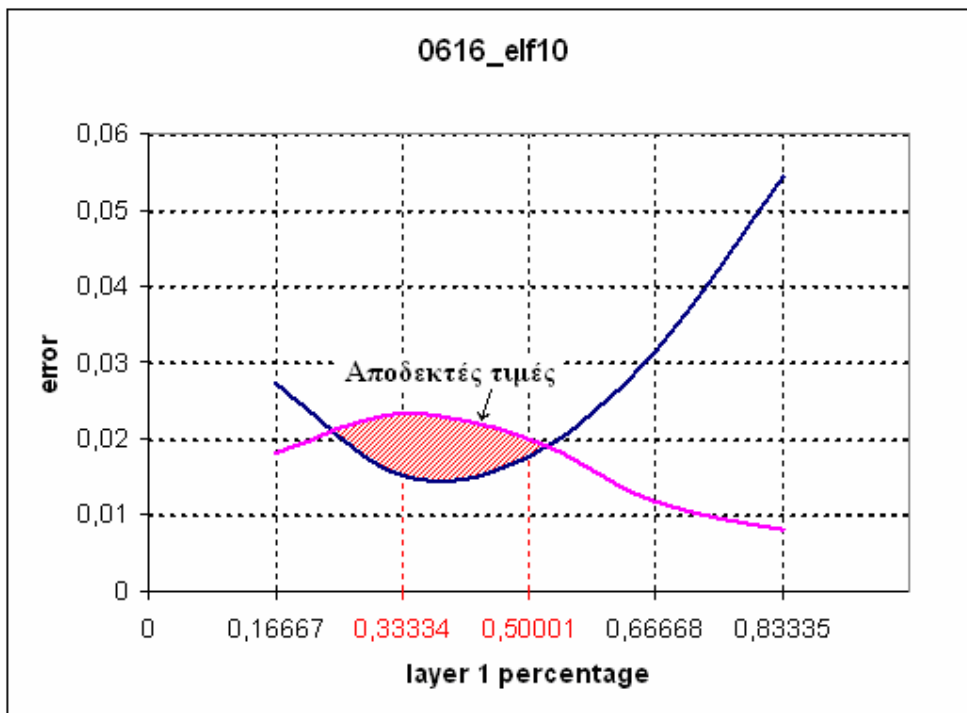
Πίνακας (3.5): Μέσο τετραγωνικό σφάλμα για λόγο ελαστικότητας 10

Αντίστοιχα έχουμε το παρακάτω διάγραμμα:



Σχήμα (3.22α): Διάγραμμα σφάλματος για λόγο ελαστικότητας 10

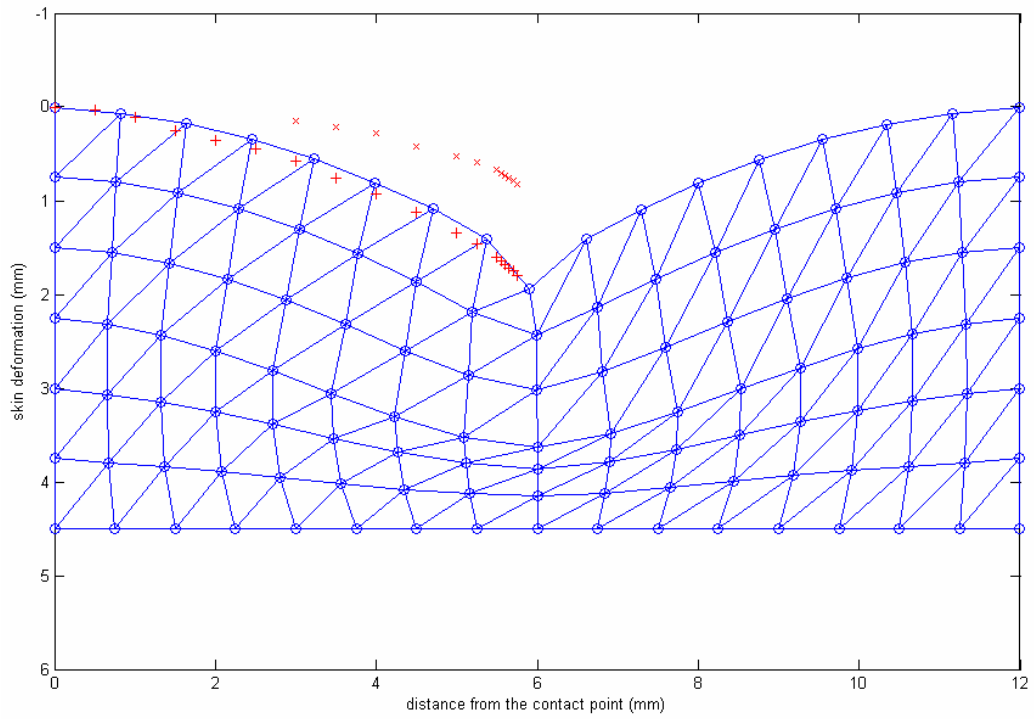
Από το τελευταίο διάγραμμα είναι εύκολο να συμπεράνουμε ότι το ποσοστό του πάχους του πρώτου στρώματος για το οποίο το διδιάστατο μοντέλο ανταποκρίνεται καλύτερα στα πειραματικά δεδομένα είναι περίπου 0,5. Με άλλα λόγια τα δύο στρώματα θα πρέπει να έχουν το ίδιο πάχος και το δεύτερο στρώμα να είναι 10 φορές πιο ελαστικό από το πρώτο.



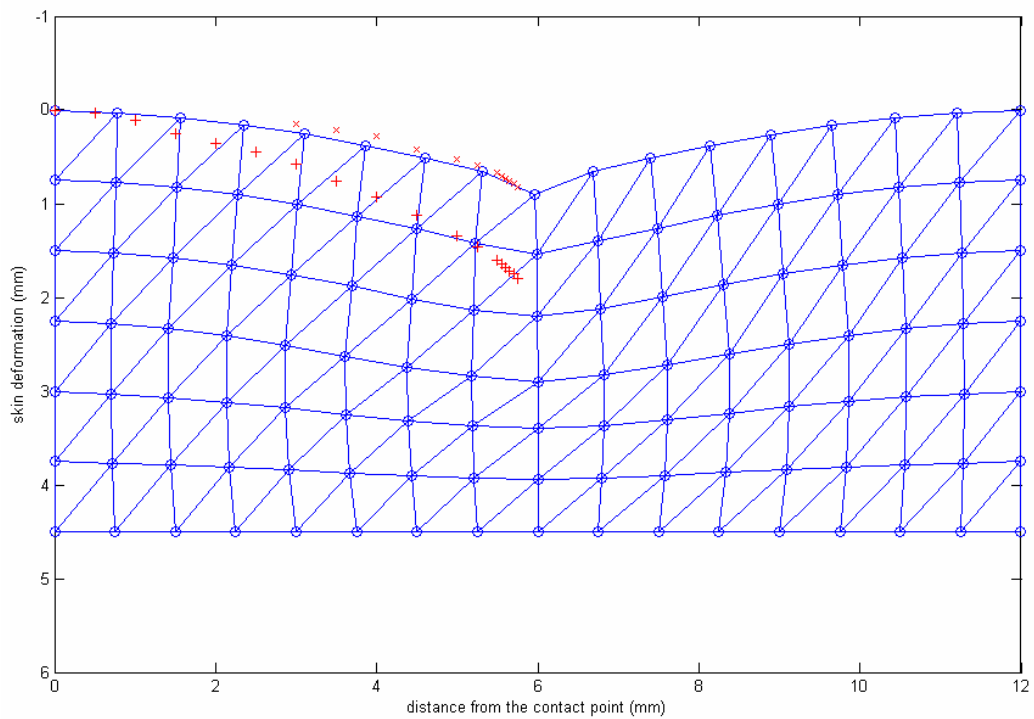
Σχήμα (3.22β): Περιοχή αποδεκτών τιμών για λόγο ελαστικότητας 10

Σημειώνεται ότι το ζεύγος τιμών που επιλέχθηκε είναι το βέλτιστο, αλλά δεν είναι μοναδικό. Παρατηρώντας το τελευταίο διάγραμμα θα μπορούσαμε να επιλέξουμε το ποσοστό 0,333 για το πρώτο στρώμα, αφού και αυτό βρίσκεται στην περιοχή αποδεκτών τιμών. Θα είχαμε έτσι λόγο πάχους 1:2 για τα δύο στρώματα (33,3% για το πρώτο και 66,7% για το δεύτερο), κάτι που ανταποκρίνεται περισσότερο στην πραγματικότητα. Πράγματι, το στρώμα του υποδόριου ιστού είναι παχύτερο από αυτό του κυρίως δέρματος. Επομένως μία υλοποίηση όπου το ανώτερο στρώμα που είναι και πιο ανελαστικό είναι λεπτότερο από το κατώτερο θα έμοιαζε περισσότερο με την κατανομή των στιβάδων του δέρματος. Ωστόσο, κάτι τέτοιο δεν είναι απαραίτητο για τη συγκεκριμένη μοντελοποίηση αφού εστιάζουμε κυρίως στην επιφάνεια της δομής, οπότε και παραμένουμε στο λόγο πάχους 1:1.

Ακολουθεί η αναπαράσταση του βέλτιστου μοντέλου:



Σχήμα (3.23α): Το βέλτιστο μοντέλο (1^ο σετ δεδομένων)



Σχήμα (3.23β): Το βέλτιστο μοντέλο (2^ο σετ δεδομένων)

3.6 Εύρεση παραμέτρων σε μεγαλύτερη κατάτμηση

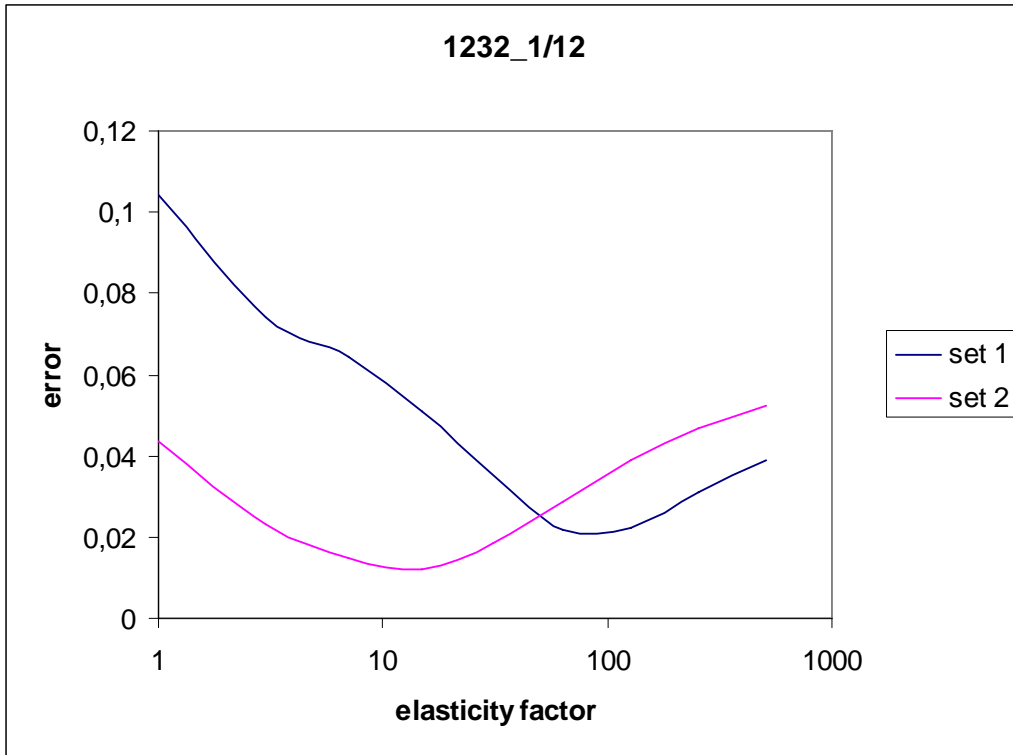
Στη συνέχεια θα εξετάσουμε εάν αυξάνοντας την κατάτμηση καταλήγουμε στα ίδια συμπεράσματα.

Έτσι, για 12 συνολικά στρώματα (layers) οριζόντια και 32 κατακόρυφα επαναλαμβάνουμε την ίδια διαδικασία και έχουμε:

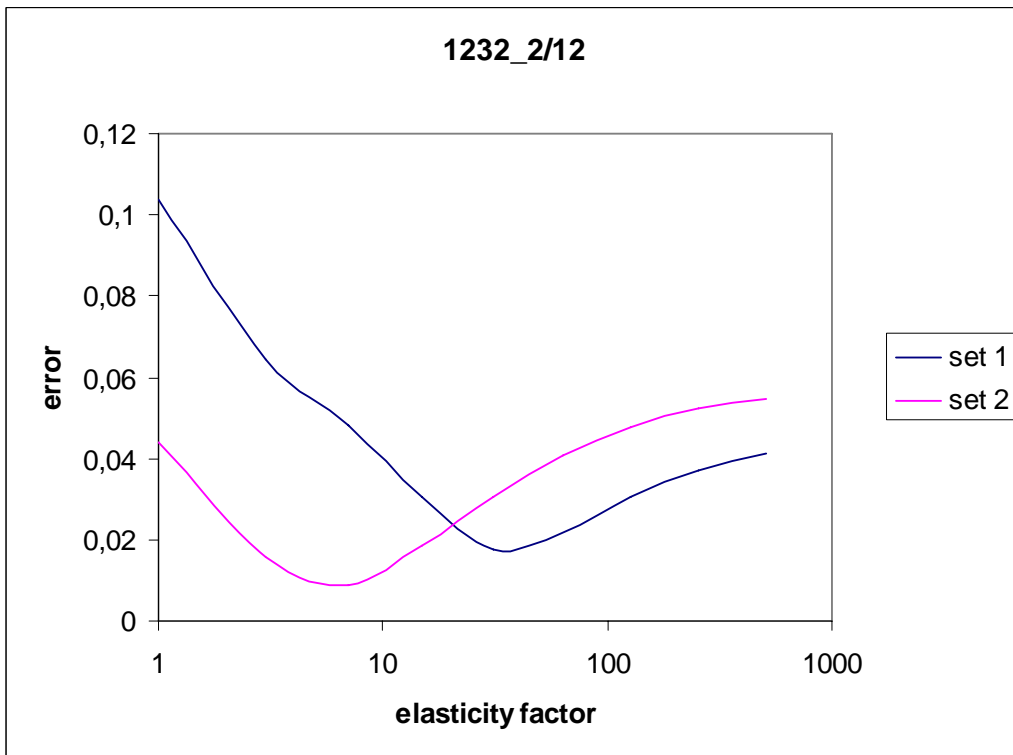
set 1	el f	1	3	7	15	31	63	127	255	511
layer 1										
1		0,10413	0,07398	0,064462	0,050921	0,035104	0,021963	0,022166	0,031159	0,03882
2		0,1036	0,064191	0,048055	0,030629	0,017802	0,021591	0,030704	0,037219	0,041015
3		0,10358	0,056798	0,035389	0,017851	0,016096	0,025593	0,032785	0,036983	0,039241
4		0,10346	0,052107	0,027042	0,011448	0,016667	0,025885	0,031621	0,034767	0,036412
5		0,10325	0,050332	0,022516	0,0079465	0,015495	0,024098	0,029129	0,031822	0,033214
6		0,10294	0,051175	0,021438	0,0057187	0,01298	0,021237	0,025991	0,028519	0,029822
7		0,10252	0,054133	0,023437	0,0055763	0,0097496	0,017764	0,022483	0,02501	0,026316
8		0,10201	0,058736	0,028238	0,0083631	0,0065385	0,013956	0,018748	0,021386	0,022763
9		0,10142	0,064655	0,035786	0,014355	0,0052989	0,010145	0,014897	0,017727	0,019249
10		0,10074	0,07173	0,046574	0,024763	0,0092691	0,0073015	0,011088	0,014085	0,015848
11		0,099955	0,080001	0,062212	0,042848	0,02442	0,010867	0,0081305	0,010344	0,012422

set 2	el f	1	3	7	15	31	63	127	255	511
layer 1										
1		0,043713	0,023095	0,014656	0,011965	0,018477	0,028908	0,039114	0,047024	0,052188
2		0,044238	0,015742	0,0089148	0,018612	0,03071	0,040862	0,047943	0,052252	0,054654
3		0,04423	0,011323	0,010588	0,025279	0,037171	0,045287	0,050177	0,052889	0,054321
4		0,044183	0,0098527	0,012444	0,028134	0,039162	0,045986	0,04984	0,051898	0,052962
5		0,0441	0,010988	0,012291	0,028036	0,038534	0,044765	0,048191	0,049991	0,050915
6		0,043983	0,013687	0,010502	0,025935	0,036267	0,042343	0,045661	0,047399	0,048288
7		0,043836	0,0172	0,0080157	0,022411	0,032826	0,039041	0,042458	0,044253	0,045174
8		0,043669	0,02119	0,0069613	0,017795	0,028403	0,035005	0,038706	0,040671	0,041683
9		0,04349	0,025522	0,010095	0,012446	0,022945	0,030198	0,034432	0,036726	0,037921
10		0,043306	0,03017	0,016723	0,0086079	0,01606	0,024166	0,029377	0,032328	0,033901
11		0,043119	0,035215	0,026298	0,015552	0,0091524	0,014959	0,021916	0,026512	0,02914

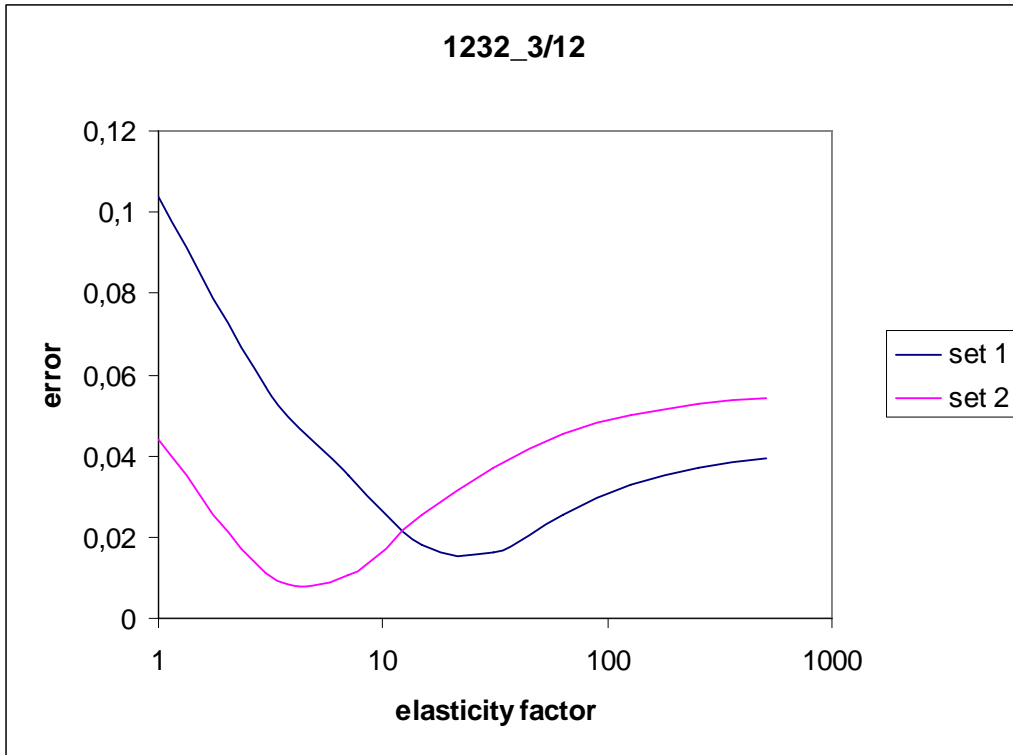
Πίνακας (3.6): Μέσο τετραγωνικό σφάλμα για διάφορες τιμές ποσοστού πάχους και συντελεστή ελαστικότητας



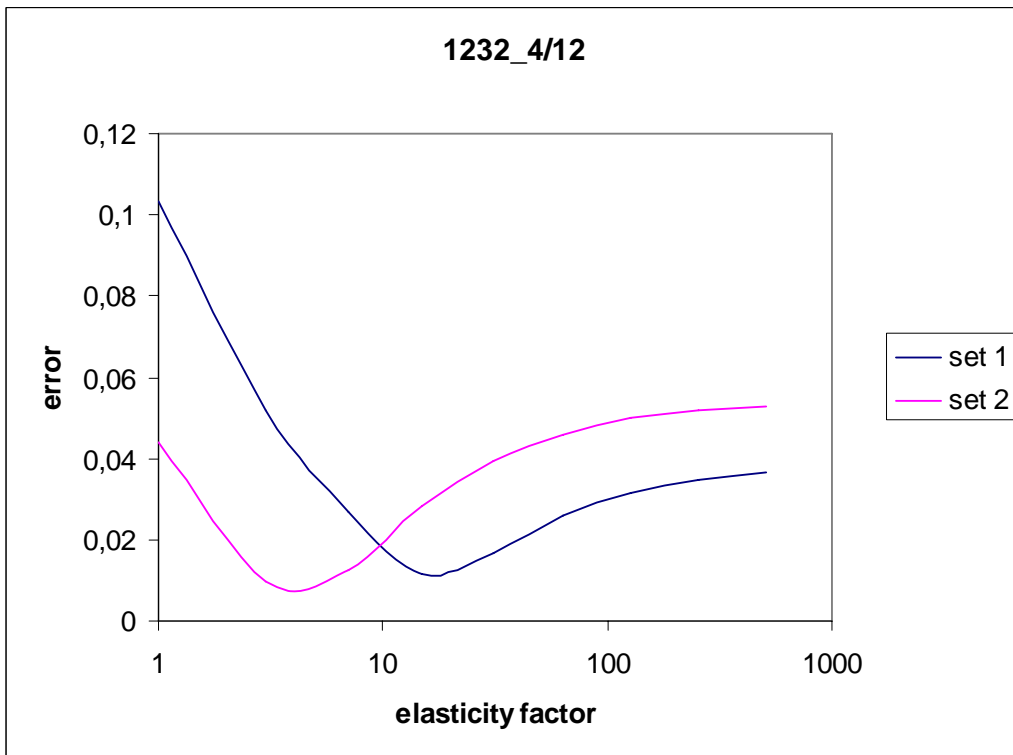
Σχήμα (3.24α): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 1/12



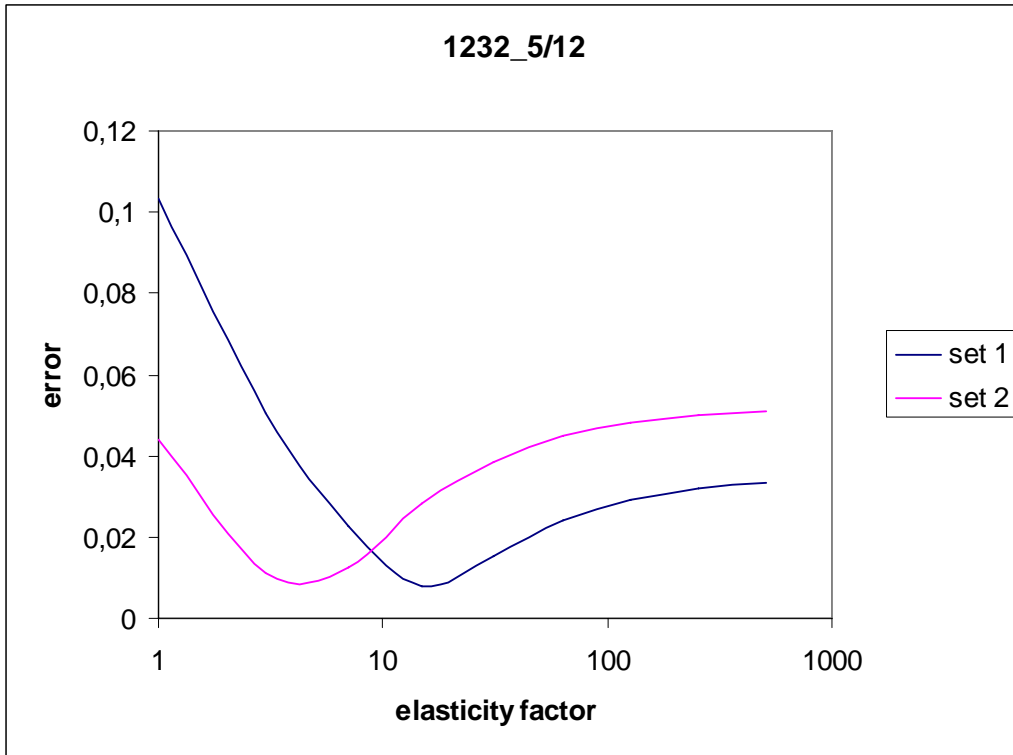
Σχήμα (3.24β): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 2/12



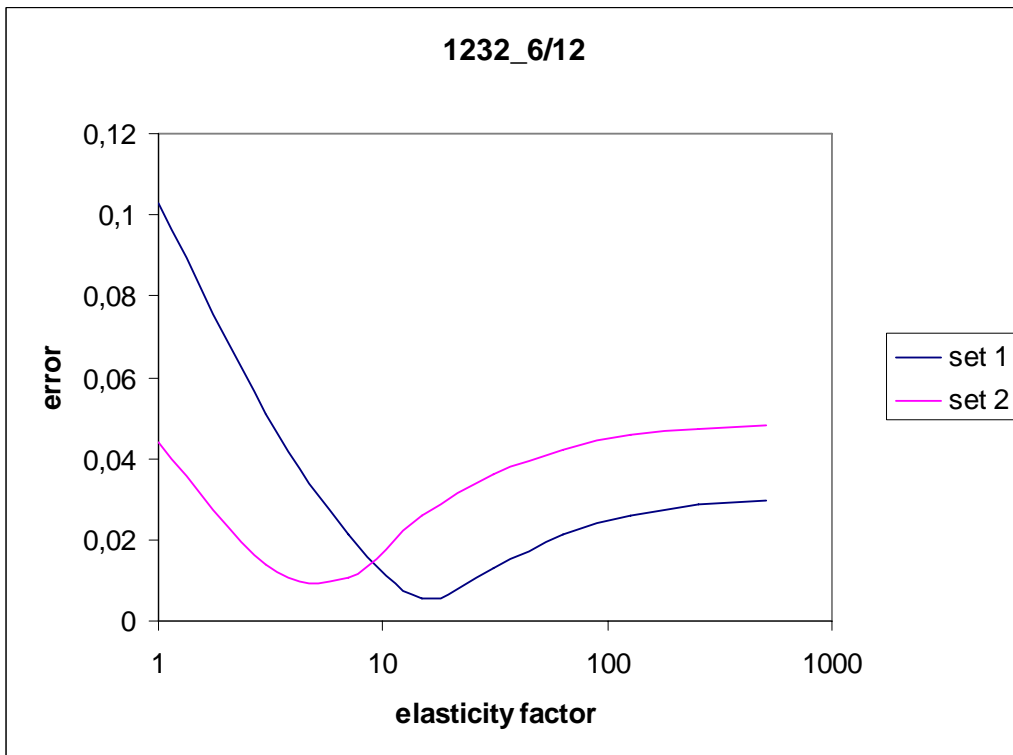
Σχήμα (3.24γ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 3/12



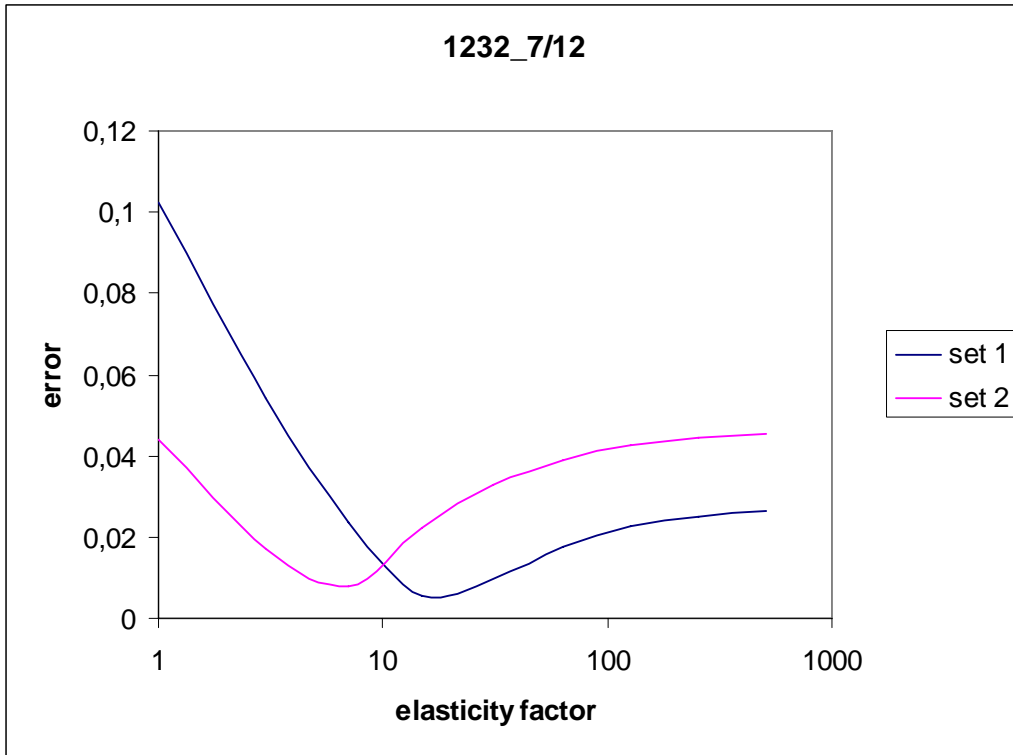
Σχήμα (3.24δ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 4/12



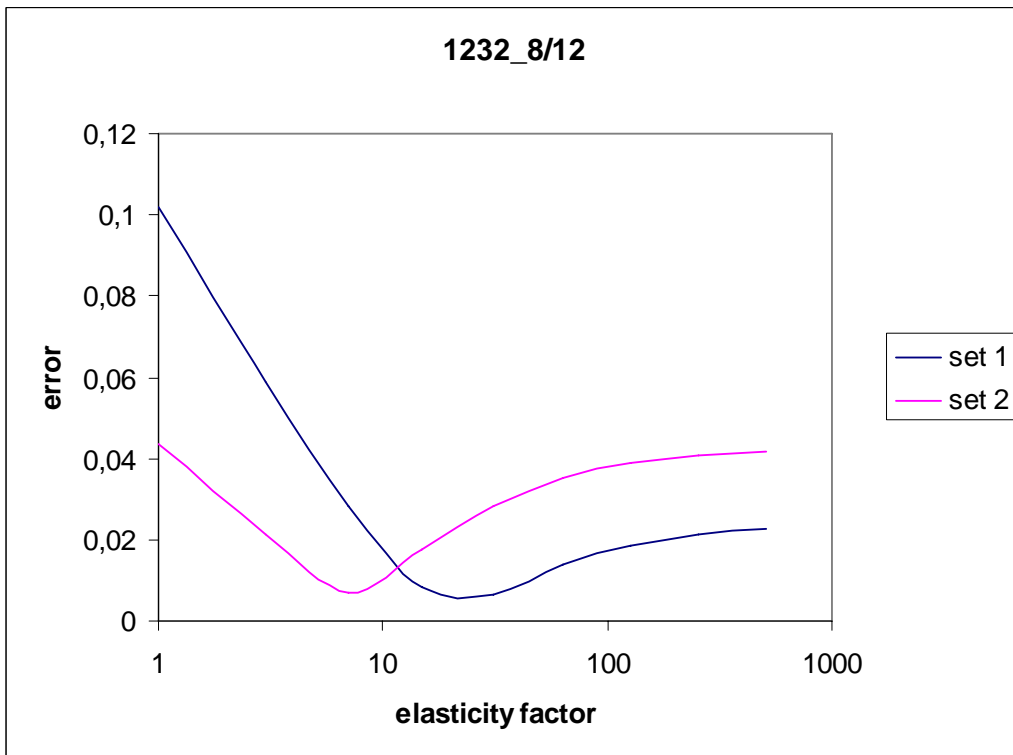
Σχήμα (3.24ε): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 5/12



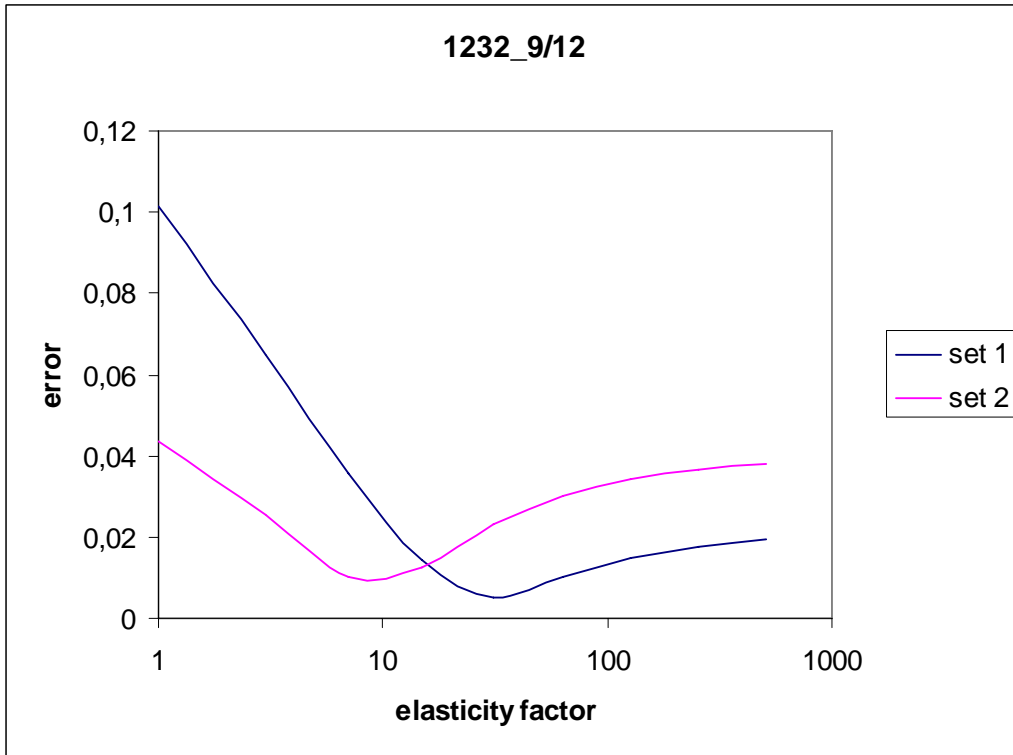
Σχήμα (3.24στ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 6/12



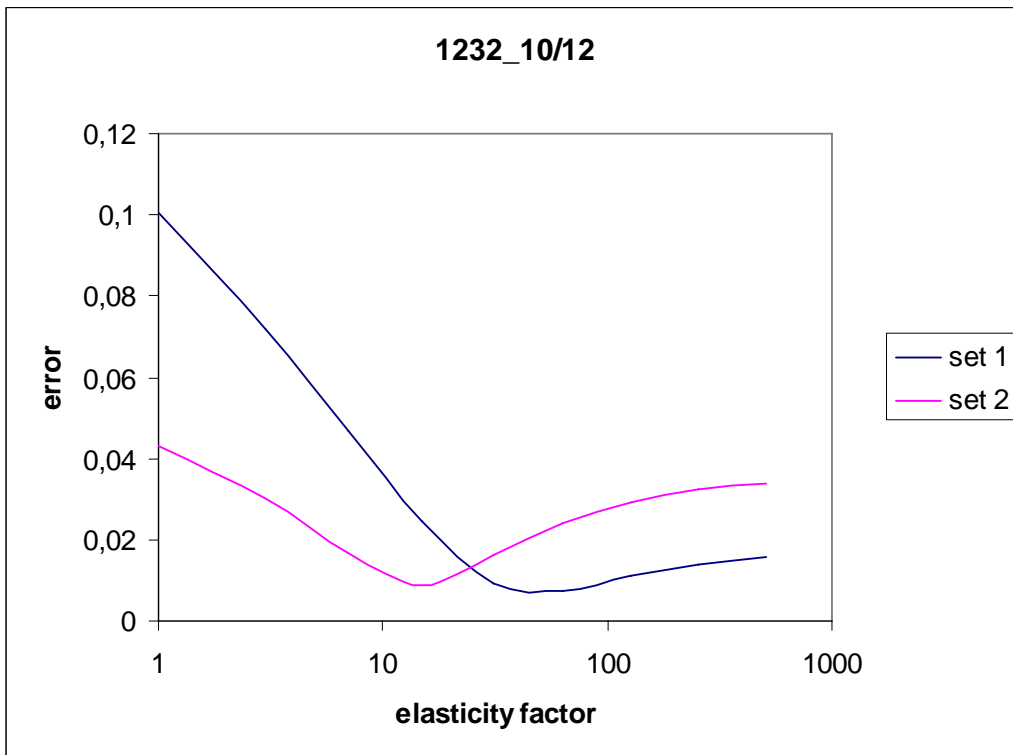
Σχήμα (3.24ζ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 7/12



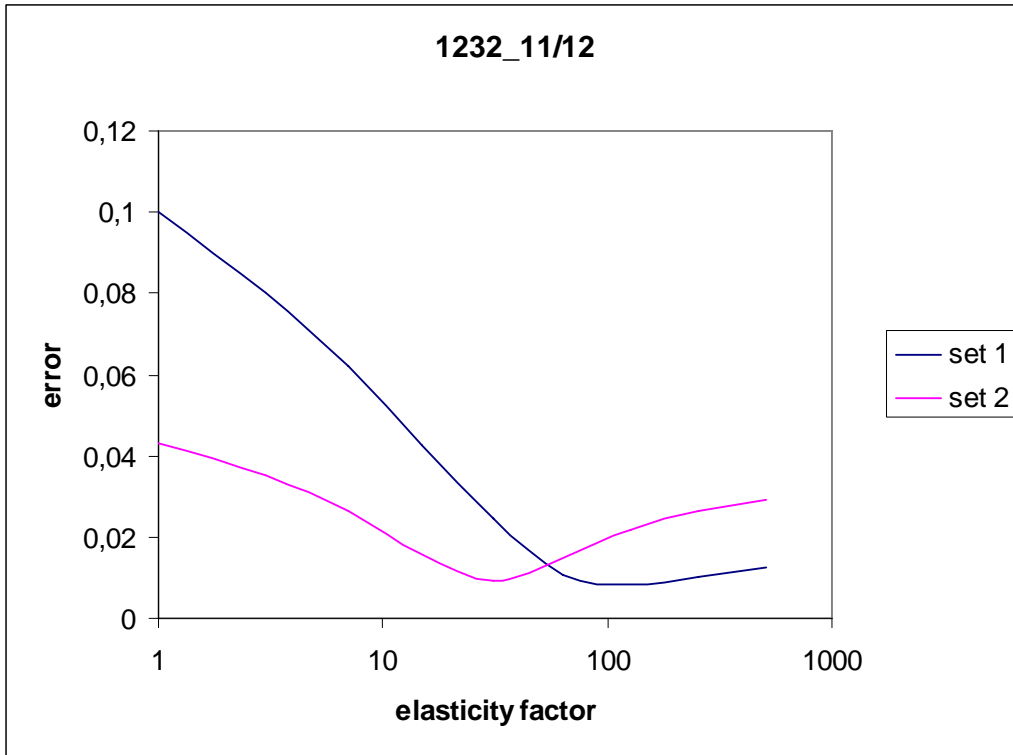
Σχήμα (3.24η): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 8/12



Σχήμα (3.240): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 9/12



Σχήμα (3.241): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 10/12



Σχήμα (3.24α): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 11/12

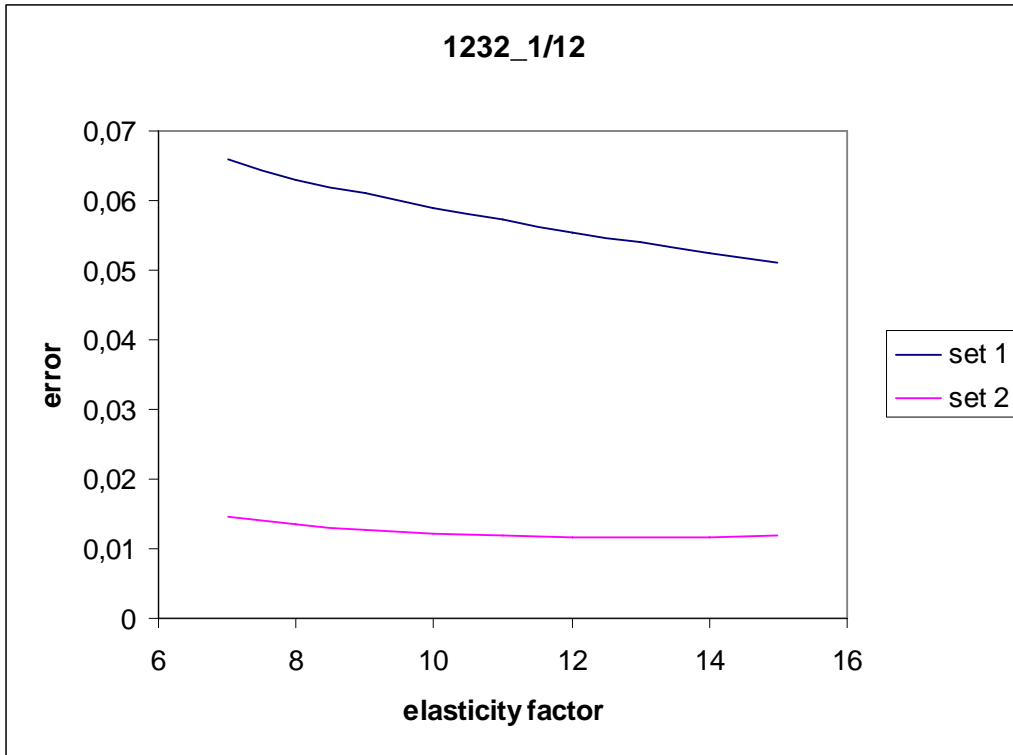
Πάλι το ενδιαφέρον εστιάζεται στην περιοχή τιμών μεταξύ 7 και 15 για το πηλίκο ελαστικότητας. Επομένως, θα συνεχίσουμε με τρόπο ανάλογο με τον προηγούμενο. Παίρνοντας τώρα γραμμικά τις τιμές από 7 έως 15 θα έχουμε:

set 1	el f	7	8	9	10	11	12	13	14	15
layer 1										
1		0,066036	0,063083	0,060972	0,059033	0,057223	0,055527	0,053932	0,052428	0,051004
2		0,049289	0,045884	0,043103	0,040567	0,038242	0,036108	0,034147	0,032344	0,030686
3		0,03703	0,032908	0,029795	0,027069	0,024671	0,022573	0,020749	0,019182	0,017852
4		0,029535	0,024825	0,021651	0,018979	0,016714	0,014833	0,013317	0,012154	0,011324
5		0,026452	0,020908	0,017678	0,01503	0,012783	0,010923	0,0094547	0,0083884	0,0077274
6		0,026949	0,020178	0,016855	0,014275	0,01202	0,010047	0,0083656	0,0069967	0,0059741
7		0,030029	0,021614	0,018011	0,015725	0,013762	0,011938	0,010258	0,008747	0,0074213
8		0,034825	0,026743	0,022713	0,019594	0,016916	0,015164	0,013822	0,012576	0,011385
9		0,042574	0,034513	0,030386	0,027238	0,024527	0,022121	0,019966	0,018027	0,016277
10		0,052611	0,045273	0,041264	0,038195	0,035531	0,033133	0,030951	0,028954	0,027121
11		0,067679	0,061119	0,057569	0,054926	0,05264	0,050545	0,048592	0,046758	0,045031

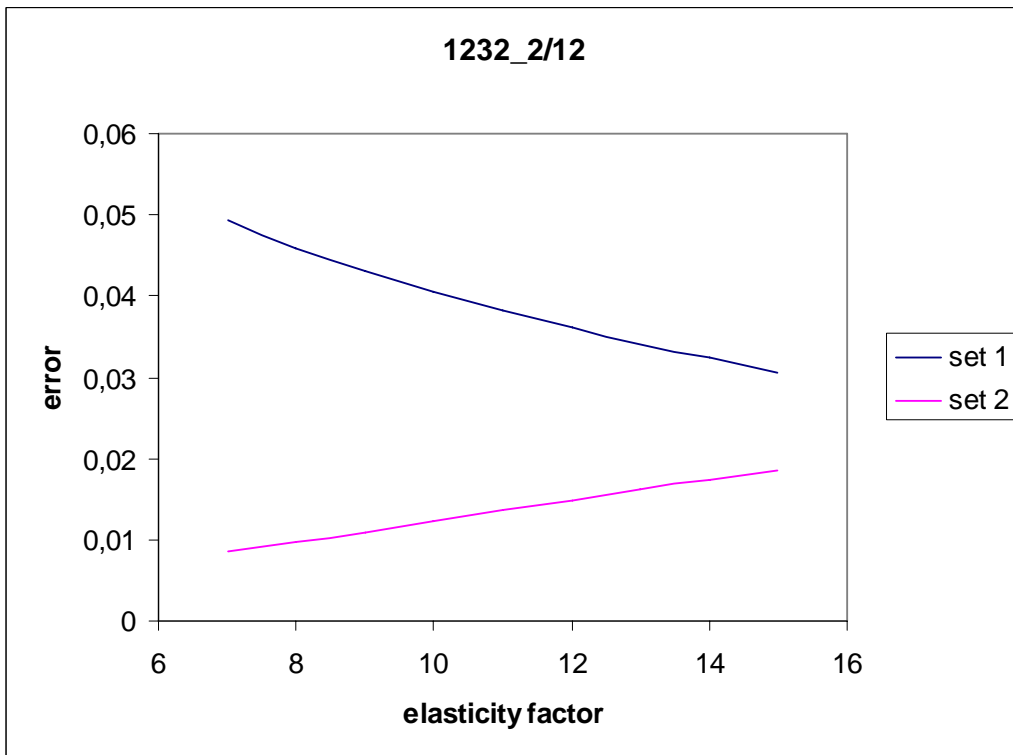
set 2	el f	7	8	9	10	11	12	13	14	15
layer 1										
1		0,014678	0,013602	0,012813	0,012256	0,011897	0,011709	0,011664	0,011736	0,011904
2		0,0085499	0,0096327	0,010893	0,012251	0,013613	0,014935	0,016201	0,017405	0,018545
3		0,0099797	0,012856	0,015216	0,017327	0,019223	0,020934	0,022487	0,023902	0,025198
4		0,011758	0,015099	0,017714	0,019984	0,021979	0,023746	0,025325	0,026743	0,028025
5		0,011584	0,014993	0,017641	0,019924	0,021919	0,023677	0,025238	0,026634	0,027889
6		0,0099141	0,013093	0,015659	0,017892	0,019853	0,021587	0,023128	0,024507	0,025748
7		0,0079089	0,010203	0,012475	0,014549	0,016418	0,018095	0,019602	0,020961	0,02219
8		0,0078534	0,0077984	0,009084	0,010637	0,012206	0,013702	0,015097	0,016385	0,017572
9		0,011533	0,0090704	0,008252	0,0082052	0,0086756	0,0094485	0,010368	0,011341	0,012315
10		0,018149	0,015072	0,013196	0,011725	0,010577	0,0097359	0,0091866	0,0089042	0,008852
11		0,027506	0,024908	0,023231	0,02176	0,020408	0,019158	0,018003	0,016938	0,015957

Πίνακας (3.7): Μέσο τετραγωνικό σφάλμα για διάφορες τιμές ποσοστού πάχους και συντελεστή ελαστικότητας

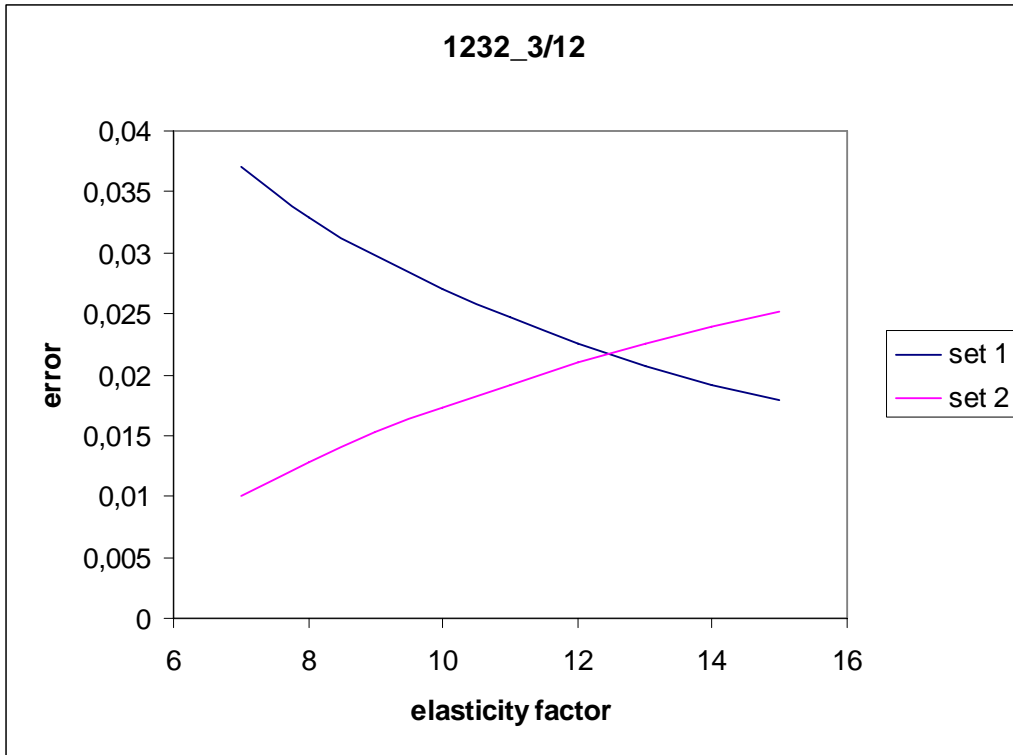
Από τον πίνακα 3.7 θα φτιάξουμε τα αντίστοιχα διαγράμματα:



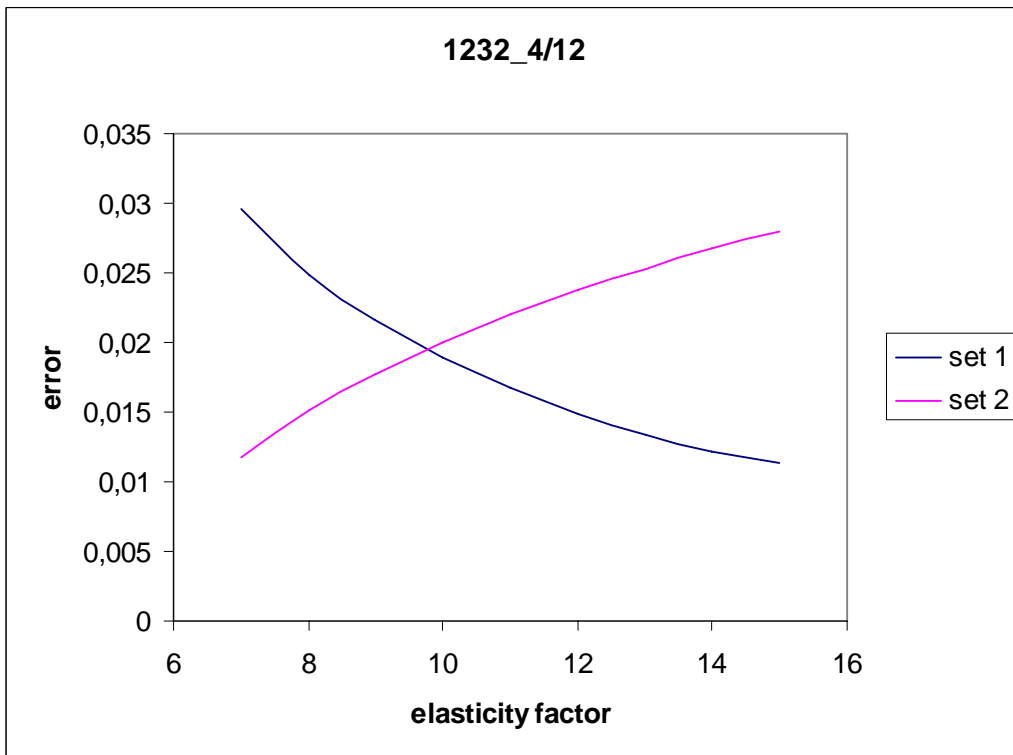
Σχήμα (3.25α): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 1/12



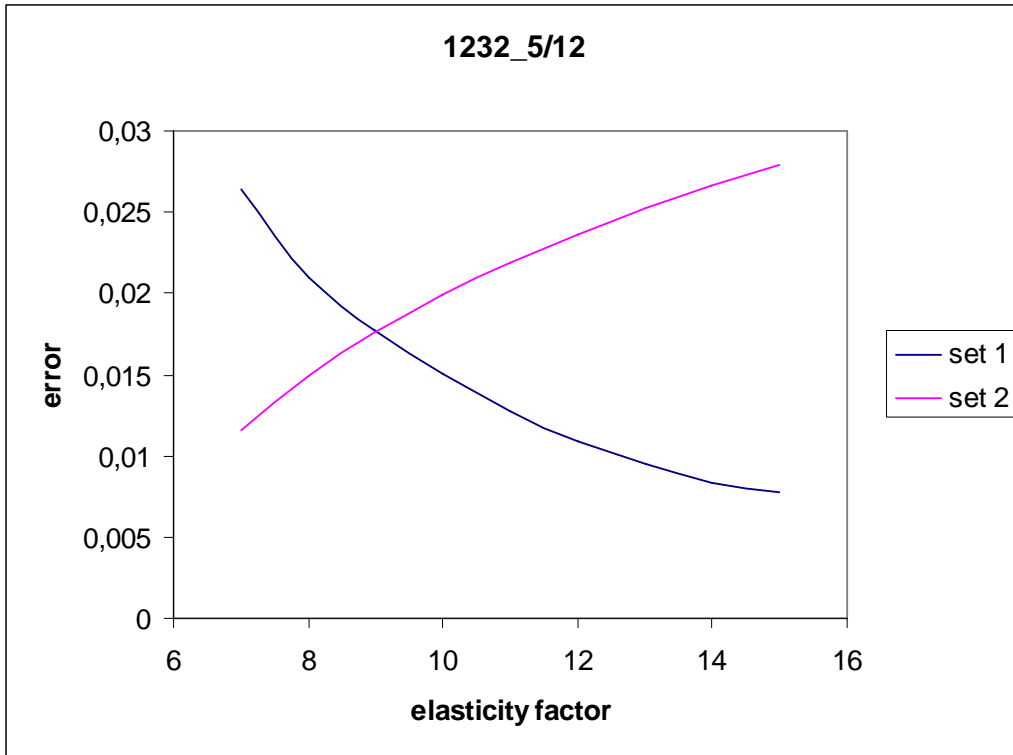
Σχήμα (3.25β): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 2/12



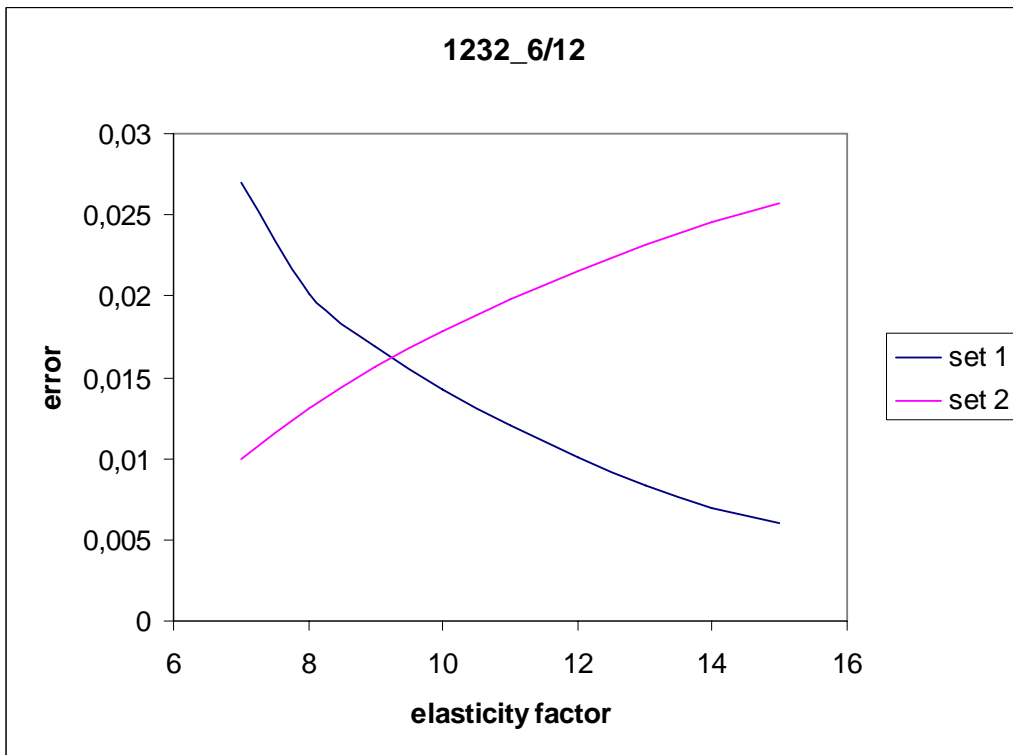
Σχήμα (3.25γ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 3/12



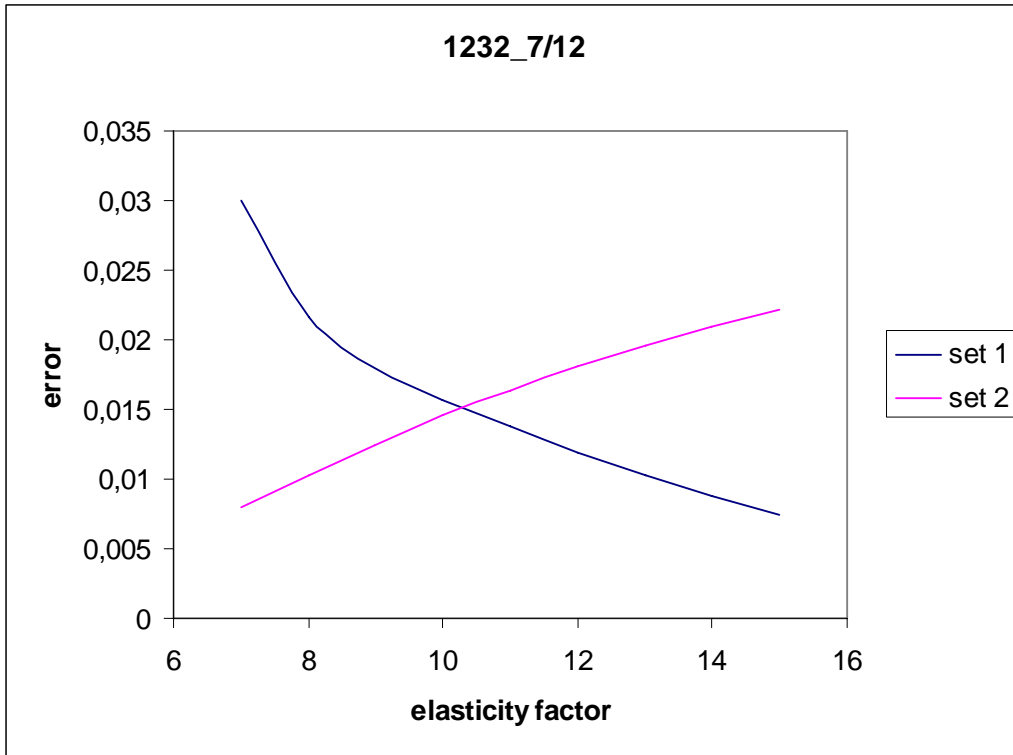
Σχήμα (3.25δ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 4/12



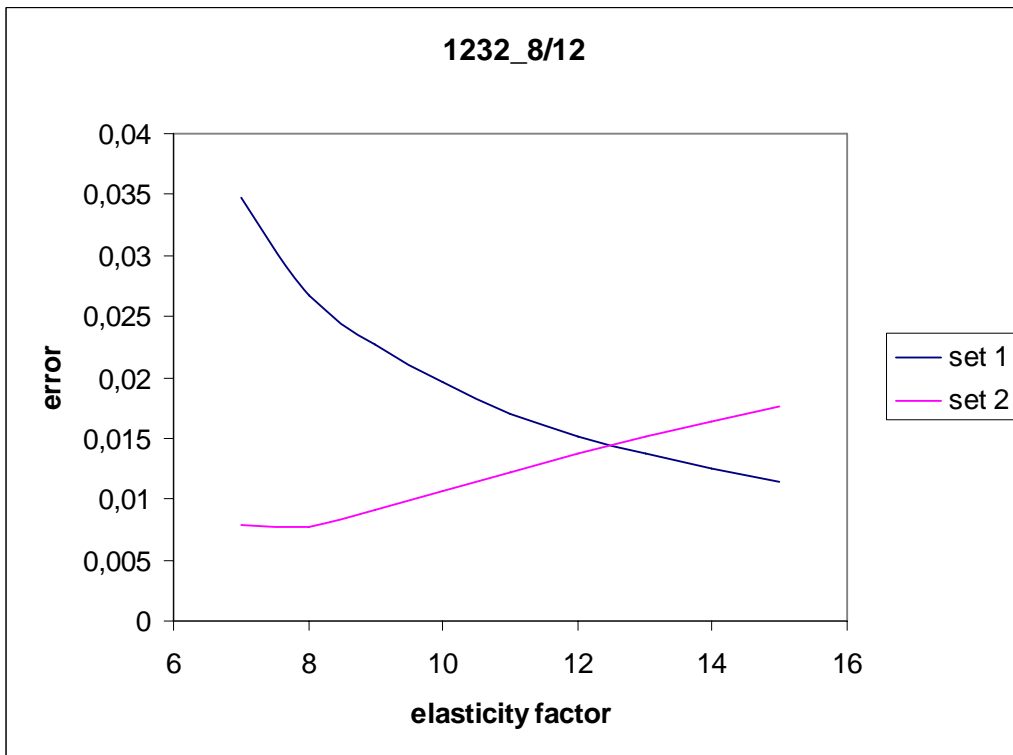
Σχήμα (3.25ε): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 5/12



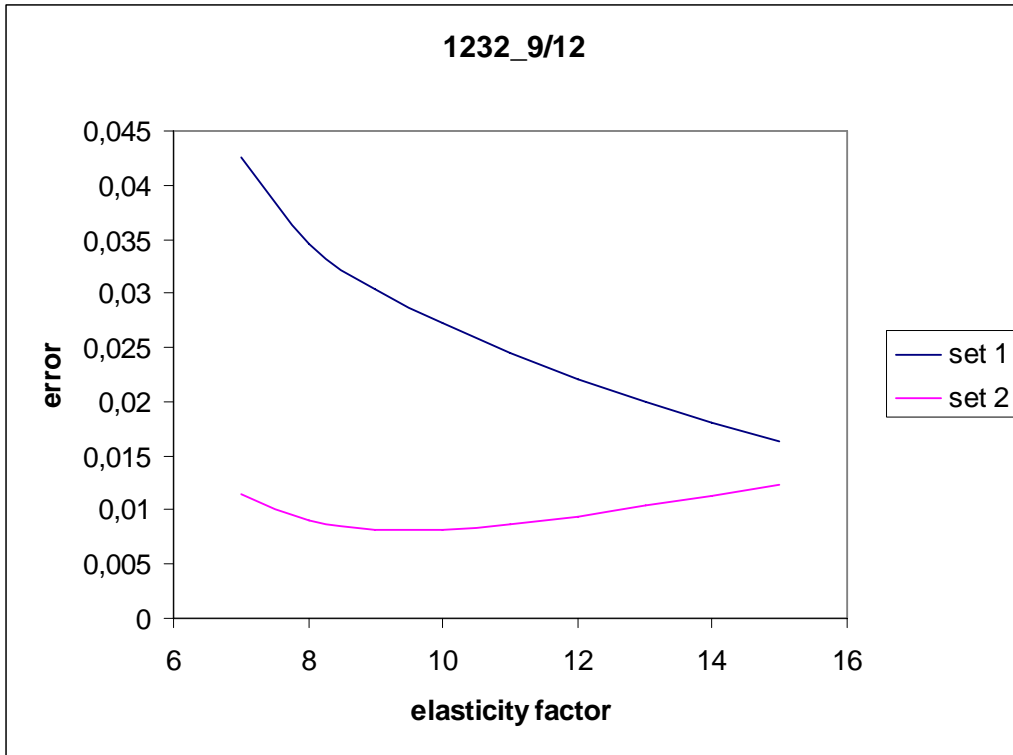
Σχήμα (3.25στ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 6/12



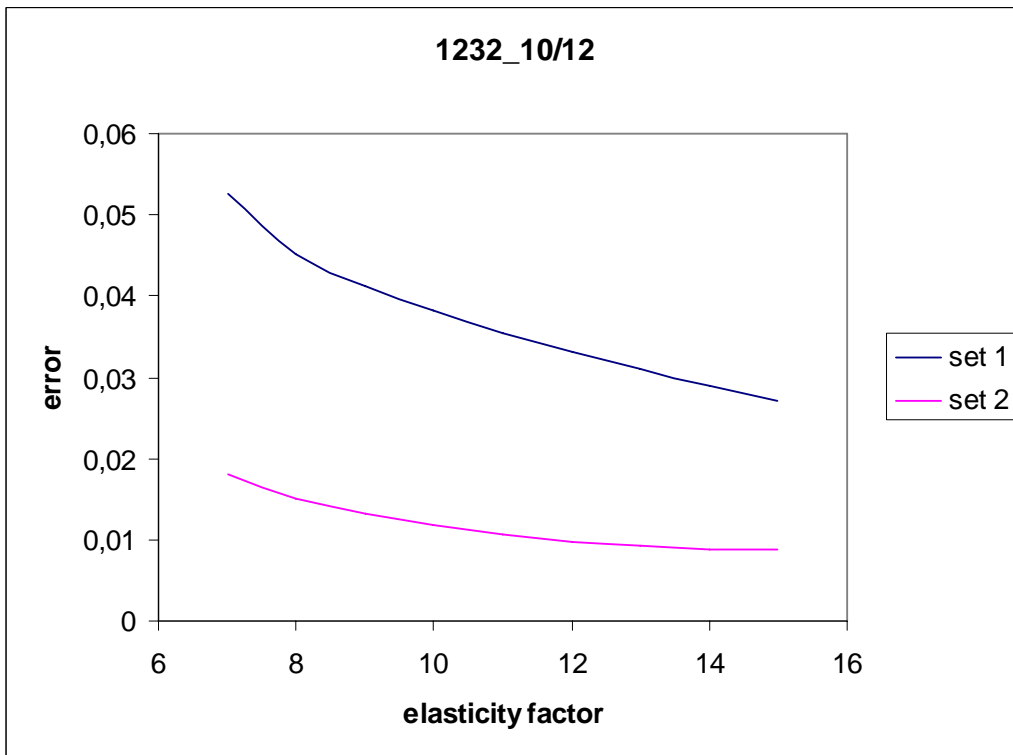
Σχήμα (3.25ζ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 7/12



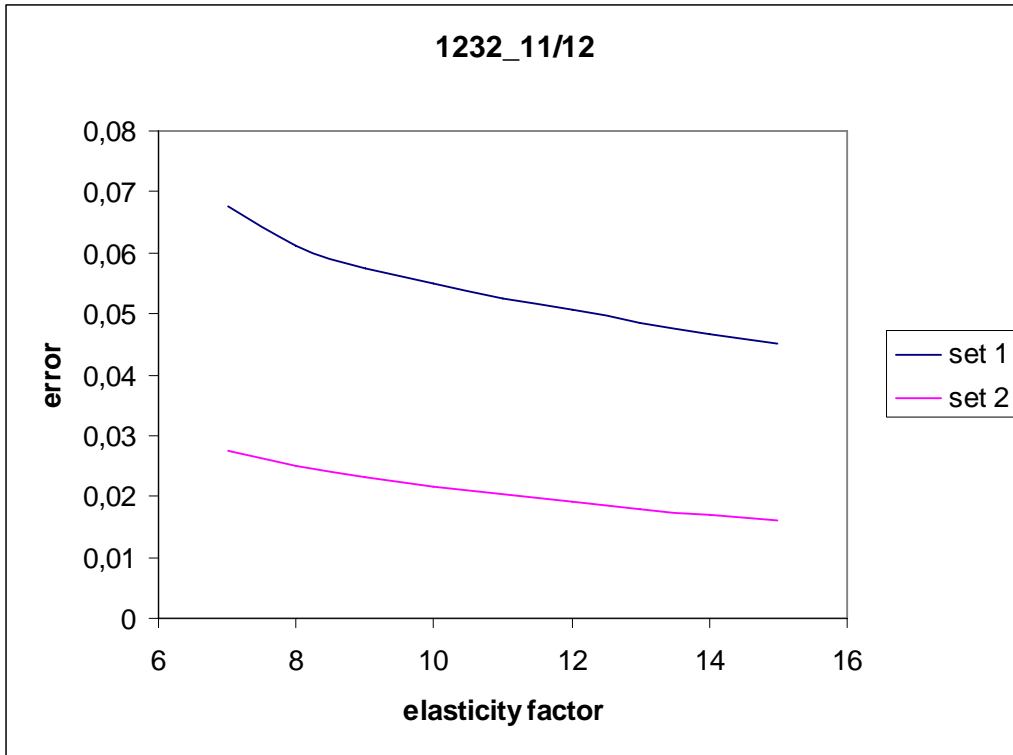
Σχήμα (3.25η): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 8/12



Σχήμα (3.25θ): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 9/12



Σχήμα (3.25ι): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 10/12



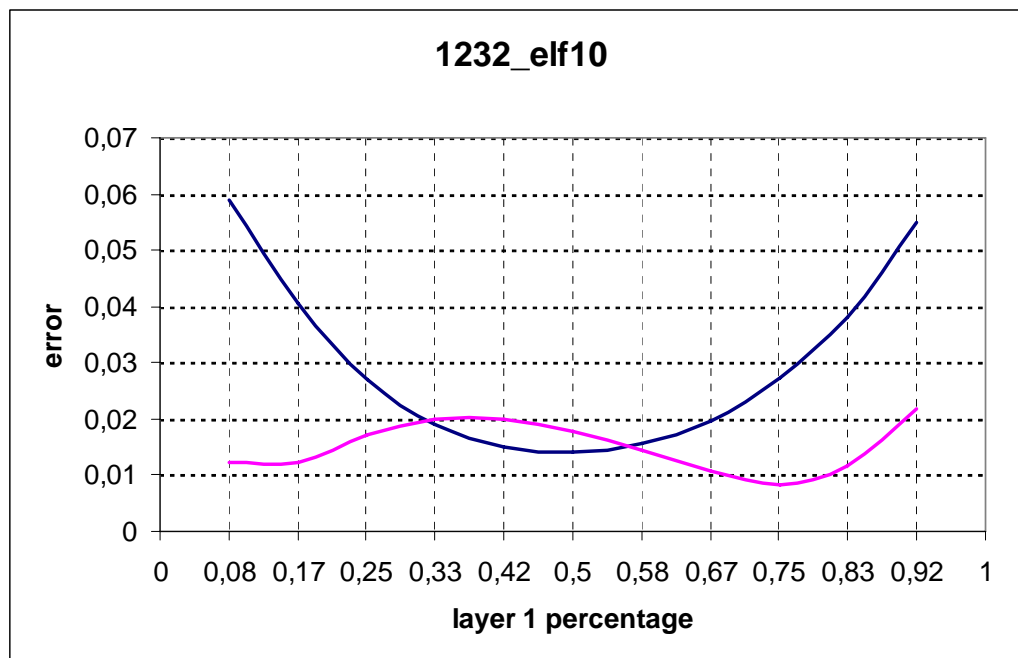
Σχήμα (3.25α): Διάγραμμα σφάλματος για πάχος πρώτου στρώματος 11/12

Πάλι καταλήγουμε σε μία τιμή του λόγου ελαστικότητας ίση με 10. Έτσι, προκύπτει ο παρακάτω πίνακας για τις διάφορες τιμές του πάχους του πρώτου στρώματος, και για τις δύο ομάδες δεδομένων.

	elasticity factor 10 between two layers										
layer1 %	0,0833	0,1667	0,2500	0,3333	0,4167	0,5000	0,5833	0,6667	0,7500	0,8333	0,9167
set 1 error	0,0590	0,0406	0,0271	0,0190	0,0150	0,0143	0,0157	0,0196	0,0272	0,0382	0,0549
set 2 error	0,0123	0,0123	0,0173	0,0200	0,0199	0,0179	0,0145	0,0106	0,0082	0,0117	0,0218

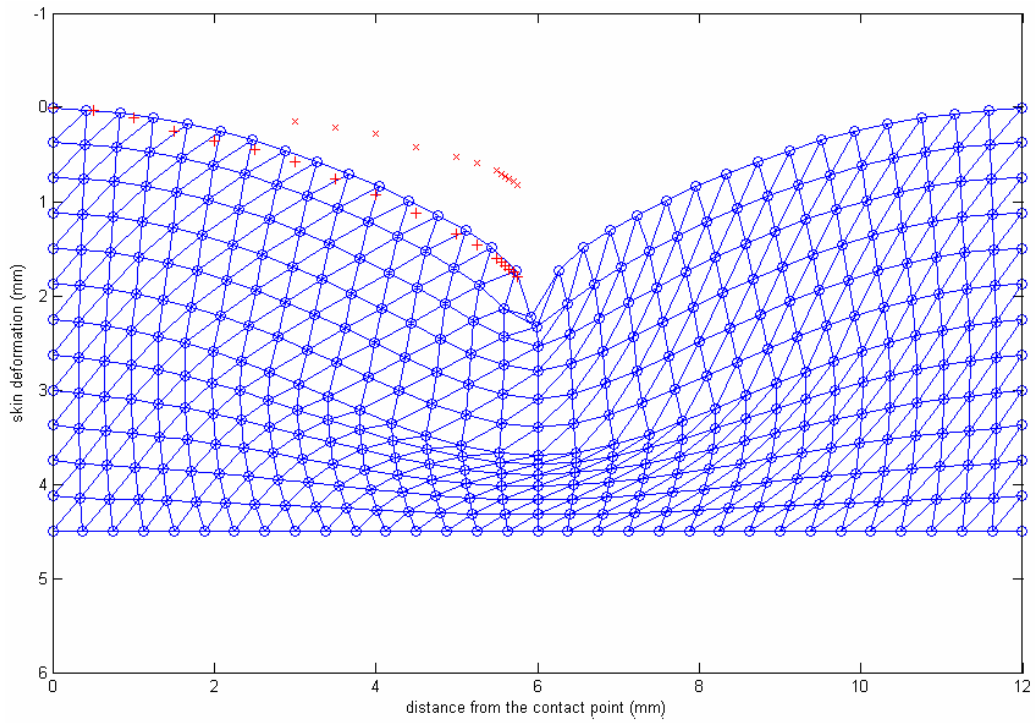
Πίνακας (3.8): Μέσο τετραγωνικό σφάλμα για λόγο ελαστικότητας 10

Αντίστοιχα έχουμε το παρακάτω διάγραμμα:

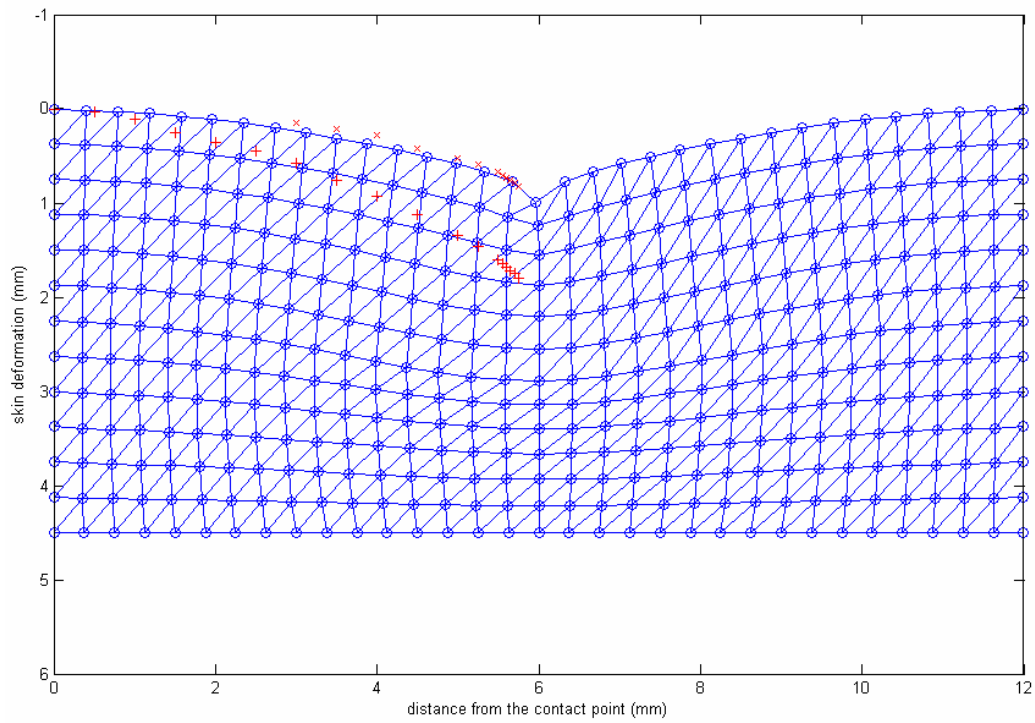


Σχήμα (3.26): Διάγραμμα σφάλματος για λόγο ελαστικότητας 10

Και πάλι το αποτέλεσμα παραμένει το ίδιο. Τα δύο στρώματα θα πρέπει να έχουν το ίδιο πάχος και το δεύτερο στρώμα να είναι 10 φορές πιο ελαστικό από το πρώτο. Συμπερασματικά, μπορούμε να πούμε ότι το αποτέλεσμα αυτό είναι ανεξάρτητο του βαθμού κατάτμησης. Έτσι, ανάλογα με τα περιθώρια που έχουμε για αυξημένη πολυπλοκότητα μπορούμε να χρησιμοποιήσουμε ένα πιο πυκνό σε πεπερασμένα στοιχεία μοντέλο, με μόνη διαφορά την πιο ομαλή εμφάνιση των καμπύλων επιφανειών. Παρακάτω φαίνεται το αποτέλεσμα αυτό:



Σχήμα (3.27α): Το βέλτιστο μοντέλο (1^ο σεντ δεδομένων)



Σχήμα (3.27β): Το βέλτιστο μοντέλο (2^ο σεντ δεδομένων)

Κεφάλαιο 4

Υλοποίηση σε σύστημα διαδραστικής προσομοίωσης

4.1 Υλοποίηση σε πραγματικό χρόνο

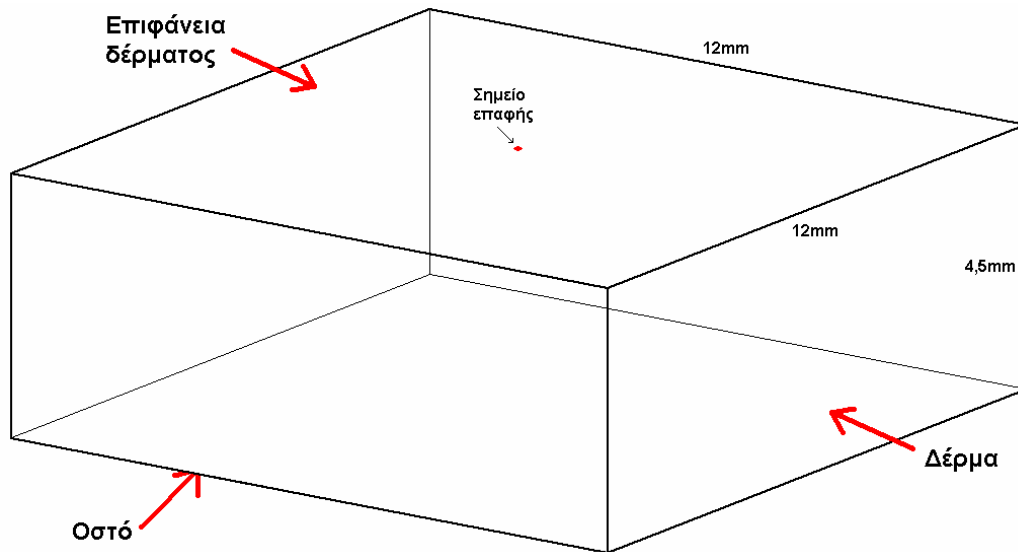
Με τη διαδικασία που ακολουθήσαμε μέχρι τώρα καταφέραμε να βρούμε τις παραμέτρους του διδιάστατου μοντέλου που αναζητούσαμε. Ωστόσο, παραμένουν ακόμη δύο προβλήματα. Το πρώτο αφορά στην απεικόνιση σε τρεις διαστάσεις και το δεύτερο στην προσομοίωση σε πραγματικό χρόνο.

Την απάντηση και στα δύο αυτά προβλήματα τη δίνει η εφαρμογή της OpenGL. Η απεικόνιση στις τρεις διαστάσεις θα πραγματοποιηθεί με χρήση της αξονικής συμμετρίας. Το διδιάστατο μοντέλο μπορεί εύκολα να μετατραπεί σε τρισδιάστατο. Για την προσομοίωση σε πραγματικό χρόνο είναι απαραίτητη η μετατροπή του κώδικα σε γλώσσα C. Το νέο πρόγραμμα θα πρέπει να δέχεται ως είσοδο το ζητούμενο βάθος διείσδυσης και να δίνει στην έξοδο ως αποτέλεσμα τις θέσεις των κόμβων των πεπερασμένων στοιχείων του μοντέλου. Τα στοιχεία αυτά θα χρησιμοποιούνται στη συνέχεια για την τρισδιάστατη απεικόνιση.

Για την κωδικοποίηση αυτή δεν έχουμε καμία διαφοροποίηση. Χρειάζεται μόνο η υλοποίηση κάποιων επιπλέον συναρτήσεων, όπως για παράδειγμα της επίλυσης γραμμικού συστήματος με πολύ μεγάλους πίνακες, οι οποίες δεν υπάρχουν έτοιμες σε γλώσσα C. Προσοχή πρέπει επίσης να δοθεί στην επιλογή του τύπου δεδομένων και την κωδικοποίηση ώστε η υλοποίηση να είναι αποδοτική και να επιτρέπει την προσομοίωση σε πραγματικό χρόνο. Ο κώδικας βρίσκεται στο παράρτημα.

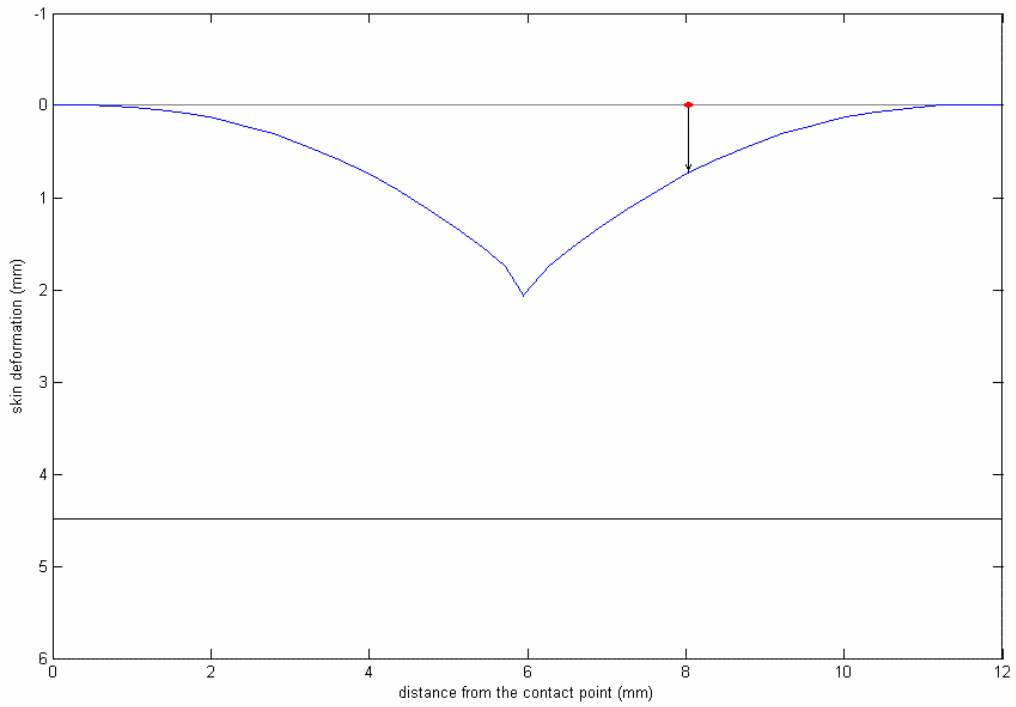
4.2 Δημιουργία τρισδιάστατου μοντέλου από το διδιάστατο

Για την απεικόνιση σε τρεις διαστάσεις θεωρούμε αρχικά ένα ορθογώνιο παραλληλεπίπεδο το οποίο προσομοιώνει ένα τμήμα δέρματος. Η επάνω επιφάνεια έχει διαστάσεις 12mm x 12mm και αποτελεί το τμήμα της επιφάνειας της επιδερμίδας που παραμορφώνεται, με το σημείο επαφής στο κέντρο. Η παράλληλη κάτω επιφάνεια είναι το οστό το οποίο βρίσκεται σε βάθος 4,5mm.

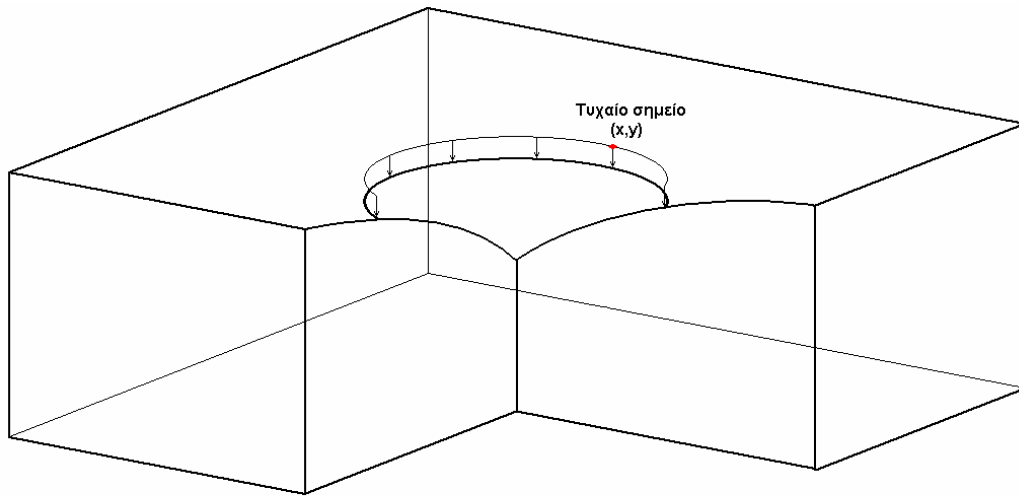


Σχήμα (4.1): Τρισδιάστατο μοντέλο

Η κάτω επιφάνεια είναι άκαμπτη (αποτελείται από τα ακλόνητα σημεία του οστού). Επίσης, εφόσον η παραμόρφωση τόσο της επιφάνειας όσο και του εσωτερικού δε διαδίδεται σε απόσταση μεγαλύτερη των 6mm από το κέντρο, οι πλευρικές επιφάνειες είναι επίσης άκαμπτες. Για τον προσδιορισμό της θέσης των σημείων της επιφάνειας χρησιμοποιούμε την αξονική συμμετρία. Έτσι για τυχαίο σημείο της επιφάνειας του δέρματος υπολογίζουμε την απόστασή του (ακτινικά) από το κέντρο, το σημείο επαφής. Στη συνέχεια βρίσκουμε για τη δεδομένη απόσταση από το κέντρο, το βάθος όπως ορίζεται από το διδιάστατο μοντέλο. Το βάθος αυτό θα είναι και το βάθος του σημείου μας.



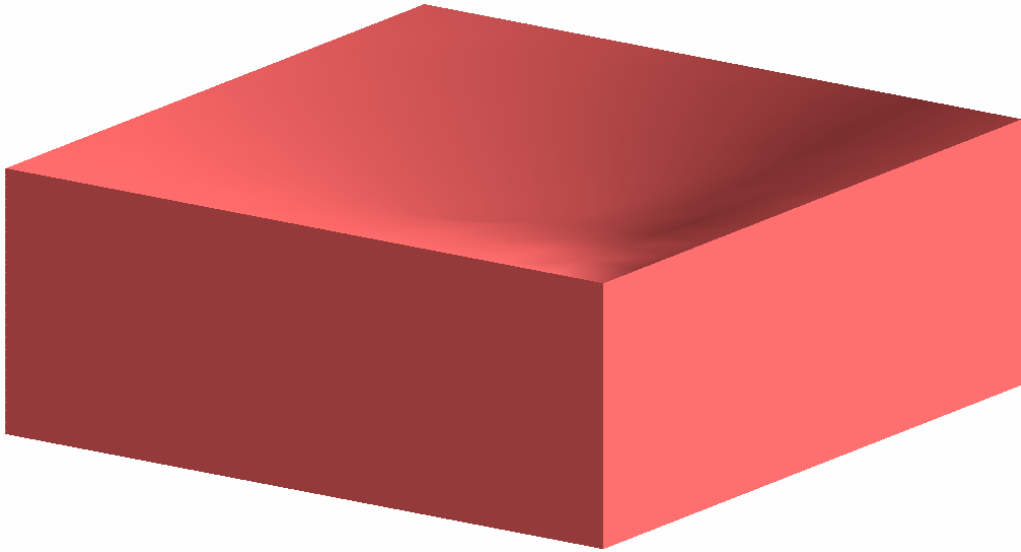
Σχήμα (4.2): Παράδειγμα μεταβολής βάθους τυχαίου σημείου της επιφάνειας στο διδιάστατο μοντέλο



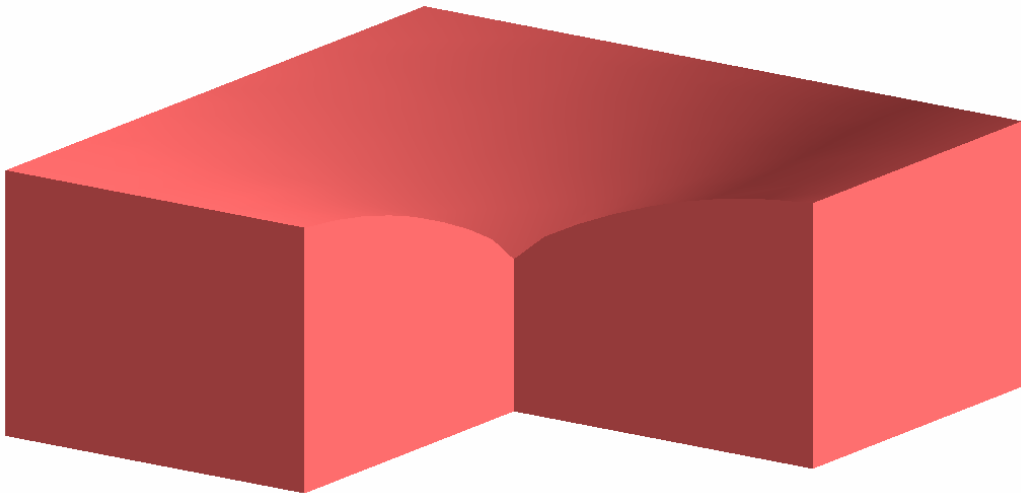
Σχήμα (4.3): Προσδιορισμός βάθους τυχαίου σημείου της επιφάνειας σε τρεις διαστάσεις

4.3 Τρισδιάστατο μοντέλο σε γραφικό περιβάλλον πραγματικού χρόνου

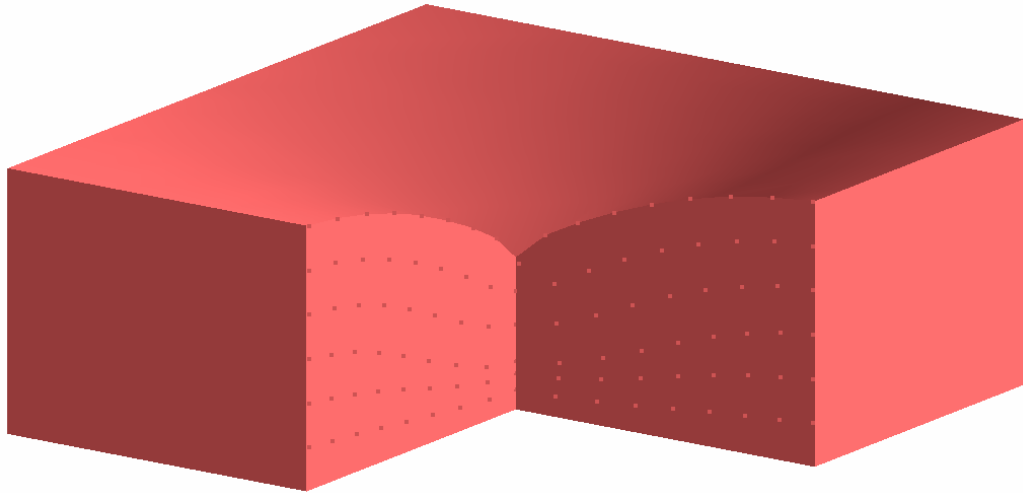
Στα σχήματα που ακολουθούν βλέπουμε το αποτέλεσμα σε τρεις διαστάσεις με εφαρμογή της υλοποίησης σε πραγματικό χρόνο.



Σχήμα (4.4α): Τμήμα δέρματος στον οποίο το κέντρο ασκείται σημειακή δύναμη



Σχήμα (4.4β): Τομή για καλύτερη απεικόνιση

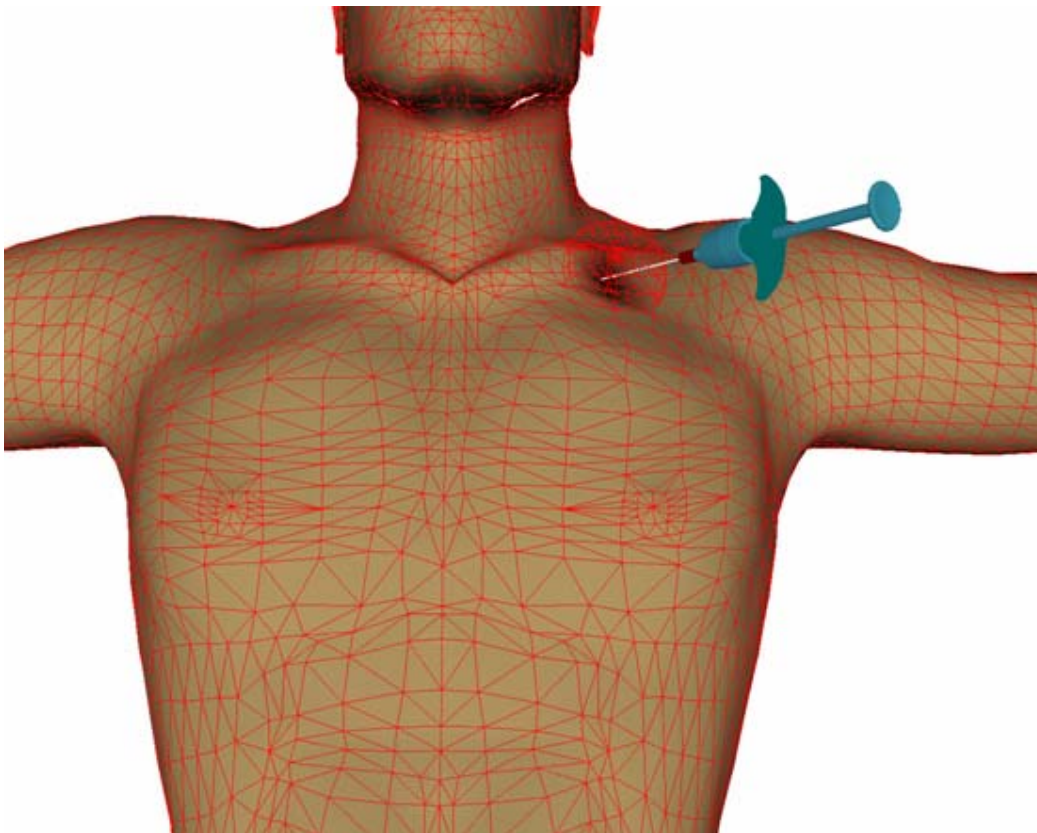


Σχήμα (4.4γ): Αναπαράσταση του εσωτερικού με εμφάνιση των κόμβων των πεπερασμένων στοιχείων

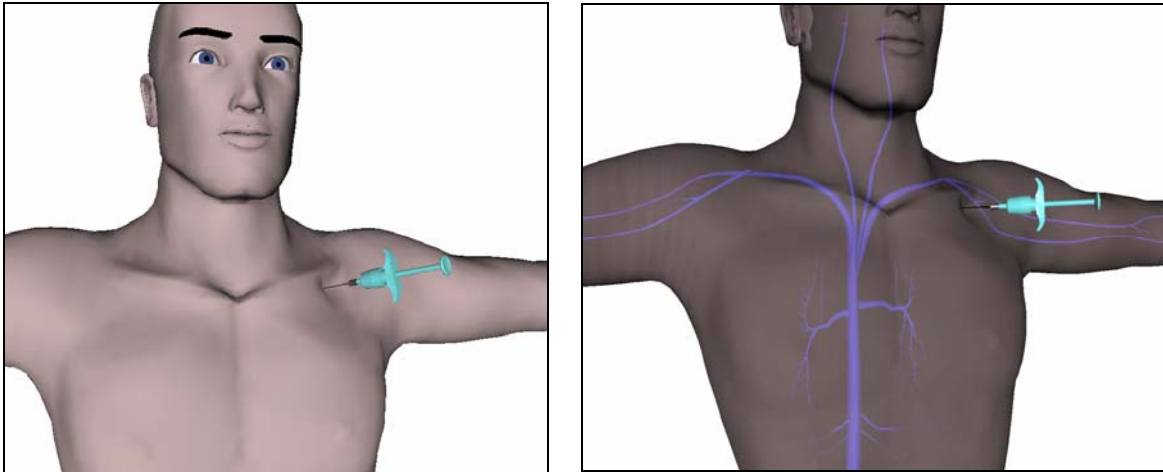
Για να προκύψουν τα σχήματα αυτά έχουν υπολογιστεί τα κάθετα διανύσματα – normals της επιφάνειας της δομής για το rendering της OpenGL. Ωστόσο, δεν εστιάστηκε η οπτική απόδοση για την απεικόνιση της δομής (shading, lighting).

4.4 Εφαρμογή σε παρακέντηση

Εφαρμογή της μεθόδου που ακολουθήσαμε για τη μοντελοποίηση της παραμόρφωσης του δέρματος από βελόνα φαίνεται στα παρακάτω σχήματα. Το αποτέλεσμα που βλέπουμε είναι σε τρεις διαστάσεις.



Σχήμα (4.5): Προσομοίωση επιφάνειας δέρματος με πεπερασμένα στοιχεία. Προσδιορισμός σημείου επαφής της βελόνας [10]



Σχήμα (4.6): Παραμόρφωση δέρματος – εικόνα σε ημιδιαφανές μοντέλο [10]



Σχήμα (4.7): Υλοποίηση σε πραγματικό χρόνο – Ανάδραση με χρήση ειδικής συσκευής. [10]

Συμπεράσματα – Μελλοντική Εργασία

Στην παρούσα εργασία μελετήθηκε η παραμόρφωση της επιφάνειας του δέρματος κατά την εφαρμογή κατακόρυφης σημειακής δύναμης σε αυτό. Κατασκευάστηκε μοντέλο τομής δέρματος σε δύο διαστάσεις, αποτελούμενο από δύο διαδοχικά στρώματα διαφορετικού πάχους και διαφορετικής ελαστικότητας. Για τη μοντελοποίηση επιλέχθηκε η μέθοδος των πεπερασμένων στοιχείων, ενώ οι παράμετροι προσαρμόστηκαν ώστε να υπάρχει ανταπόκριση σε πειραματικά δεδομένα της διεθνούς βιβλιογραφίας. Υπολογίστηκε ότι τα δύο στρώματα πρέπει να έχουν ίδιο πάχος ενώ η σχετική τους ελαστικότητα να είναι 10, με το κατώτερο στρώμα 10 φορές πιο ελαστικό από το ανώτερο. Στη συνέχεια η διδιάστατη δομή η οποία είχε υλοποιηθεί σε matlab επεκτάθηκε και στις τρεις διαστάσεις με χρήση της αξονικής συμμετρίας. Το τρισδιάστατο μοντέλο λειτουργεί σε πραγματικό χρόνο και παρέχει στο χρήστη τη δυνατότητα αλλαγής του βάθους διείσδυσης. Για να επιτευχθεί αυτό πραγματοποιήθηκε εκ νέου υλοποίηση σε γλώσσα C/C++ με γραφικό περιβάλλον OpenGL.

Η εργασία αυτή αποτελεί τμήμα επίλυσης ενός μεγαλύτερου προβλήματος. Οι παράμετροι του μοντέλου που προσδιορίστηκαν βρίσκουν εφαρμογή στην προσομοίωση μιας παρακέντησης. Στο σύστημα που δημιουργείται υπάρχει και η αίσθηση της αφής, ο χρήστης αισθάνεται τη δύναμη από την ανάδραση του σώματος κατά την επαφή ή τη μετακίνηση της βελόνας με χρήση ειδικού οργάνου συνδεδεμένου με τον υπολογιστή. Για να επιτευχθεί αυτό εκτός από το κομμάτι της απεικόνισης είναι απαραίτητες και άλλες διαδικασίες όπως ο υπολογισμός των δυνάμεων της ανάδρασης του συστήματος, ή η ανίχνευση σύγκρουσης (collision detection) για τον προσδιορισμό του σημείου επαφής.

Απτικά συστήματα αυτής της μορφής εμφανίζουν ιδιαίτερο ενδιαφέρον. Στο προσεχές μέλλον σχεδιάζεται η ενσωμάτωσή τους σε μια πλατφόρμα ιατρικής εκπαίδευσης μέσω υπολογιστή. Το περιβάλλον αυτό της εικονικής πραγματικότητας είναι ιδανικό για την άσκηση κλινικών δεξιοτήτων σχετικών με την παρακέντηση. Η προσομοίωση καταστάσεων και διαδικασιών αποτελεί σίγουρα ένα κομμάτι που βρίσκεται σε μεγάλο βαθμό μόνο σε θεωρητικό επίπεδο. Αυτό δυσχεραίνει κατά πολύ

την εκπαίδευση νέων ιατρών αφού είναι απαραίτητη η εφαρμογή σε ανθρώπους όπου δεν υπάρχουν περιθώρια σφαλμάτων. Η απόκτηση εμπειρίας καθίσταται πολύ επίπονη και χρονοβόρα.

Από τη στιγμή που θα γίνουν τα πρώτα βήματα στην ενσωμάτωση της εικονικής πραγματικότητας στις ιατρικές εφαρμογές είναι σίγουρο ότι θα υπάρξει ραγδαία πρόοδος. Εξάλλου, το ενδιαφέρον είναι στραμμένο προς την κατεύθυνση αυτή και η υποδομή από τεχνικής απόψεως είναι διαθέσιμη. Τα συστήματα αφής βρίσκονται ήδη σε εξέλιξη η επεξεργαστική ισχύς είναι ταχύτατα αυξανόμενη, παρέχοντας τη δυνατότητα παράλληλης επεξεργασίας πολλών δεδομένων, την ενσωμάτωση πολλών λεπτομερειών. Αυτό που απομένει είναι το προγραμματιστικό κομμάτι, το οποίο έχει αυξημένη δυσκολία, αφού για την υλοποίηση θα πρέπει να συνυπολογιστούν πολλές παράμετροι. Το αποτέλεσμα πρέπει να ανταποκρίνεται τόσο οπτικά όσο και απτικά στην πραγματικότητα.

Οι δυνατότητες που θα παρέχουν συστήματα αυτής της μορφής σίγουρα δε θα περιοριστούν στον εκπαιδευτικό τομέα. Εφόσον ληφθούν τα απαραίτητα πειραματικά δεδομένα και ενσωματωθούν αρκετές λειτουργίες και μηχανισμοί του ανθρώπινου σώματος, το έδαφος θα είναι πρόσφορο για τη διεξαγωγή πειραμάτων σε εικονικό περιβάλλον. Ο πειραματισμός σε ανθρώπινα σώματα που αποτελεί μέχρι τώρα περιοριστικό παράγοντα, θα γίνει ευκολότερος καθώς τα μοντέλα των ανθρώπινων οργανισμών θα τελειοποιούνται. Για να επιτευχθεί αυτό θα χρειαστεί φυσικά η λήψη πολλών μετρήσεων και το μοντέλο θα είναι πολύ γενικό, δε θα ταιριάζει απόλυτα σε κάθε άνθρωπο.

Αν κοιτάξουμε λίγο πιο μακριά, μπορεί στο μέλλον να ενσωματωθούν συστήματα αναγνώρισης ανθρώπινων χαρακτηριστικών ή και μέτρησης παραμέτρων στο γενικότερο μοντέλο. Έτσι με την εισαγωγή ενός ασθενούς, αφού πραγματοποιηθούν οι κατάλληλες εξετάσεις θα μπορεί αυτόματα να προσομοιωθεί το σώμα του με τις ιδιαιτερότητές του σε εικονικό περιβάλλον. Στη συνέχεια θα είναι πολύ εύκολο να γίνουν δοκιμές για τον προσδιορισμό της βέλτιστης διαδικασίας θεραπευτικής αγωγής. Τα αποτελέσματα, ακόμη και μακροχρόνια θα μπορούν να προσομοιωθούν σε ελάχιστο χρόνο και να προσδιοριστούν με μεγάλη ακρίβεια. Ένα περίπλοκο συνθλιπτικό κάταγμα για παράδειγμα που απαιτεί σήμερα έναν έμπειρο χειρισμό θα μπορεί να προσομοιωθεί εύκολα σε εικονικό περιβάλλον. Η πορεία της θεραπείας θα μπορεί να προσδιοριστεί άμεσα και με ακρίβεια από την πρώτη στιγμή, χωρίς την ανάγκη μεγάλης ιατρικής εμπειρίας. Η προσομοίωση θα παρέχει τα απαραίτητα στοιχεία και αποτελέσματα.

Σίγουρα στον τομέα της υγείας οι ανάγκες είναι άκορεστες. Όσο περισσότερα τα εργαλεία, όσο μεγαλύτερες οι δυνατότητες, τόσο πιο πολλά και τα ερωτηματικά. Το ενδιαφέρον θα είναι πάντα στραμμένο σε αυτή την κατεύθυνση. Η εικονική πραγματικότητα είναι ήδη ένα πολλά υποσχόμενο εργαλείο για την εξέλιξη των

μεθόδων διάγνωσης ή θεραπείας και θα αποτελέσει σίγουρα τμήμα μελλοντικής αναζήτησης.

Παράρτημα

Ο κώδικας

1 Κώδικας matlab

```
clear all
close all
% It is defined that x axis is the horizontal axis, (x>0 right) and
% y axis is the vertical axis, (y>0 down)
% The first node (top left) is in position 0,0 and the last node (bottom
% right) is in position (noi-1)*L,(noj-1)*L

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Deformation data from [9]
%%% 1st set of deformation data: distance dx from the load (in mm)
ddx1 = [0.25;0.30;0.35;0.40;0.45;0.50;0.75;1.0;1.5;2.0;2.5;3.0;3.5;4.0;4.5;5.0;5.5;6.0];
%%% 1st set of deformation data: vertical skin displacement (in mm) for 2mm depth of
%%% indentation, measurements on the figures of the manuscript
ddef1 = [0.0;
        0.18;
        0.30;
        0.43;
        0.55;
        0.70;
        1.20;
        1.60;
        2.42;
        3.11;
        3.70;
        4.35;
        4.80;
        5.15;
        5.50;
        6.00;
        6.30;
        6.40];
%%% 1st set of deformation data: vertical skin displacement dy (in mm),
%%% scaling/transformation to obtain original measures
dscale = 0.45/1.6;
ddy1 = -(ddef1*dscale - 1.8);
[sddx1,n] = size(ddx1);
```

```

%% 2nd set of deformation data: distance dx from the load (in mm)
ddx2 = [0.25;0.30;0.35;0.40;0.45;0.50; 0.75;1.0;1.5;2.0;2.5;3.0]; %3.5;4.0;4.5;5.0;5.5;6.0];
%% 2nd set of deformation data: vertical skin displacement (in mm) for 1mm depth of
%% indentation, measurements on the figures of the manuscript
ddef2 = [0.25;
         0.40;
         0.50;
         0.60;
         0.70;
         0.80;
         1.10;
         1.35;
         1.70;
         2.20;
         2.45;
         2.65
        ];
%% 2nd set of deformation data: vertical skin displacement dy (in mm),
%% scaling/transformation to obtain original measure scale = 0.45/1.6;
ddy2 = - (ddef2*dscale - 0.9);
[sddx2,n] = size(ddx2);
%%
%%
%% Number of half of the nodes in x direction (horizontal) (from the first
%% to the center node (the rest of the nodes are symmetric and their position
%% needs not be evaluated).
noi=17;
nload = (noi+1)/2; %% node to be loaded (middle of the structure)
%% Number of the nodes in y direction (vertical)
noj=7; %%: num_of_layers + 1
nlayers = noj-1;
%% Lx: Total Length of structure in x dimension (in mm)
Lx = 12;
%% L: Length of each node (in mm)
L=Lx/(noi-1); %%
%% The middle of these nodes shall be forced to move down (y direction) distance dist (in mm) (dist>0)
scale_y = 1.0; %5.0*dist/(nlayers*L); %%scale_factor for y direction
%dist = dist/scale_y;

%% Area of each trinagular element
A=L*L/2;
%% Additional parameters
E = 0.2; %% MPa %%Elasticity (Young's) modulus of epidermis
v = 0.48; %% Poisson ratio

%% The global stiffness matrix will be of size (noi*noj,noi*noj) Each of both
%% dimensions is equal with the total number of nodes
GK(:,:)=zeros(2*noi*noj,2*noi*noj);

%% The global force vector will be a column of size 2*noi*noj
GF(:,:)=zeros(2*noi*noj,1);

%% The global unknown node vector matrix will be a column of size 2*noi*noj
GU(:,:)=ones(2*noi*noj,1);
GU = GU .* (-1); % Value -1 equals to unknown value

Cr=[0 0 0 0 0 0]; % correspondance matrix (used during the calculation of GK, GF)

```

```

% Definition of matrix B1 for the first element in a square
B1 = [L 0 0 -L 0
      0 L 0 -L 0 0
      L L -L 0 0 -L];
B1 = B1 ./ (2*A);

```

```

% Definition of inverted B1, BT1
BT1 = [L 0 L
       0 L L
       0 0 -L
       0 -L 0
       -L 0 0
       0 0 -L];
BT1 = BT1 ./ (2*A);

```

```

% Definition of matrix B2 for the second element in a square
B2 = [-L 0 0 0 L 0
      0 -L 0 L 0 0
      -L -L L 0 0 L];
B2 = B2 ./ (2*A);

```

```

% Definition of inverted B2, BT2
BT2 = [-L 0 -L
       0 -L -L
       0 0 L
       0 L 0
       L 0 0
       0 0 L];
BT2 = BT2 ./ (2*A);

```

```

% Definition of matrix D (stress type)
D = [1 v 0
     v 1 0
     0 0 (1-v)/2];
D = D .* (E/(1+v*v));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                L O O P
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

for set = 1:2,
if set == 1,
    dist = 1.8; %(in mm)
    sddx = sddx1;
    ddx = ddx1;
    ddy = ddy1;
end
if set == 2,
    dist = 0.9;
    sddx = sddx2;
    ddx = ddx2;
    ddy = ddy2;
end
end

```

```

j_layer1 = 3; % j_layer1/num_of_layers = layer 1 thickness percentage
el_f = 10; % elasticity factor between 2 layers

%2nd layer --> el_f times greater elasticity modulus
Delf = D/el_f;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creation of each stiffness matrix (K) and addition to the global (GK)
% Creation of each force matrix (F) and addition to the global (GF)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1 : noi-1
    for j = 1 : noj-1 % for every sqare (each square contains 2 elements)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % First element in the square [ (i,j), (i,j+1), (i+1,j+1), (i+1,j) ]
        % For this element node 1 is at the coordinates (i,j) node 2 at
        % (i,j+1) and node 3 at (i+1,j)
        D1 = D; % first layer
        if j > j_layer1,
            D1 = Delf; % second layer
        end
        % Stiffness matrix K1 of the element
        K1 = BT1 * D1 * B1;
        Cr(1)=2*((i-1)*noj+j)-1;
        Cr(2)=2*((i-1)*noj+j);
        Cr(3)=2*((i-1)*noj+j+1)-1;
        Cr(4)=2*((i-1)*noj+j+1);
        Cr(5)=2*[i*noj+j]-1;
        Cr(6)=2*[i*noj+j];
        for k = 1:6
            for l = 1:6
                GK(Cr(k),Cr(l))=GK(Cr(k),Cr(l))+K1(k,l);
            end
        end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Second element in the square [ (i,j), (i,j+1), (i+1,j+1), (i+1,j) ]
        % For this element node 1 is at the coordinates (i+1,j+1) node 2 at
        % (i+1,j) and node 3 at (i,j+1)
        D2 = D; % first layer
        if j > j_layer1,
            D2 = Delf; % second layer
        end
        % Stiffness matrix K2 of the element using [5.60]
        K2 = BT2 * D2 * B2;
        Cr(1)=2*[i*noj+j+1]-1;
        Cr(2)=2*[i*noj+j+1];
        Cr(3)=2*[i*noj+j]-1;
        Cr(4)=2*[i*noj+j];
        Cr(5)=2*((i-1)*noj+j+1)-1;
        Cr(6)=2*((i-1)*noj+j+1);
        for k = 1:6
            for l = 1:6
                GK(Cr(k),Cr(l))=GK(Cr(k),Cr(l))+K2(k,l);
            end
        end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% At this point matrixes GF and GK have been created
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

%%%%%%%%%%
% Step1 :
% Calculation of disterr (error in distance pushing down the middle element)
%%%%%%%%%%

GF_tmp1 = GF;
GK_tmp1 = GK;

%%%%%%%%%%
% Boundary conditions
% Node (nload,1) with coordinates ((nload-1)*L,0)
% is pushed down at distance dist
i=nload;
j=1;
GU(2*((i-1)*noj+j)) = (j-1)*L + dist;
% Every node with coordinates x=0 cannot be moved
i=1;
for j = 1:noj
    GU(2*((i-1)*noj+j)-1) = 0;    %(i-1)*L;
    GU(2*((i-1)*noj+j)) = 0;      %(j-1)*L;
end
% Every node with coordinates y=(noj-1)*L cannot be moved
% These are all the nodes to the bottom.
j = noj;
for i = 1:noi
    GU(2*((i-1)*noj+j)-1) = 0;    %(i-1)*L;
    GU(2*((i-1)*noj+j)) = 0;      %(j-1)*L;
end
% Every node with coordinates x=(noi-1)*L cannot be moved
i=noi;
for j = 1:noj
    GU(2*((i-1)*noj+j)-1) = 0;    %(i-1)*L;
    GU(2*((i-1)*noj+j)) = 0;      %(j-1)*L;
end
%%%%%%%%%%

ndof = 2*noi*noj; %% ndof: total number of degrees of freedom in the FEM
%% index_dof_act[] <-- contains only the indices of active dof
%% (number of active dofs: ndof_act)
%% index_dof_all[] <-- contains -1 if this dof is fixed, or the index
%% of this dof in vector: index_dof_act[]
i2=0;
for i=1:ndof,
    if (GU(i) == -1) %%active d.o.f. (not fixed due to a boundary condition)
        i2=i2+1;
        index_dof_act(i2) = i;
        index_dof_all(i) = i2;
    else
        index_dof_all(i) = -1; %%d.o.f. fixed due to a boundary condition
    end
end
ndof_act = i2; %% ndof_act: number of active degrees of freedom in the FEM
GF2 = zeros(ndof_act,1);
for j = 1:ndof,
    if (index_dof_all(j) ~= -1) %% (GU(j) == -1)
        %%d.o.f. active (not fixed due to a boundary condition)

```

```

j2 = index_dof_all(j);
for i2 = 1:ndof_act,
    i = index_dof_act(i2);
    GK2(i2,j2) = GK(i,j);
end
else
%% (index_dof_all(j)~=1), (i.e. GU(j)~=1)
%%d.o.f. fixed due to a boundary condition
for i2=1:ndof_act,
    i = index_dof_act(i2);
    GF2(i2) = GF2(i2) - GK(i,j)*GU(j);
end
end
end

%% The system { GK2 * GU_act = GF2 } is ready to be solved using a known method
GU_act = inv(GK2)*GF2;
GU2_def = GU;
%%--> GU2_def contains "deviations" from "rest positions" of all respective dofs
for i2=1:ndof_act,
    i = index_dof_act(i2);
    GU2_def(i) = GU_act(i2);
end
GU2 = zeros(ndof,1);
%%--> GU2 contains new "deviated positions" of all respective dofs
for i=1:noi,
    for j=1:noj
        GU2(2*((i-1)*noj+j)-1) = (i-1)*L;
        GU2(2*((i-1)*noj+j)) = (j-1)*L;
    end
end
GU2 = GU2 + GU2_def;

%% Definition of Node Coordinates Matrix (NC)
for i = 1:noi*noj
    NC(i,1) = GU2(2*i-1);
    NC(i,2) = scale_y*GU2(2*i);
end

%% Calculation of distance error (disterr)
n = 1;
while (Lx/2-ddx(1) > NC(n+noj,1)),
    n = n + noj;
end
disterr = NC(n,2)-ddy(1)+(Lx/2-ddx(1)-NC(n,1))*(NC(n+noj,2)-NC(n,2))/(NC(n+noj,1)-NC(n,1));

%% Step2 :
%% Calculation of actual error and yerrtotal

```

```

GF = GF_tmp1;
GK = GK_tmp1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Boundary conditions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Node (nload,1) with coordinates ((nload-1)*L,0) is pushed down at distance dist
i=nload;
j=1;
GU(2*[(i-1)*noj+j]) = (j-1)*L + dist - disterr;
% Every node with coordinates x=0 cannot be moved
i=1;
for j = 1:noj
    GU(2*[(i-1)*noj+j]-1) = 0;    %(i-1)*L;
    GU(2*[(i-1)*noj+j]) = 0;      %(j-1)*L;
end
% Every node with coordinates y=(noj-1)*L cannot be moved
% These are all the nodes to the bottom.
j = noj;
for i = 1:noi
    GU(2*[(i-1)*noj+j]-1) = 0;    %(i-1)*L;
    GU(2*[(i-1)*noj+j]) = 0;      %(j-1)*L;
end
% Every node with coordinates x=(noi-1)*L cannot be moved
i=noi;
for j = 1:noj
    GU(2*[(i-1)*noj+j]-1) = 0;    %(i-1)*L;
    GU(2*[(i-1)*noj+j]) = 0;      %(j-1)*L;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ndof = 2*noi*noj; %% ndof: total number of degrees of freedom in the FEM
%% index_dof_act[] <-- contains only the indices of active dof
%% (number of active dofs: ndof_act)
%% index_dof_all[] <-- contains -1 if this dof is fixed,
%% or the index of this dof in vector: index_dof_act[]
i2=0;
for i=1:ndof,
    %%
    if (GU(i) == -1) %%active d.o.f. (not fixed due to a boundary condition)
        %%
        i2=i2+1;
        index_dof_act(i2) = i;
        index_dof_all(i) = i2;
    else
        index_dof_all(i) = -1; %%d.o.f. fixed due to a boundary condition
    end
end
ndof_act = i2; %% ndof_act: number of active degrees of freedom in the FEM
GF2 = zeros(ndof_act,1);
for j = 1:ndof,
    if (index_dof_all(j) ~= -1) %% (GU(j) == -1) %%d.o.f. active (not fixed due to a boundary condition)
        j2 = index_dof_all(j);
        for i2 = 1:ndof_act,
            i = index_dof_act(i2);
            GK2(i2,j2) = GK(i,j);
        end
    end
end

```

```

else
    GU(j)~=1) %%d.o.f. fixed due to a boundary condition
    for i2=1:ndof_act,
        i = index_dof_act(i2);
        GF2(i2) = GF2(i2) - GK(i,j)*GU(j);
    end
end
end

%% The system { GK2 * GU_act = GF2 } is ready to be solved using a known method
GU_act = inv(GK2)*GF2;
GU2_def = GU;
%%--> GU2_def contains "deviations" from "rest positions" of all respective dofs
for i2=1:ndof_act,
    i = index_dof_act(i2);
    GU2_def(i) = GU_act(i2);
end
GU2 = zeros(ndof,1);
%%--> GU2 contains new "deviated positions" of all respective dofs
for i=1:noi,
    for j=1:noj
        GU2(2*((i-1)*noj+j)-1) = (i-1)*L;
        GU2(2*((i-1)*noj+j)) = (j-1)*L;
    end
end
GU2 = GU2 + GU2_def;

%% Definition of Node Coordinates Matrix (NC)
for i = 1:noi*noj
    NC(i,1) = GU2(2*i-1);
    NC(i,2) = scale_y*GU2(2*i);
end

%% Calculation of error factor
error = 0;
n = 1;
for m = 1:sddx %for all the deformation data
    while (Lx/2-ddx(sddx+1-m) > NC(n+noj,1)),
        n = n + noj;
    end
    yerr = NC(n,2)-ddy(sddx+1-m)+(Lx/2-ddx(sddx+1-m)-NC(n,1))*(NC(n+noj,2)-
        NC(n,2))/(NC(n+noj,1)-NC(n,1));

    yerrtotal(m) = yerr;
    error = error + yerr * yerr;
end
error = sqrt(error)/sddx;

%% Model representation
figure
% Definition of Node Adjacency Matrix (NA)
NA(:,:)=zeros(noi*noj,noi*noj);

```

```

for i = 1:noi
    for j = 1:noj
        if i~=noi
            NA((i-1)*noj+j,i*noj+j)=1;
        end
        if j~=noj
            NA((i-1)*noj+j,(i-1)*noj+j+1)=1;
            if i~=1
                NA((i-1)*noj+j,(i-2)*noj+j+1)=1;
            end
        end
    end
end
end

% Plotting
gplot(NA,NC,'-o');
axis ij
axis([0,12,-1, 6]); %([0,12,-1, 11]); %[-1,6,-6,1]);
hold on
plot((Lx/2-ddx1),ddy1,'r+');
ylabel('skin deformation (mm)')
xlabel('distance from the contact point (mm)')
plot((Lx/2-ddx2),ddy2,'rx');
end

```

2 Κώδικας C (εφαρμογή σε OpenGL)

Βασικό αρχείο

```
#include <windows.h>
#include <math.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>
#include "fem1.h"
fem1 *fem_model;          //Finite Element Deformation Model
float def;                //Deformation value (magnitude)
int mode=1;
// Mouse and trackball variables
static int  mousebutton_down = 0;
static int  mouse_posx, mouse_posy;
static int  d_mouse_posx=0, d_mouse_posy=0;
//===== set Viewport (screen window) =====
void setViewport(int left, int right, int bottom, int top)
{
    glViewport(left, bottom, right - left, top - bottom);
}
void myInit(GLint mywidth, GLint myheight)
{
    glClearColor(1.0,1.0,1.0,0.0);
    //set (white) background color
    glColor3f(0.0f, 0.0f, 0.0f);
    //set the drawing color

    //set 3D World-Coordinates Window:
    //glOrtho(left,right, bottom,top, near,far)
    GLfloat aspect = (GLfloat)mywidth/(GLfloat)myheight;
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-8.0*aspect, 8.0*aspect, -8.0, 8.0, -50, 100.0);
    //-----
    setViewport(0, mywidth, 0, myheight);
    //set viewport window (screen coordinates)
    fem_model = new fem1();
}
void myDisplay(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    //clear screen
    //set properties of surface material
    GLfloat mat_ambient[] = {0.8f, 0.3f, 0.3f, 1.0f}; //gray
    GLfloat mat_diffuse[] = {0.9f, 0.4f, 0.4f, 1.0f};
    GLfloat mat_specular[] = {1.0f, 1.0f, 1.0f, 1.0f};
    GLfloat mat_shininess[] = {50.0f};
    glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    //set light sources
    GLfloat lightIntensity[] = {0.7f, 0.7f, 0.7f, 1.0f};
    GLfloat lightIntensity_amb[] = {0.2f, 0.2f, 0.2f, 1.0f};
    GLfloat lightPosition0[] = {5.0f, 2.0f, 0.0f, 0.0f};
    glLightfv(GL_LIGHT0, GL_POSITION, lightPosition0);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, lightIntensity);
}
```

```

glLightfv(GL_LIGHT0, GL_AMBIENT, lightIntensity_amb);
GLfloat lightPosition1[] = {-10.0f, -5.0f, 0.0f, 0.0f};
glLightfv(GL_LIGHT1, GL_POSITION, lightPosition1);
glLightfv(GL_LIGHT1, GL_DIFFUSE, lightIntensity);
glLightfv(GL_LIGHT1, GL_AMBIENT, lightIntensity_amb);
//set the Camera model
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(0.5, 0.2, 0.5, 0.0,0.0,0.0, 0.0, 2.0, 0.0);
//(eye.x,y,z, look.x,y,z, up.x,y,z);
fem_model->deform(def);
glColor3f(0.6, 0.6, 0.6);
glPushMatrix();
glTranslatef(5.0, 1.0, 2.0);
glRotatef(180.0, 0.0, 0.0, 1.0);
glRotatef(-10.0, 0.0, 1.0, 0.0);
int i,j,c1;
float ddata[(noi+3)/2][3];
float dist,defo;
glColor3f(1.0, 0.6, 0.6);
int j1 = 1;
int j2 = 7;
if (mode!=0) {
    if (def>=0) {
        glBegin(GL_TRIANGLE_STRIP); //GL_TRIANGLE_STRIP
        //GL_QUAD_STRIP
        for (i=noi/2+1; i<=noi; i++){
            glNormal3f(0.0, 0.0, 1.0);
            glVertex3f(12-fem_model->NC[j1+(i-1)*noj][1],
                fem_model->NC[j1+(i-1)*noj][2], -6.0);
            glVertex3f(12-fem_model->NC[j2+(i-1)*noj][1],
                fem_model->NC[j2+(i-1)*noj][2], -6.0);
        }
        glEnd();
        glBegin(GL_TRIANGLE_STRIP); //GL_TRIANGLE_STRIP
        //GL_QUAD_STRIP
        for (i=noi/2+1; i<=noi; i++){
            glNormal3f(1.0, 0.0, 0.0);
            glVertex3f(6.0, fem_model->NC[j1+(i-1)*noj][2],-12+fem_model-
                >NC[j1+(i-1)*noj][1]);
            glVertex3f(6.0, fem_model->NC[j2+(i-1)*noj][2],-12+fem_model-
                >NC[j2+(i-1)*noj][1]);
        }
        glEnd();
    }
    else {
        glBegin(GL_TRIANGLE_STRIP); //GL_TRIANGLE_STRIP
        //GL_QUAD_STRIP
        for (i=1; i<=noi/2+1; i++){
            glNormal3f(0.0, 0.0, 1.0);
            glVertex3f(fem_model->NC[j1+(i-1)*noj][1],
                fem_model->NC[j1+(i-1)*noj][2], -6.0);
            glVertex3f(fem_model->NC[j2+(i-1)*noj][1],
                fem_model->NC[j2+(i-1)*noj][2], -6.0);
        }
        glEnd();
        glBegin(GL_TRIANGLE_STRIP); //GL_TRIANGLE_STRIP
        //GL_QUAD_STRIP
    }
}

```

```

        for (i=noi/2+1; i<=noi; i++){
            glNormal3f(1.0, 0.0, 0.0);
            glVertex3f(6.0, fem_model->NC[j1+(i-
                1)*noj][2], -12+fem_model-
                >NC[j1+(i-1)*noj][1]);
            glVertex3f(6.0, fem_model->NC[j2+(i-
                1)*noj][2], -12+fem_model-
                >NC[j2+(i-1)*noj][1]);
        }
        glEnd();
    }
}
glBegin(GL_QUADS); //right side
glNormal3f(-1.0, 0.0, 0.0);
glVertex3f(fem_model->NC[j1][1], fem_model->NC[j1][2],
            0-6*mode*mode);
glVertex3f(fem_model->NC[j2][1], fem_model->NC[j2][2],
            0-6*mode*mode);
glVertex3f(fem_model->NC[j2][1], fem_model->NC[j2][2],
            -12.0);
glVertex3f(fem_model->NC[j1][1], fem_model->NC[j1][2],
            -12.0);
glEnd();
glBegin(GL_QUADS); //front side
glNormal3f(0.0, 0.0, 1.0);
glVertex3f(0+6*mode*mode, fem_model->NC[j1][2], fem_model-
>NC[j1][1]);
glVertex3f(0+6*mode*mode, fem_model->NC[j2][2], fem_model-
>NC[j2][1]);
glVertex3f(12.0, fem_model->NC[j2][2], fem_model->NC[j2][1]);
glVertex3f(12.0, fem_model->NC[j1][2], fem_model->NC[j1][1]);
glEnd();
glBegin(GL_QUADS); //left side
glNormal3f(1.0, 0.0, 0.0);
glVertex3f(12+fem_model->NC[j1][1], fem_model->
            NC[j1][2], 0);
glVertex3f(12+fem_model->NC[j2][1], fem_model->
            NC[j2][2], 0);
glVertex3f(12+fem_model->NC[j2][1], fem_model->
            NC[j2][2], -12.0);
glVertex3f(12+fem_model->NC[j1][1], fem_model->
            NC[j1][2], -12.0);
glEnd();
glBegin(GL_QUADS); //rear side
glNormal3f(0.0, 0.0, -1.0);
glVertex3f(0, fem_model->NC[j1][2], fem_model->
            NC[j1][1]-12);
glVertex3f(0, fem_model->NC[j2][2], fem_model->
            NC[j2][1]-12);
glVertex3f(12.0, fem_model->NC[j2][2], fem_model->
            NC[j2][1]-12);
glVertex3f(12.0, fem_model->NC[j1][2], fem_model->
            NC[j1][1]-12);
glEnd();
ddata[1][1]=0.0; // deformation data
ddata[1][2]=fem_model->NC[noj*(noi-1)/2+1][2];
for (i=2; i<=(noi+1)/2; i++) {
    ddata[i][1]=6.0-fem_model->NC[noj*((noi+1)/2-i+1)][1];
    ddata[i][2]=fem_model->NC[noj*((noi+1)/2-i+1)][2];
}

```



```

ddata[(noi+1)/2+1][1]=15.0; ddata[(noi+1)/2+1][2]=0.0;
GLfloat drho; //, dtheta;
GLfloat dyy, dxx;
for (j=0; j<=11; j++){
    glBegin(GL_TRIANGLE_STRIP); /**top deformable surface**
    //GL_TRIANGLE_STRIP//GL_QUAD_STRIP
        for (i=0; i<=12; i++){
            if ((mode!=0)&&(j<=5)&&(i==0)) i=6;
            dist = sqrt((j+0-6)*(j+0-6) + (i-6)*(i-6));
            c1 = 1;
            while (ddata[c1][1]<dist) c1++;
            defo = ddata[c1-1][2] + (ddata[c1][2]-
                ddata[c1-1][2])*(dist-ddata[c1-1][1])/
                (ddata[c1][1]-ddata[c1-1][1]);
            if (c1>=16) defo=0.0;
            if ((dist>=12) || (dist==0)) glNormal3f(0.0, 1.0, 0.0);
            else {
                dyy = (ddata[c1][2] - ddata[c1-1][2]);
                dxx = (ddata[c1][1] - ddata[c1-1][1]);
                //dtheta = atan(dyy/dxx);
                drho = sqrt(dyy*dyy + dxx*dxx);
                glNormal3f(dxx*i/drho/dist, dyy/drho,
                    dxx*(j+0)/drho/dist);
            }
            glVertex3f(i, defo, -j);
            dist = sqrt((j+1-6)*(j+1-6) + (i-6)*(i-6));
            c1 = 1;
            while (ddata[c1][1]<dist) c1++;
            defo = ddata[c1-1][2] + (ddata[c1][2]-
                ddata[c1-1][2])*(dist-ddata[c1-1][1])/
                (ddata[c1][1]-ddata[c1-1][1]);
            if (c1>=16) defo=0;
            if ((dist>=12) || (dist==0)) glNormal3f(0.0, 1.0, 0.0);
            else {
                dyy = (ddata[c1][2] - ddata[c1-1][2]);
                dxx = (ddata[c1][1] - ddata[c1-1][1]);
                //dtheta = atan(dyy/dxx);
                drho = sqrt(dyy*dyy + dxx*dxx);
                glNormal3f(dxx*i/drho/dist, dyy/drho,
                    dxx*(j+1)/drho/dist);
            }
            glVertex3f(i, defo, -j-1);
        }
    glEnd();
}
glPointSize(4.0);
glColor3f(1.0, 0.0, 0.0);
if (mode==-1) {
    glBegin(GL_POINTS);
        for (j=1; j<=(noj-1); j++)
            for (i=1; i<noi/2+2; i++){
                glVertex3f(fem_model->NC[j+(i-1)*noj][1], fem_model-
>NC[j+(i-1)*noj][2], -5.95);
            }
    glEnd();
    glBegin(GL_POINTS);
        for (j=1; j<=(noj-1); j++)
            for (i=1; i<noi/2+1; i++){
                glVertex3f(5.95, fem_model->NC[j+(i-1)*noj][2], -
fem_model->NC[j+(i-1)*noj][1]);
            }
}

```

```

        glEnd();
    }
    }
    glPopMatrix();
    glFlush(); //send all output to display (for GLUT_SINGLE)
    glutSwapBuffers(); //for double buffering (GLUT_DOUBLE)
}
void Update()
{
    glutPostRedisplay();
}
void myMouse(int button, int state, int x, int y)
{
    if (state == GLUT_DOWN)
    {
        switch (button)
        {
            case GLUT_LEFT_BUTTON:
                mousebutton_down = 1;
                if (glutGetModifiers() & GLUT_ACTIVE_SHIFT) {
                    def -= 0.1f;
                    if (def<0.0f) def=0.0f;
                }
                else {
                    def += 0.1f;
                    if (def>2.0f) def=2.0f;
                }
                break;
            case GLUT_RIGHT_BUTTON:
                mousebutton_down = 2;
                def -= 0.1f;
                if (def<0.0f) def=0.0f;
                break;
            case GLUT_MIDDLE_BUTTON:
                mousebutton_down = 3;
                if (mode==0) mode=1;
                else if (mode==1) mode=-1;
                else mode=0;
                break;
        }
        printf("%f\n", def);
        mouse_posx = x;
        mouse_posy = y;
    }
    else
    {
        mousebutton_down = 0;
        d_mouse_posx = d_mouse_posy = 0;
    }
}
void myMovedMouse(int mouseX, int mouseY)
{
    d_mouse_posx = mouse_posx - mouseX;
    d_mouse_posy = mouse_posy - mouseY;
    mouse_posx = mouseX;
    mouse_posy = mouseY;
    switch (mousebutton_down)
    {
        case 1: def -= 0.005*d_mouse_posy;
                if (def>2.0f) def=2.0f;
    }
}

```

```

        if (def<=-2.0f) def=-2.0f;
    break;
case 3: def -= 0.005*d_mouse_posy;
        if (def>2.0f) def=2.0f;
        if (def<=-2.0f) def=-2.0f;
    break;
}
}
void myKeyboard(unsigned char theKey, int mouseX, int mouseY)
{
}
void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE| GLUT_RGB | GLUT_DEPTH);
    //GLUT_SINGLE
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100,150);
    glutCreateWindow("Simple 3D Shaded Scene");
    glutDisplayFunc(myDisplay);
    glutIdleFunc(Update); // User provided Update function
    glutMouseFunc(myMouse); // mouse pressed/released
    glutMotionFunc(myMovedMouse); // mouse moved
    glutKeyboardFunc(myKeyboard); // key pressed
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_LIGHT1);
    glShadeModel(GL_SMOOTH); //smooth (Gouraud) shading
    glEnable(GL_DEPTH_TEST); //hidden surface removal
    glEnable(GL_NORMALIZE); //normalize vectors for proper shading
    myInit(640, 480);
    glutMainLoop();
}

```

Αρχείο fem1.cpp

```
/* ===== */
/* === 2D Finite Element Deformation Model === */
/* === Inputs Deformation Magnitude === */
/* === Outputs Node Coordinates NC[0:120][0:2] === */
/* === i.e. 120 points (x,y), of a 2D FEM === */
/* ===== */
#include "fem1.h"
fem1::fem1()
{
    //DEBUG FLAGS
    debug_flag4 = 0; //1
    nload = (noi+1)/2;
    nlayers = noj-1;
    Lx = 12.0;
    L = Lx/(noi-1);
    scale_y = 1.0;
    A = L*L/2;
    E = 0.2f;
    v = 0.48f;
    for (i=1;i<=noi*noj*2;i++) {
        GF[i] = 0.0;
        GF2[i] = 0.0;
        GU[i] = -1.0;
        for (j=1;j<=noi*noj*2;j++)
            GK[i][j]=0.0;
            GK2[i][j]=0.0;
    }
    for (i=1;i<=6;i++)
        Cr[i]=0;

    for (i=1;i<=3;i++)
        for (j=1;j<=6;j++) {
            B1[i][j]=0;
            BT1[j][i]=0;
        }

    B1[1][1]=B1[2][2]=B1[3][1]=B1[3][2]=1/L;
    B1[1][5]=B1[2][4]=B1[3][3]=B1[3][6]=-1/L;
    BT1[1][1]=BT1[1][3]=BT1[2][2]=BT1[2][3]=1/L;
    BT1[3][3]=BT1[4][2]=BT1[5][1]=BT1[6][3]=-1/L;

    for (i=1;i<=3;i++)
        for (j=1;j<=6;j++) {
            B2[i][j]=-1.0f*B1[i][j];
            BT2[j][i]=-1.0f*BT1[j][i];
        }

    D[1][1]=D[2][2]=1.0;
    D[2][1]=D[1][2]=v;
    D[3][3]=(1-v)/2;
    D[1][3]=D[2][3]=D[3][1]=D[3][2]=0;
    for (i=1;i<=3;i++)
        for (j=1;j<=3;j++)
            D[i][j]=D[i][j]*(E/(1+v*v));
}
```



```

Cr[3] = 2 * ((ib-1) * noj + jb + 1) - 1;
Cr[4] = 2 * ((ib-1) * noj + jb + 1);
Cr[5] = 2 * (ib * noj + jb) - 1;
Cr[6] = 2 * (ib * noj + jb);
for (k=1;k<=6;k++) {
    for (l=1;l<=6;l++) {
        GK[Cr[k]][Cr[l]] = GK[Cr[k]][Cr[l]] +
            K1[k][l];
    }
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Second element in the square [ (i,j), (i,j+1), (i+1,j+1), (i+1,j) ]
%% For this element node 1 is at the coordinates (i+1,j+1) node 2 at
%% (i+1,j) and node 3 at (i,j+1)
    if (jb<=j_layer1) {
        for (i=1;i<=3;i++) {
            for (j=1;j<=3;j++) {
                D2[i][j]=D[i][j];
            }
        }
    } else {
        for (i=1;i<=3;i++) {
            for (j=1;j<=3;j++) {
                D2[i][j]=Delf[i][j];
            }
        }
    }
// Stiffness matrix K2 of the element
i=6; j=3; k=3;
for (i1=1;i1<=i;i1++) {
    for (k1=1;k1<=k;k1++) {
        BT2D2[i1][k1] = 0;
        for (j1=1;j1<=j;j1++) {
            BT2D2[i1][k1] = BT2D2[i1][k1] +
                BT2[i1][j1] * D2[j1][k1];
        }
    }
}
i=6; j=3; k=6;
for (i1=1;i1<=i;i1++) {
    for (k1=1;k1<=k;k1++) {
        K2[i1][k1] = 0;
        for (j1=1;j1<=j;j1++) {
            K2[i1][k1] = K2[i1][k1] +
                BT2D2[i1][j1] * B2[j1][k1];
        }
    }
}
Cr[1] = 2 * (ib * noj + jb + 1) - 1;
Cr[2] = 2 * (ib * noj + jb + 1);
Cr[3] = 2 * (ib * noj + jb) - 1;
Cr[4] = 2 * (ib * noj + jb);
Cr[5] = 2 * ((ib-1) * noj + jb + 1) - 1;
Cr[6] = 2 * ((ib-1) * noj + jb + 1);
for (k=1;k<=6;k++) {
    for (l=1;l<=6;l++) {
        GK[Cr[k]][Cr[l]] = GK[Cr[k]][Cr[l]] +
            K2[k][l];
    }
}

```

```

    }
}
}
//% At this point matrixes GF and GK have been created
//% end of constructor fem1(...)

void fem1::deform(float in_dist)
{
// Boundary conditions
// Node (nload,1) with coordinates ((nload-1)*L,0) is pushed down at distance dist
    i = nload;
    j = 1;
    GU[2*((i-1)*noj+j)] = (j-1)*L+in_dist;
// Every node with coordinates x=0 cannot be moved
    i = 1;
    for (j=1;j<=noj;j++) {
        GU[2*((i-1)*noj+j)-1] = 0;
        GU[2*((i-1)*noj+j)] = 0;
    }
//Every node with coordinates y=(noj-1)*L cannot be moved
//These are all the nodes to the bottom.
    j = noj;
    for (i=1;i<=noi;i++) {
        GU[2*((i-1)*noj+j)-1] = 0;
        GU[2*((i-1)*noj+j)] = 0;
    }
//Every node with coordinates x=(noi-1)*L cannot be moved
    i=noi;
    for (j=1;j<=noj;j++) {
        GU[2*((i-1)*noj+j)-1] = 0;
        GU[2*((i-1)*noj+j)] = 0;
    }
    ndof = 2 * noi * noj;
// ndof: total number of degrees of freedom in the FEM

    i2 = 0;
    for (i=1;i<=ndof;i++) {
        if (GU[i] == -1) {
            i2++;
            index_dof_act[i2] = (float)i;
            index_dof_all[i] = (float)i2;
        } else {
            index_dof_all[i] = -1;
        }
    }
    ndof_act = i2;

    for (j=1;j<noi*noj*2+1;j++) {
        GF2[j] = 0.0;
        for (j2=1;j2<noi*noj*2+1;j2++) {
            GK2[j][j2] = 0.0;
        }
    }

    for (j=1;j<=ndof;j++) {
        if (index_dof_all[j] != -1.0) {
            j2 = (int)index_dof_all[j];

```

```

        for (i2=1;i2<=ndof_act;i2++) {
            i = (int)index_dof_act[i2];
            GK2[i2][j2] = GK[i][j];
        }
    } else {
        for (i2=1;i2<=ndof_act;i2++) {
            i = (int)index_dof_act[i2];
            GF2[i2] = GF2[i2] - GK[i][j]*GU[j];
        }
    }
}

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//% The system { GK2 * GU_act = GF2 } is ready to be solved using a known method
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dim = 0;
for (i=1;i<noi*noj*2+1;i++) {
    for (j=1;j<noi*noj*2+1;j++) {
        if (GK2[i][j]!=0.0) {
            if (dim<i) dim = i;
            if (dim<j) dim = j;
        }
    }
}

for (gpas=0; gpas<dim; gpas++) {
    ok = 1;
    if ( (GK2[gpas+1][gpas+1] < EPS) && (GK2[gpas+1][gpas+1] > MEPS) ) {
        ok = 0;
        i = gpas + 1;
        while ( (i<dim) && (GK2[i+1][gpas+1] < EPS) &&
            (GK2[i+1][gpas+1] > MEPS) ) {
            i++;
        }
        if (i<dim) {
            ok = 1;
            for (j=gpas; j<dim; j++) {
                atmp = GK2[gpas][j+1];
                GK2[gpas+1][j+1] = GK2[i+1][j+1];
                GK2[i+1][j+1] = atmp;
            }
            btmp = GF2[gpas+1];
            GF2[gpas+1] = GF2[i+1];
            GF2[i+1] = btmp;
        }
    }
    if (ok) {
        for (i=(gpas+1); i<dim; i++) {
            if ( (GK2[i+1][gpas+1] > EPS) || (GK2[i+1][gpas+1] < MEPS) ) {
                mk[i+1][gpas+1] = GK2[i+1][gpas+1]/GK2[gpas+1][gpas+1];
                GF2[i+1] = GF2[i+1] - mk[i+1][gpas+1]*GF2[gpas+1];
                for (j=gpas+1; j<dim; j++)
                    GK2[i+1][j+1] = GK2[i+1][j+1] -
                        mk[i+1][gpas+1]*GK2[gpas+1][j+1];
            }
        }
    }
}
ok = 1;

```



```

if ( (GK2[dim][dim] < EPS) && (GK2[dim][dim] > MEPS) )
    ok = 0;
else
    GU_act[dim] = GF2[dim]/GK2[dim][dim];
i = dim-2;
while ( (i>=0) && (ok) ) {
    if ( (GK2[i+1][i+1] < EPS) && (GK2[i+1][i+1] > MEPS) ) ok=0;
    else {
        GU_act[i+1] = GF2[i+1];
        for ( j=dim-1; j>=(i+1); j-- ) {
            GU_act[i+1] = GU_act[i+1] - GK2[i+1][j+1]*GU_act[j+1];
        }
        GU_act[i+1] = GU_act[i+1]/GK2[i+1][i+1];
    }
    i = i - 1;
}
//%%
//% GU_act has now been calculated
//%
for (i=1;i<=noi*noj*2+1;i++) {
    GU2_def[i] = GU[i];
//%--> GU2_def contains "deviations" from "rest positions" of all respective dofs
}
for (i2=1;i2<=ndof_act;i2++) {
    i = (int)index_dof_act[i2];
    GU2_def[i] = GU_act[i2];
}

for (i=1;i<=noi*noj*2+1;i++) {
    GU2[i] = 0;
//%--> GU2 contains new "deviated positions" of all respective dofs
}
for (i=1;i<=noi;i++) {
    for (j=1;j<=noj;j++) {
        GU2[2*((i-1)*noj+j)-1] = (i-1)*L;
        GU2[2*((i-1)*noj+j)] = (j-1)*L;
    }
}
for (i=1;i<=noi*noj*2+1;i++) {
    GU2[i] = GU2[i] + GU2_def[i];
}

//%
//% Definition of Node Coordinates Matrix (NC)
//%
for (i = 1;i<=noi*noj;i++) {
    NC[i][1] = GU2[2*i-1];
    NC[i][2] = scale_y*GU2[2*i];
}
//%
//% Calculation of error factor
//%
if (debug_flag4) {
    for (i = 1;i<=noi*noj;i++) {
        printf ("NC[%i][1] %f\t NC[%i][2] %f\n", i,
            NC[i][1], i, NC[i][2]);
    }
    debug_flag4 = 0;
}
} // end of deform(...)

```

Αρχείο fem1.h

```
/**
// *****
//   Filename : fem1.h
// -----
// Header File for a Finite Element Deformation Class

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define EPS 0.0001
#define MEPS (-0.0001)

// The following 2 values can be changed
#define noi 17
#define noj 7

// The following 2 values can be changed
static int j_layer1 = 3;
static int el_f = 10;

class fem1
{
public:

    // Constructor.
    fem1(); // flag to activate force feedback

    void deform(float in_dist);
    float NC[120][3];

private:

//const int distcorr = 0;
    int debug_flag4;
    float ddx[19];
    float ddy[19];
    float ddx1[19];
    float ddef1[19];
    float ddx2[19];
    float ddef2[19];
    float ddy1[19];
    float ddy2[19];
    float yerrtotal[19];
    float GK[noi*noj*2+1][noi*noj*2+1];
    float GK2[noi*noj*2+1][noi*noj*2+1];
    float GF[noi*noj*2+1];
    float GF2[noi*noj*2+1];
    float GK_tmp1[noi*noj*2+1][noi*noj*2+1];
    float GF_tmp1[noi*noj*2+1];
    float index_dof_act[noi*noj*2+1];
    float index_dof_all[noi*noj*2+1];
    float GU[noi*noj*2+1];
    float GU2[noi*noj*2+1];
    float mk[noi*noj*2+1][noi*noj*2+1];
    float GU_act[noi*noj*2+1];
    float GU2_def[noi*noj*2+1];
    int Cr[7];
    float B1[4][7];

```

```
float BT1[7][4];
float B2[4][7];
float BT2[7][4];
float D[4][4];
float Delf[4][4];
float D1[4][4];
float D2[4][4];
float BT1D1[7][4];
float K1[7][7];
float BT2D2[7][4];
float K2[7][7];

int nload, nlayers, i, j, ib, jb, k, l, m, n, set, sddx, sddx1, sddx2, dim;
int i1, i2, k1, j1, j2, /*disterrloop,*/ ndof, ndof_act, gpas, ok, chk, chk2;
float dscale, Lx, L, dist, scale_y, A, E, v, atmp, btmp, error, yerr;
```

```
};
```


Βιβλιογραφία

- [1] Sarah F. F. Gibson, Brian Mirtich (1997), "A Survey of Deformable Modeling in Computer Graphics", Technical Report No: TR97-19, *Mitsubishi Electric Research Laboratories (MERL)*, Cambridge, Massachusetts, November 1997. <http://www.merl.com/reports/index.html>
- [2] Simon P. DiMaio, S. E. Salcudean (2003), "Needle Insertion Modeling and Simulation", *IEEE Transactions on Robotics and Automation*, Special Issue on Medical Robotics, vol. 19, no. 5, pp. 864-875, October 2003.
- [3] Stéphane Cotin, Hervé Delingette, Nicholas Ayache (1999), "Real-time elastic deformations of soft tissues for surgery simulation", *IEEE Transactions on Visualization and Computer Graphics*, 5(1): 62-73, March 1999.
- [4] Morten Bro-Nielsen (1998), "Finite Element Modeling in Surgery Simulation", *Proceedings of the IEEE*, vol. 86, no. 3, pp. 490-503, March 1998.
- [5] Ανδρέας Δ. Κατσάμπας, *Μαθήματα Δερματολογίας & Αφροδισιολογίας*, Κλινική Αφροδισίων & Δερματικών Νόσων, Πανεπιστήμιο Αθηνών, Νοσοκομείο «Α. Συγγρός», Ιατρικές Εκδόσεις Π. Χ. Πασχαλίδης.
- [6] Rod Lakes, University of Wisconsin, Technical Report, <http://silver.neep.wisc.edu/~lakes/PoissonIntro.html>
- [7] Simon P. DiMaio (2003): *Modelling, Simulation and Planning of Needle Motion in Soft Tissues*, Ph.D. Thesis, The University of British Columbia, September 2003
- [8] M J Fagan (1996), *Finite Element Analysis: Theory and Practice*, Addison-Wesley, February 1996

- [9] M. A. Srinivasan (1989), "Surface deflection of primate fingertip under line load", *J Biomechanics Vol 22*, No 4, pp 343-349, 1989.
- [10] C.S. Tzafestas, Y. Koumpouros, K. Birbas (2004), "Paracentesis modeling and VR-based interactive simulation with haptic display for clinical skill training and assessment," in *Proceedings of the International Conference on Integrated Modeling & Analysis in Applied Control & Automation (IMAACA2004)*, I3M: *International Mediterranean Modeling Multiconference*, Genoa, Italy, October 28-30 2004. http://www.liophant.org/i_m_cs/i3m2004/imaaca/
- [11] C.S. Tzafestas, Y. Koumpouros, K. Birbas, "Haptic interaction in VR-based paracentesis simulation for dexterity enhancement and assessment", in *Proceedings of the 18th International Congress and Exhibition on Computer Assisted Radiology and Surgery – CARS 2004*, June 23-26, 2004 Chicago, USA.