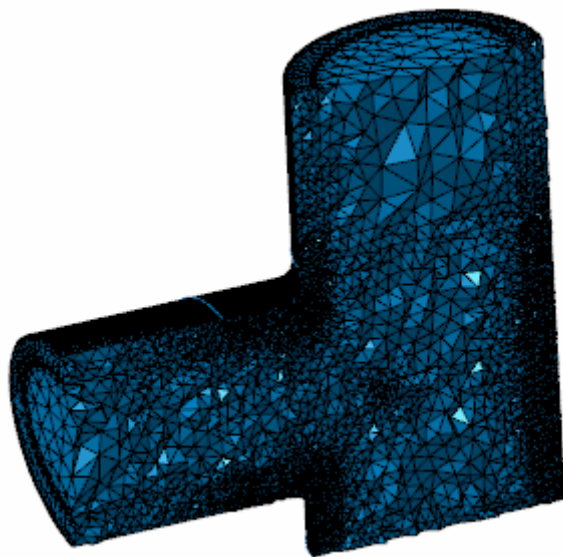




ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΘΕΡΜΙΚΩΝ ΣΤΡΟΒΙΛΟΜΗΧΑΝΩΝ

Διπλωματική Εργασία:

Το Λογισμικό CATIA ως Εργαλείο
Παραμετροποίησης και Γένεσης Πλέγματος σε
Προβλήματα Βελτιστοποίησης Μορφής



Τζώνης Ιωάννης
Κατασκευαστής Μηχανολόγος Μηχανικός

Επιβλέπον: Επ.Καθηγητής Κ.Χ Γιαννάκογλου

Αθήνα, Μάρτιος 2009

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. *Γιαννάκογλου Κ. Χ.* για την εμπιστοσύνη που μου έδειξε όσον αφορά την εκπόνηση της διπλωματικής αυτής εργασίας, για την καθοδήγησή του αλλά και για το ίδιο το αντικείμενο της εργασίας που μου ανάθεσε, διότι αποδείχτηκε εξαιρετικά ενδιαφέρον και διότι συμπίπτει με τα ενδιαφέροντά μου, όσον αφορά τις σπουδές μου.

Επίσης, θα ήθελα να ευχαριστήσω όλους όσους απασχολούνται στο Εργαστήριο Θερμικών Στροβιλομηχανών και κυρίως την Δρ. Ασούτη Β. για την βοήθεια που μου προσέφεραν επί του αντικειμένου αλλά και την υπομονή τους.

Οφείλω να πω ένα μεγάλο ευχαριστώ στους αδελφικούς μου φίλους Αβραμίδη Αβραάμ και Σαπινά Ιωάννη για την υποστήριξη τους και τις αστείρευτες γνώσεις τους στο αντικείμενο των ηλεκτρονικών υπολογιστών, τμήμα των οποίων μου μετέδωσαν, καθώς και τον επίσης αδελφικό μου φίλο και συνάδελφο Ζάχο Δήμο και την κοπέλα μου Βάθη Παναγιώτα για την συμπαράσταση και εμπύχωσή μου.

Τέλος, το μεγαλύτερο ευχαριστώ στην οικογένειά μου και σε όλους τους φίλους που με υποστήριξαν και στάθηκαν δίπλα μου παρά την πίεση και το άγχος που πολλές φορές επηρέαζε την συμπεριφορά μου απέναντί τους.

Ολοκληρώνοντας, αφιερώνω την παρούσα διπλωματική εργασία στον παππού μου Γλυνιά Ιωάννη, μία από τις επιθυμίες του οποίου ήταν η αποφοίτησή μου από την σχολή των Μηχανολόγων Μηχανικών του ΕΜΠ.

Πρόλογος

Όσα θα ακολουθήσουν στο παρόν σύγγραμμα εκπονήθηκαν και κατεγράφησαν στα πλαίσια διπλωματικής εργασίας υπό την επίβλεψη του κ. Γιαννάκογλου Κ.Χ. Αφορούν το λογισμικό CATIA, το οποίο περιγράφεται εντός του συγγράμματος, και την ένταξή του στον τομέα της βελτιστοποίησης. Επίσης, αφορούν την δημιουργία κατάλληλου κώδικα για την αυτοματοποίηση της διαδικασίας, μέσω της γλώσσας προγραμματισμού Microsoft Visual Basic.

Και τα δύο αυτά λογισμικά είναι ευρέως διαδεδομένα στο εμπόριο και συνδυάζουν πληθώρα δυνατοτήτων, χρήσιμων και μη για το παρόν αντικείμενο. Η εκμάθησή τους είναι ιδιαίτερα απαιτητική αλλά παρ' όλ' αυτά η εξάπλωσή τους σε διάφορους τομείς όπως ο βιομηχανικός κλάδος είναι ραγδαία.

Αν και σχετικά νέα λογισμικά συγκριτικά με τους κλάδους εφαρμογής τους, αναγνωρίστηκαν γρήγορα και συνεχώς εξελίσσονται. Οι δυνατότητές τους είναι ανεξάντλητες. Έτσι, η εργασία αυτή αποσκοπεί κατά κάποιον τρόπο στην διεύρυνση του αντικειμένου εφαρμογής τους και εκμετάλλευσης των δυνατοτήτων τους, μέσω των μεθόδων βελτιστοποίησης, σε διάφορα τμήματα του βιομηχανικού κλάδου όπως η έρευνα και ανάπτυξη αλλά και η ίδια η παραγωγή προϊόντων.

Σημειώνεται τέλος, ότι με τον όρο βελτιστοποίηση, όσον αφορά την διπλωματική αυτή εργασία, εννοείται η βελτιστοποίηση μορφής σχήματος πρωτογενών τεμαχίων, δηλαδή η εύρεση της βέλτιστης γεωμετρίας ενός αντικειμένου βάσει κριτηρίων. Επίσης, η εφαρμογή των όσων υλοποιούνται στη συνέχεια έγινε στο Εργαστήριο Θερμικών Στροβιλομηχανών του ΕΜΠ και επικεντρώνεται σε γεωμετρικές μορφές που αφορούν ροές ρευστών και στην επίλυση αυτών μέσω μεθόδων Υπολογιστικής Ρευστομηχανικής.

Περιεχόμενα

Πίνακας Σχημάτων	3
Πίνακας Πινάκων	5
1. Εισαγωγή.....	6
1.1 Σκοπός της Παρούσας Εργασίας	7
1.2 Δομή της Παρούσας Εργασίας	8
2. Εξελικτικοί Αλγόριθμοι Βελτιστοποίησης	9
2.1 Σύντομη Ιστορική Αναδρομή Μεθόδων Βελτιστοποίησης.....	9
2.2 Βελτιστοποίηση μέσω Εξελικτικών Αλγορίθμων	10
2.2.1 Περιγραφή Εξελικτικού Αλγορίθμου	12
2.2.2 Μειονέκτημα των EA – Κόστος Υπολογιστικού Χρόνου	14
2.3 EASY (Evolutionary Algorithms System).....	15
3. Το Λογισμικό CATIA	16
3.1 Γενικά.....	16
3.2 Τμήματα του CATIA.....	17
3.3 Το περιβάλλον του CATIA	18
3.4 Visual Basic Editor	21
3.5 Γραμμή εντολών DOS.....	23
3.6 Απαιτήσεις.....	23
4. Το CATIA στη Βελτιστοποίηση.....	24
4.1 Ο Ρόλος του CATIA	24
4.2 Υλοποίηση και Αυτοματοποίηση της Διαδικασίας	26
4.2.1 Γεωμετρία	27
4.2.2 Design Table	29
4.2.3 Μη-Δομημένο Πλέγμα	31
4.2.4 Macros	34

4.2.5	Συμπληρωματικοί κώδικες	38
4.2.6	Ολοκλήρωση αυτοματοποίησης	45
5.	CFD – Επιλύτης Navier – Stokes	47
6.	Πρακτική Εφαρμογή	48
6.1	Γενική περιγραφή.....	48
6.2	Κατασκευή Γεωμετρίας.....	49
6.3	Παραμετροποίηση της γεωμετρίας.....	53
6.4	Δημιουργία Πλέγματος.....	59
6.5	Προσαρμογή Τμημάτων Κώδικα “VarUpdate” και “Processing_of_Data” στην Εφαρμογή.....	63
6.6	Δοκιμή Εφαρμογής σε Διαδικασία Βελτιστοποίησης.....	64
6.6.1	Διαδικασία Βελτιστοποίησης με χρήση του EASY	65
6.6.2	Δοκιμές Επίλυσης Ροών / Συνεργασία CATIA – Επιλύτη	68
7.	Συμπεράσματα και Προτάσεις Μελλοντικής Εργασίας.....	73
7.1	Ανακεφαλαίωση / Συμπεράσματα.....	73
7.2	Προτάσεις Περαιτέρω Ανάπτυξης του Αντικειμένου	74
8.	Βιβλιογραφία.....	77
8.1	Έντυπη Βιβλιογραφία	77
8.2	Ψηφιακή Βιβλιογραφία	78
Παράρτημα Ι.....		79
Παράρτημα ΙΙ.....		82
Παράρτημα ΙΙΙ.....		86

Πίνακας Σχημάτων

Αριθμός Σχήματος	Ονομασία Σχήματος	Σελ.
2.1	Οπτική απεικόνιση εξελικτικού αλγορίθμου	13
3.1	Κεντρικό παράθυρο CATIA	19
3.2	Start menu	20
3.3	Τμήμα δέντρου χαρακτηριστικών σχεδιασμού για το προϊόν Part1	21
3.4	Προγραμματιστικό περιβάλλον Visual Basic Editor	22
4.1	Λειτουργία προσθήκης CATIA στη βελτιστοποίηση	25
4.2	Διαδικασία βελτιστοποίησης - θέση CATIA επί του συνόλου	26
4.3	Ενδεικτική μορφή design table αρχείου	27
4.4	Διακλάδωση σωληνώσεων σχεδιασμένη στο CATIA	28
4.5	Παράμετροι (constraints) γεωμετρίας σωλήνωσης όπως αυτοί φαίνονται εντός του περιβάλλοντος του CATIA	29
4.6	Knowledge toolbar	30
4.7	Παράθυρο δημιουργίας Design table	30
4.8	Παράθυρα καθορισμού παραμέτρων πλέγματος	32
4.9	Αλληλοεξάρτηση αρχείων	33
4.10	Περιγραφή δομής ExportedMesh_Temp.dat αρχείου	35
4.11	Τμήμα της μακροεντολή TaskCode της βιβλιοθήκης VBAProject1.catvba	37
4.12	Τμήμα κώδικα VarUpdate	39
4.13	Τμήμα κώδικα Processing_of_Data	44
4.14	Διάγραμμα ροής της διαδικασίας	46
6.1	Σωλήνωση μορφής T	48
6.2	Όγκος ρευστού εντός της σωλήνωσης	49
6.3	Μορφή γεωμετρίας μετά την εκτέλεση των εντολών fillet	52
6.4	Τελική γεωμετρία όγκου ρευστού	53
6.5	Λανθασμένη γεωμετρία λόγω υπέρβασης επιτρεπών ορίων	55
6.6	Ενδεικτική μορφή γεωμετρίας για τιμές μεταβλητών (1) = 110, (2) = 85, (3) = 85, (4) = 60, (5) = 45, (6) = 140 (7) = 7	56
6.7	Δέντρο παραγόμενου πλέγματος και κάθετη τομή αυτού	62
6.8	Γράφημα τιμών όγκου συναρτήσεως του πλήθους των αξιολογήσεων	66
6.9	Γεωμετρικό Μοντέλο (μετά από 60 αξιολογήσεις)	66

Πίνακας Σχημάτων (Συνέχεια)

6.10	Γεωμετρικό Μοντέλο (μετά από 420 αξιολογήσεις)	67
6.11	Γεωμετρικό Μοντέλο (μετά από 840 αξιολογήσεις)	67
6.12	Απεικόνιση βέλτιστης γεωμετρίας	68
6.13	Απεικόνιση αριθμού Mach (1 ^η δοκιμή)	69
6.14	Κάθετη τομή ($y=0$) ροής με απεικόνιση του αριθμού Mach	70
6.15	Βέλη ταχυτήτων ροής (1 ^η δοκιμή)	70
6.16	Απεικόνιση αριθμού Mach (2 ^η δοκιμή)	71
6.17	Βέλη ταχυτήτων ροής (2 ^η δοκιμή)	72
6.18	Απεικόνιση αριθμού Mach (2 ^η δοκιμή) σε κλίμακα όμοια με της 1 ^η δοκιμής	72

Πίνακας Πινάκων

Αριθμός Πίνακα	Ονομασία Πίνακα	Σελ.
3.1	Τμήματα CATIA	18
4.1	Περιγραφή logfr	40
6.1	Ανεξάρτητες μεταβλητές γεωμετρίας	58
6.2	Σύνολο επιφανειακών πλεγμάτων παραγόμενων με το εργαλείο Advanced Surface Mesher	60
6.3	Σύνολο επιφανειακών πλεγμάτων παραγόμενων με το εργαλείο Extrude Mesher with Translation	61

1. Εισαγωγή

Όπως είναι γνωστό, η χρήση μεθόδων βελτιστοποίησης αποτελεί πλέον μια βασικότερη συνιστώσα σε πολλές περιοχές έρευνας σχεδόν σε κάθε επιστήμη. Και αυτό γιατί έχουν αυξηθεί ραγδαία οι απαιτήσεις τόσο των αναγκών που προϋπάρχουν ή προκύπτουν με το πέρασμα του χρόνου, όσο και των εφαρμογών που τις καλύπτουν.

Γενικότερα παρατηρείται μία τάση, σε κάθε ερώτημα που προκύπτει, όχι απλά εύρεσης μίας λύσης αυτού, αλλά εύρεση της καλύτερης δυνατής λύσης με βάση κριτήρια που ποικίλουν ανά περίπτωση, μέσω μεθόδων βελτιστοποίησης.

Πολύ σημαντικό ρόλο στην ανάπτυξη και χρήση των μεθόδων βελτιστοποίησης, λόγω της μεγάλης πολυπλοκότητας των προβλημάτων που πρόκειται να διαχειριστούν, έπαιξαν οι ηλεκτρονικοί υπολογιστές (H/Y), οι οποίοι, με την αυξημένη υπολογιστική ισχύ τους κατέστησαν δυνατή την επίλυση δύσκολων μαθηματικών μοντέλων σε μικρά χρονικά διαστήματα και με ικανοποιητική, αλλά ταυτόχρονα και ελεγχόμενη, ακρίβεια.

Έτσι, έχουμε φτάσει σε ένα σημείο όπου αυξάνεται συνεχώς η εκμετάλλευση των δυνατοτήτων των H/Y, χάρη στη δημιουργία λογισμικών που ενσωματώνουν μαθηματικά μοντέλα, γραφικά περιβάλλοντα (GUI), γλώσσες προγραμματισμού κ.α.

Φυσικά, δεν θα μπορούσε να αποτελέσει εξαίρεση ο κλάδος της βιομηχανίας, του οποίου οι ανάγκες προσαρμόζονται και εξελίσσονται συνεχώς, παράλληλα με την τεχνολογία. Η ενσωμάτωση της βελτιστοποίησης, είτε σε επίπεδο ερευνών είτε σε επίπεδο παραγωγής μίας βιομηχανίας, συντελεί σημαντικά στην ανταγωνιστικότητα αυτής και κατ' επέκταση και στην επίτευξη των στόχων της, αφού της δίνει την δυνατότητα αντιμετώπισης προβλημάτων και προκλήσεων, μειώνοντας σημαντικά το οποιαδήποτε μορφής κόστος προκύπτει.

Η εκμετάλλευση, όμως, των H/Y από το συγκεκριμένο κλάδο δεν περιορίζεται μόνο σε αυτό. Είναι γνωστή η συμβολή διαφόρων software, εξειδικευμένων και μη, στην σύγχρονη παραγωγική διαδικασία, με μέγιστο αντιπρόσωπο αυτών τα προγράμματα CAD και CAD - CAM - CAE τα οποία διευκολύνουν τον σχεδιασμό προϊόντων και την αυτοματοποίηση της παραγωγής. Ένα τέτοιο πρόγραμμα είναι και το CATIA, το οποίο θα μας απασχολήσει και στη συνέχεια. Πρόκειται για ολοκληρωμένο πακέτο λογισμικού, τύπου CAD - CAM -

CAE, διεθνώς διαδεδομένο, με τεράστιες δυνατότητες, οι οποίες έχουν αποδειχθεί έμπρακτα στο πέρασμα του χρόνου, σε πληθώρα εφαρμογών.

Περιορίζοντας τον γενικότερο κλάδο της βιομηχανίας στον τομέα της μηχανικής των ρευστών (ή οποιασδήποτε εφαρμογής σχετίζεται με αυτά) παρατηρείται και ένας άλλος τρόπος εκμετάλλευσης της υπολογιστικής ισχύος, μέσω μεθόδων Υπολογιστικής Ρευστοδυναμικής (CFD - Computational Fluid Dynamics). Λόγω της πολυπλοκότητας και τις δυσκολίας επίλυσης της ροής ενός ρευστού, είναι ολοφάνερη η τεράστια σημασία χρήσης CFD με τη βοήθεια Η\Υ σε ένα μεγάλο εύρος βιομηχανιών όπως αεροπλοΐα, ναυσιπλοΐα, αυτοκινητοβιομηχανίες, βιομηχανίες υδραυλικών εγκαταστάσεων κ.λπ.

Η Υπολογιστική Ρευστοδυναμική χρειάζεται παραμετροποίηση των μορφών που εμπλέκονται στη ροή (αεροτομών, αγωγών, πτερυγίων, πτερύγων, αεροσκαφών, οχημάτων, κλπ), γένεση υπολογιστικών πλεγμάτων και, τελικά, την αριθμητική επίλυση των εξισώσεων ροής στο προηγούμενο πλέγμα. Από τα αποτελέσματα της αριθμητικής αυτής επίλυσης, με μετεπεξεργασία, υπολογίζεται ο βαθμός απόδοσης (ή άλλα συναφή μεγέθη) κάθε συστήματος που αποτελεί το στόχο σε ένα πρόβλημα βελτιστοποίησης.

Συγκεντρώνοντας τα όσα αναφέρθηκαν, συμπεραίνουμε ότι η βελτιστοποίηση, τα λογισμικά CAD και CAD - CAM - CAE, όπως το CATIA, αλλά και η χρήση των CFD επί του αντίστοιχου αντικειμένου, αποτελούν ένα αναπόσπαστο κομμάτι του συνεχώς εξελισσόμενου κλάδου της βιομηχανίας. Όταν πρόκειται ειδικά για σχεδιασμό και κατασκευή προϊόντων πολύπλοκης γεωμετρίας ή για υλοποίηση εφαρμογών με υψηλές απαιτήσεις ακρίβειας και πολύ συγκεκριμένους στόχους, φαντάζει αδύνατη η επίτευξη αυτών χωρίς τα ανωτέρω και ιδιαίτερα χωρίς CAD, των οποίων οι δυνατότητες αυξάνονται συνεχώς με ραγδαίους ρυθμούς.

1.1 Σκοπός της Παρούσας Εργασίας

Η παρούσα εργασία στοχεύει στην ένταξη του λογισμικού CATIA σε υπάρχουσα μέθοδο βελτιστοποίησης, δίνοντας τη δυνατότητα ενσωμάτωσης, στη διαδικασία αυτής, της σχεδίασης του προς βελτιστοποίηση αντικειμένου αλλά κυρίως, με την εκμετάλλευση του engineering (CAE) τμήματός του, της προσθήκης, βελτίωσης και αυτοματοποίησης διεργασιών σχετικών με την δημιουργία μη-δομημένου πλέγματος (unstructured computational grid). Λογική συνέχεια του

τελευταίου αποτελεί είτε η προσομοίωση (simulation) καταστάσεων φορτίσεως και καταπόνησης του αντικειμένου από το ίδιο το CATIA, είτε η μαθηματική επίλυση άλλων προβλημάτων (λ.χ. πεδίων ροής γύρω ή μέσα στις σχεδιαζόμενες γεωμετρίες) από κατάλληλο εξωτερικό λογισμικό (λ.χ. λογισμικό CFD).

Προφανώς, ο συνδυασμός των πλεονεκτημάτων που προσφέρει το CATIA με τα ήδη γνωστά της βελτιστοποίησης θα αναδείξουν το τελικό αποτέλεσμα των όσων ακολουθούν ως ένα χρησιμότερο εργαλείο αυτοματοποιημένης βελτιστοποίησης δύσκολων μορφών κατά τη σχετική έρευνα σε πανεπιστήμια, ερευνητικά κέντρα ή/και βιομηχανίες.

Αν και το CATIA είναι ένα από τα αρκετά λογισμικά τύπου CAD – CAM – CAE που κυκλοφορούν παγκοσμίως, εν τούτοις επιλέχθηκε έναντι των άλλων λόγω της διαθεσιμότητάς του στο Εργαστήριο Θερμικών Στροβιλομηχανών ΕΜΠ (για συντομία θα αναφέρεται στη συνέχεια: ΕΘΣ), του μεγάλου εύρους των δυνατοτήτων του και της ευρύτατης χρήσης του από βιομηχανίες σε ολόκληρο τον κόσμο.

1.2 Δομή της Παρούσας Εργασίας

Παρακάτω, περιγράφονται αρχικά οι εξελικτικοί αλγόριθμοι και ο τρόπος λειτουργίας τους, στοιχεία τα οποία κρίθηκαν απαραίτητα για την εκπόνηση της διπλωματικής εργασίας, αν και η εργασία αυτή δεν επένδυσε στην ανάπτυξη (αλλά μόνο στη χρήση) τέτοιων αλγορίθμων. Ακολουθώς γίνεται μία πρώτη γνωριμία με το λογισμικό CATIA και στη συνέχεια παρουσιάζεται η συμβολή του στη διαδικασία βελτιστοποίησης μέσω εξελικτικών αλγορίθμων, τα όσα υλοποιήθηκαν για την πραγματοποίηση του σκοπού αυτού αλλά και τα παράγωγα αυτής της υλοποίησης (αρχεία, κώδικες κ.λπ.).

Επίσης, αναφέρονται τα βασικά χαρακτηριστικά του επιλύτη των εξισώσεων Navier – Stokes του ΕΘΣ καθώς και οι παράμετροι που ορίστηκαν σε αυτόν για τη χρήση του στην παρούσα διπλωματική.

Τέλος, περιγράφεται και δοκιμάζεται η συμβολή του λογισμικού αυτού σε διαδικασία βελτιστοποίησης, μέσω συγκεκριμένης εφαρμογής, στο ΕΘΣ ακολουθούμενη από συμπεράσματα όσον αφορά τα αποτελέσματα που προκύπτουν, τυχόν αδυναμίες του συνόλου υπό τη δημιουργηθείσα μορφή αλλά και προτάσεις θεμάτων μελλοντικής εργασίας επί του θέματος με σκοπό την περαιτέρω ανάπτυξη και τελειοποίηση της διαδικασίας βελτιστοποίησης μέσω CATIA.

2. Εξελικτικοί Αλγόριθμοι Βελτιστοποίησης

Στο συγκεκριμένο κεφάλαιο αναφέρονται ορισμένα στοιχεία σχετικά με τους εξελικτικούς αλγορίθμους. Προφανώς δεν γίνεται εις βάθος αναφορά σε αυτούς αλλά μία γενική περιγραφή όσων θεωρούνται απαραίτητα για την πληρότητα τις συγκεκριμένης διπλωματικής, αφού με έναν τέτοιο αλγόριθμο θα συνεργαστεί το CATIA.

2.1 Σύντομη Ιστορική Αναδρομή Μεθόδων Βελτιστοποίησης

Η πρώτη τεχνική βελτιστοποίησης, γνωστή ως *steepest descend*, δημιουργήθηκε από τον Gauss μετά το 1800. Ιστορικά όμως, ο πρώτος όρος που χρησιμοποιήθηκε για τις τεχνικές αυτές ήταν ο “γραμμικός προγραμματισμός” (*linear programming*), ο οποίος εισάχθηκε από τον George Dantzig κατά την δεκαετία του 1940. Ο όρος προγραμματισμός αρχικά δεν αναφερόταν στη χρήση ηλεκτρονικών υπολογιστών, αν και σήμερα οι υπολογιστές χρησιμοποιούνται ευρέως στην επίλυση μαθηματικών προβλημάτων. Αντ’ αυτού, ο όρος αναφέρεται στη λειτουργία προγράμματος-τομέα του στρατού των Ηνωμένων Πολιτειών Αμερικής το οποίο ασχολούνταν με τον προτεινόμενο χρονικό προγραμματισμό της εκπαίδευσης των αμερικανών στρατιωτών και των logistics του στρατού (αποθήκες, μεταφορές κ.λπ.), προβλήματα που μελετούσε ο Dantzig εκείνη την εποχή. Επιπλέον, αργότερα, ο όρος αυτός φάνηκε πολύ χρήσιμος για την απόκτηση κυβερνητικών κονδυλίων μια και παρέπεμπε σε μελέτες υψηλής τεχνολογίας που θεωρούνταν πολύ σημαντικές από τους τότε κυβερνώντες της Αμερικής.

Πιο συγκεκριμένα, μόλις ξέσπασε ο 2ος Παγκόσμιος Πόλεμος, ο Dantzig ανέλαβε διευθυντής του Combat Analysis Branch, ενός από τους κλάδους ανάλυσης μαχών στα αρχηγεία του αμερικανικού στρατού, όπου προσπαθούσε να αντιμετωπίσει προκλήσεις όπως οι αλυσίδες ανεφοδιασμών και ο έλεγχος εκατοντάδων χιλιάδων αντικειμένων-προϊόντων και ανθρώπων. Από τη θέση αυτή, όπως είναι κατανοητό, έπρεπε να επιλύει προβλήματα σε πραγματικούς χρόνους και με σημαντικές επιπτώσεις. Έτσι, με βάση την εμπειρία του κατά τα χρόνια του πολέμου αλλά και τις γνώσεις του, εισήγαγε το 1947 το πρόβλημα του γραμμικού προγραμματισμού και πρότεινε τη μέθοδο Simplex για την επίλυση αυτού.

Το 1952 έγινε μαθηματικός ερευνών στην RAND Corporation, όπου άρχισε να πειραματίζεται με τη χρήση του γραμμικού προγραμματισμού πάνω στους υπολογιστές.

Από το σημείο αυτό και μετά, έγινε φανερό σε όλο τον κόσμο το πόσο σημαντική είναι η συμβολή του ανθρώπου αυτού στην επίλυση πολύπλοκων μαθηματικών μοντέλων. Κατ' επέκταση άρχισε να εμφανίζεται μία τάση “μαθηματικοποίησης” των προβλημάτων που εμφανίζονταν στον πραγματικό κόσμο και επιζητούσαν την καλύτερη ανάμεσα σε πολλές λύσεις τους. Ταυτόχρονα, η ταχύτατη ανάπτυξη των ηλεκτρονικών υπολογιστών έκανε προσβάσιμη την ισχύ τους σε ολόένα και περισσότερους ανθρώπους με αποτέλεσμα την ανάπτυξη και εξέλιξη τέτοιου τύπου μεθόδων από πολλούς μελετητές (αλλά και απλούς ανθρώπους στη σημερινή εποχή), κάτι το οποίο είναι δύσκολο να καταγραφεί ιστορικά.

Συμπερασματικά, η βάση την οποία δημιούργησε ο Dantzig κατά την εποχή του 2ου Παγκοσμίου πολέμου χρησιμοποιήθηκε και αναπτύχθηκε από πολλούς, πήρε διάφορες μορφές αναλόγως τις εφαρμογές χρήσης της και κατέληξε στη σημερινή εποχή ένα αναπόσπαστο κομμάτι των περισσότερων επιστημών, ως επί το πλείστον σε επίπεδο ερευνών και μελετών.

2.2 Βελτιστοποίηση μέσω Εξελικτικών Αλγορίθμων

Οι Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms - EAs) [Gian00] αποτελούν, στη σημερινή εποχή, μία από τις ευρύτερα χρησιμοποιούμενες στοχαστικές μεθόδους βελτιστοποίησης. Τα βασικά στοιχεία που έχει να επιδείξει αυτή η μέθοδος, έναντι των υπολοίπων, και που οδήγησαν στην επικράτησή της, σχετίζονται με την έλλειψη μαθηματικού υπόβαθρου, την εύκολη προσαρμογή της σε κάθε νέο πρόβλημα, την δυνατότητα επίλυσης προβλημάτων πολλών στόχων αλλά και την ικανότητά της να μην εγκλωβίζεται σε τοπικά ακρότατα. Το τελευταίο ειδικά αποτελεί πρόβλημα πολλών άλλων μεθόδων οι οποίες καταλήγουν να δίνουν ανεπαρκείς λύσεις. Σημαντικό πλεονέκτημα των EA είναι η άμεση χρήση τους σε κάθε νέο πρόβλημα, χωρίς παρεμβάσεις στη διαδικασία βελτιστοποίησης, αν και η σε βάθος γνώση του προβλήματος και χρήση στοιχείων από αυτή με τη μορφή τελεστών βοηθάει σημαντικά στη σύγκλιση της μεθόδου.

Κύριο χαρακτηριστικό των αλγορίθμων αυτών είναι ο χειρισμός πλήθους υποψηφίων λύσεων (population – based methods) σε κάθε γενιά (επανάληψη της

διαδικασίας), σε αντίθεση με άλλες ντετερμινιστικές αλλά και ορισμένες στοχαστικές μεθόδους (όπως π.χ. η μέθοδος της προσομοιούμενης ανόπτησης). Ο πληθυσμός αυτός υπόκειται σε διαδικασία εξέλιξης. Για το σκοπό αυτό γίνεται χρήση στοχαστικών μεθόδων εξέλιξης, προερχόμενων από φυσικά μοντέλα. Ο όρος εξέλιξη υπονοεί ακριβώς ότι και η θεωρία εξέλιξης των οργανισμών του Δαρβίνου (1960). Ότι δηλαδή, από τον τρέχον πληθυσμό των υποψήφιων λύσεων (τρέχουσα γενιά) θα προκύψει ένας νέος πληθυσμός “απογόνων” με ενδεχομένως καλύτερα χαρακτηριστικά από αυτά των “προγόνων”.

Η πιο διαδεδομένη μορφή εξελικτικών αλγορίθμων είναι οι γενετικοί αλγόριθμοι (generic algorithms – ΓΕ), ιδίως στον τομέα της αεροδυναμικής και των στροβιλομηχανών. Οι εξελικτικές στρατηγικές (evolution strategies – ΕΣ) αποτελούν μία άλλη διαδεδομένη τεχνική ενώ ο γενετικός προγραμματισμός (ΓΕ) είναι μία τρίτη κατηγορία αυτών, αν και διαφέρει αρκετά από τις υπόλοιπες διότι ως άτομα του πληθυσμού χρησιμοποιούνται προγράμματα υπολογιστή.

Κάθε κατηγορία των ΕΑ έχει τα δικά της διακριτά γνωρίσματα, ωστόσο οι σύγχρονοι ΕΑ αλληλοδανείζονται στοιχεία όπως οι τελεστές εξέλιξης ή ο τρόπος κωδικοποίησης, ανάλογα με την εμπειρία ή την καταλληλότητά τους ανά περίπτωση. Παρ’ όλ’ αυτά, για να χαρακτηριστεί κάποιος αλγόριθμος βελτιστοποίησης ως εξελικτικός πρέπει να έχει τα ακόλουθα:

- Να χρησιμοποιεί πληθυσμούς ατόμων, αντί μεμονωμένων, τα οποία εξελίσσονται συγχρόνως.
- Η εξέλιξη του πληθυσμού από γενιά σε γενιά να καθορίζεται από την τιμή καταλληλότητας (μεγιστοποίηση) ή την τιμή κόστους ατόμων (ελαχιστοποίηση), με βάση κατάλληλη συνάρτηση – στόχο.
- Να χαρακτηρίζεται από δυναμική διαδοχή πληθυσμών, με τη δημιουργία νέων ατόμων και την διαγραφή άλλων βάσει της τιμής καταλληλότητάς τους.
- Να υπάρχει κληρονομικότητα χαρακτηριστικών από τους γονείς στους απογόνους, κατά την εξέλιξη από γενιά σε γενιά, αλλά και στοχαστική εμφάνιση νέων χαρακτηριστικών.

2.2.1 Περιγραφή Εξελικτικού Αλγορίθμου

Αναφέρθηκε ήδη ότι οι εξελικτικοί αλγόριθμοι χρησιμοποιούν, για την επίλυση ενός προβλήματος, έναν πληθυσμό υποψήφιων λύσεων, κατά την εξέλιξη των οποίων, ένα σύνολο από μ άτομα (γονείς) παράγει ένα πληθυσμό λ απογόνων. Αυτοί ακριβώς οι απόγονοι είναι και οι νέες υποψήφιες λύσεις που πιθανότατα να εμφανίζουν ικανοποιητικότερα αποτελέσματα ως προς την επίτευξη του αρχικού στόχου, συγκριτικά με τους “γονείς” τους. Από το σύνολο αυτών ο αλγόριθμος επιλέγει, βασιζόμενος στην καταλληλότητα καθενός, εκείνους που θα αποτελέσουν τους μ γονείς της επόμενης γενιάς. Η διαδικασία αυτή επαναλαμβάνεται έως ότου ικανοποιηθεί κάποιο προκαθορισμένο κριτήριο σύγκλισης. Τέτοιου είδους κριτήρια είναι συνήθως τα παρακάτω:

- I. Το μέγεθος βελτίωσης της λύσης μετά από ένα αριθμό αξιολογήσεων ή γενεών
- II. Η επαρκής ομογενοποίηση του πληθυσμού
- III. Η ανάλωση του υπολογιστικού χρόνου που έχει επιτραπεί για την εκάστοτε εφαρμογή

Στη συνέχεια, αναφέρονται τα βήματα που ακολουθούνται κατά την εκτέλεση ενός ΕΑ [Gian00] :

1^ο Βήμα:

Καθορίζονται βασικές παράμετροι της μεθόδου όπως το μέγεθος των δύο βασικών πληθυσμών (μ για τους γονείς και λ για τους απογόνους), το κριτήριο σύγκλισης κ.α. Επιλέγεται τυχαία η μηδενική γενιά, δηλαδή τα μ μέλη του αρχικού πληθυσμού γονέων.

2^ο Βήμα:

Γίνεται αξιολόγηση των λ απογόνων με βάση μία συνάρτηση στόχο. Σε κάθε έναν από αυτούς αποδίδεται εν τέλει μία τιμή καταλληλότητάς τους (fitness score).

Σε αυτό το σημείο, εάν ο στόχος περιλαμβάνει την βελτιστοποίηση μορφής σώματος στην ρευστομηχανική, αεροδυναμική κ.λπ., όπως π.χ. στην εφαρμογή που θα ακολουθήσει, απαιτείται η αριθμητική επίλυση του πεδίου ροής για την αξιολόγηση κάθε ατόμου. Αυτό συνεπάγεται την κλήση ενός λογισμικού αριθμητικής επίλυσης των εξισώσεων Navier-Stokes για κάθε έναν απόγονο (λ φορές συνολικά). Προφανώς, κάτι τέτοιο είναι επίπονο όσον αφορά το υπολογιστικό κόστος.

3ο Βήμα:

Με κριτήριο την τιμή καταλληλότητας, επιλέγεται, από κάποιους μηχανισμούς και από το σύνολο των γονέων και των απογόνων, ο πληθυσμός των γονέων της νέας γενιάς.

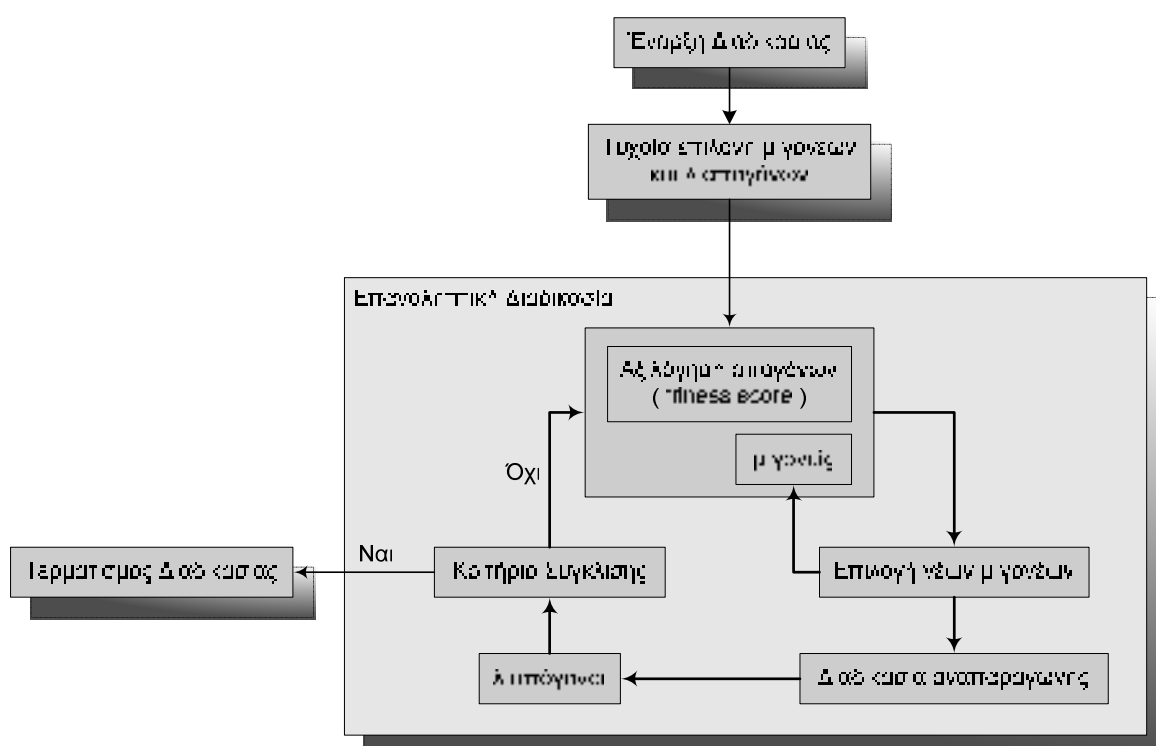
4ο Βήμα:

Οι γονείς της νέας γενιάς υπόκεινται σε διαδικασία αναπαραγωγής για να δώσουν τους λ απογόνους της νέας γενιάς. Πρόκειται για την διαδοχική εφαρμογή διαφόρων τελεστών (π.χ. διασταύρωσης, μετάλλαξης)

5ο Βήμα:

Ελέγχεται το κριτήριο σύγκλισης. Εάν αυτό ικανοποιείται, τότε οι λ απόγονοι της τρέχουσας γενιάς αποτελούν και την έξοδο του ΕΑ ως λύση του προβλήματος. Εάν όμως δεν ικανοποιείται ο αλγόριθμος επαναλαμβάνει τα βήματα 2 έως και 5.

Οι εξελικτικοί αλγόριθμοι έχουν βρει εφαρμογή σε πολλούς τομείς βελτιστοποίησης και θα μας απασχολήσουν στη συνέχεια με την εφαρμογή ενός από αυτούς κατά την εκπόνηση της παρούσας εργασίας.



Σχήμα 2.1: Οπτική Απεικόνιση Εξελικτικού Αλγορίθμου

2.2.2 Μειονέκτημα των ΕΑ – Κόστος Υπολογιστικού Χρόνου

Παρά την χρησιμότητα και την επικράτηση των Εξελικτικών Αλγορίθμων, έχουν ένα βασικό μειονέκτημα το οποίο επηρεάζει σημαντικά την εφαρμοσιμότητά τους και τη χρήση τους γενικότερα. Σε κάθε επανάληψη της μεθόδου, απαιτείται η αξιολόγηση κάθε ατόμου του τρέχοντος πληθυσμού απογόνων και η απόδοση σε αυτούς ενός fitness score. Εύκολα συμπεραίνουμε ότι, εάν η αξιολόγηση της τιμής της αντικειμενικής συνάρτησης (objective function evaluation – OFE) για ένα άτομο είναι απαιτητική, τότε, ο συνολικός χρόνος που θα χρειαστεί θα είναι απαγορευτικά μεγάλος. Ταυτόχρονα, οι ανάγκες σε υπολογιστική ισχύς θα είναι υπερβολικά αυξημένες και ασύμφορες.

Είναι προφανές ότι σε περιπτώσεις αεροδυναμικής βελτιστοποίησης ή ρευστομηχανικών εφαρμογών, όπου, για την αξιολόγηση, γίνεται κλήση λογισμικού αριθμητικής επίλυσης των εξισώσεων Navier-Stokes (κώδικας Υπολογιστικής Ρευστομηχανικής), το μειονέκτημα που προαναφέρθηκε γίνεται ιδιαίτερα εμφανές.

Αποτέλεσμα αυτού του γεγονότος είναι η προσπάθεια μείωσης του υπολογιστικού χρόνου, σε τέτοιες εφαρμογές, με διάφορους τρόπους. Ο γενικός στόχος για την αποτελεσματική εξοικονόμηση χρόνου είναι η μείωση του αριθμού κλήσεων του κώδικα Υπολογιστικής Ρευστοδυναμικής. Πιο συγκεκριμένα, ο κώδικας αυτός (που θα αποκαλείται “*λογισμικό ακριβούς αξιολόγησης*”) καλείται να “τρέξει” για ορισμένα από τα μέλη του πληθυσμού και όχι για όλα.

Η λογική της προσεγγιστικής προ-αξιολόγησης αποτρέπει κατ’ αρχάς την αξιολόγηση όλων των πιθανών λύσεων, υποσχόμενων και μη με το λογισμικό ακριβούς αξιολόγησης. Υπάρχουν δύο κατηγορίες μεθόδων που επικεντρώνονται σε αυτό, οι οποίες αναφέρονται επιγραμματικά , αφού περαιτέρω ανάλυση θεωρείται περιττή:

- Χαμηλότερης ακρίβειας μοντέλα αξιολόγησης (surrogate evaluation models), που επιλύουν το φυσικό πρόβλημα με περισσότερες παραδοχές, μικρότερη ακρίβεια και, άρα, περισσότερο οικονομικά.
- Υποκατάστατα μοντέλα προσέγγισης - εκτίμησης (surrogate approximation models) ή αλλιώς μεταπρότυπα.

Όσον αφορά την πρώτη κατηγορία, πρέπει να εξετάζονται οι παραδοχές του μοντέλου και η ακρίβειά του έτσι ώστε να διεξάγονται σωστά αποτελέσματα για την

επίλυση του φυσικού προβλήματος. Στην περίπτωση των μεταπροτύπων είναι απαραίτητη η ύπαρξη γνώσεων πάνω στο πρόβλημα που εξετάζεται και η χρήση μιας βάσεως δεδομένων με εκτιμήσεις, αξιολογήσεις και επιδόσεις παλαιότερων λύσεων.

2.3 EASY (Evolutionary Algorithms System)

Το λογισμικό EASY (Evolutionary Algorithms System) είναι ένα λογισμικό βελτιστοποίησης γενικής χρήσης, το οποίο είναι ικανό να επιλύσει προβλήματα ενός αλλά και πολλαπλών στόχων. Βασίζεται στους εξελικτικούς αλγόριθμους και έχει αναπτυχθεί και διατίθεται για χρήση από το Εργαστήριο Θερμικών Στροβιλομηχανών του Εθνικού Μετσόβιου Πολυτεχνείου.

Είναι προγραμματισμένο με τέτοιο τρόπο ώστε να διαχειρίζεται προβλήματα ελαχιστοποίησης. Φυσικά, είναι εφικτή και η διαχείριση προβλημάτων μεγιστοποίησης, εφόσον προηγηθεί κατάλληλη μετατροπή τους σε τύπου ελαχιστοποίησης, προτού χρησιμοποιηθεί το λογισμικό. Οι συναρτήσεις στόχος του εκάστοτε προβλήματος μπορούν να είναι είτε θετικές είτε αρνητικές. Έχει χρησιμοποιηθεί εκτεταμένα σε εφαρμογές που αφορούν επιστήμες μηχανικών.

Βασικό του χαρακτηριστικό είναι η δυνατότητα καθορισμού όλων των παραμέτρων της βελτιστοποίησης που παρέχεται κυρίως για τους έμπειρους χρήστες, ενώ παράλληλα υπάρχουν ενσωματωμένα πακέτα επιλογών προς χρήση για τους αρχάριους. Το περιβάλλον διασύνδεσης του EASY με τον χρήστη έχει σχεδιαστεί σε πλατφόρμα Java, ενώ το βασικό σύστημα εξέλιξης έχει προγραμματιστεί σε γλώσσα προγραμματισμού C++ [Eas00] .

Το συγκεκριμένο λογισμικό θα χρησιμοποιηθεί, στα πλαίσια της παρούσας διπλωματικής εργασίας, για την επίλυση ενός προβλήματος βελτιστοποίησης σε συνεργασία με το CATIA συμβάλλοντας έτσι στην ενσωμάτωση του τελευταίου στο σύνολο της διαδικασίας, γεγονός που αποτελεί τον πρωταρχικό στόχο.

3. Το Λογισμικό CATIA

Έχει γίνει ήδη κατανοητό ότι το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η προσθήκη και η αυτοματοποίηση διαδικασιών, σε υπάρχουσα μέθοδο βελτιστοποίησης, διαμέσου του λογισμικού CATIA. Σε αυτό το σημείο κρίνεται σκόπιμη η περιγραφή αυτού του προγράμματος έτσι ώστε ο αναγνώστης να αποκτήσει την επαφή που απαιτείται για την κατανόηση όσων ακολουθούν...

3.1 Γενικά

Το λογισμικό **Computer Aided Three-dimensional Interactive Application**, εν συντομία CATIA, αποτελεί μία ολοκληρωμένη πλατφόρμα για CAD (Computer Aided Design), CAM (Computer Aided Manufacturing) και CAE (Computer Aided Engineering) εφαρμογές ψηφιακού σχεδιασμού και προσομοίωσης προϊόντων. Αναπτύχθηκε στα τέλη της δεκαετίας του 70' από την γαλλική εταιρία **Dassault Systems**, θυγατρική εταιρία της Dassault Aviation με αρχικό στόχο την εσωτερική χρήση του από την τελευταία και την δημιουργία του γνωστού μαχητικού Mirage. Το 1981 άρχισε να πωλείται και να διανέμεται από την IBM ενώ το 1984 επιλέχθηκε από την Boeing Company ως το βασικό σχεδιαστικό πρόγραμμα της εταιρίας. Το 1988 δημιουργήθηκε νέα έκδοση του λογισμικού το οποίο από τότε συνεχώς αναπτύσσεται και εξελίσσεται. Σήμερα, το CATIA είναι μία πλατφόρμα λογισμικού η οποία αναφέρεται στη διαχείριση πολλαπλών σταδίων του κύκλου ζωής ενός προϊόντος (π.χ. ο τρισδιάστατος σχεδιασμός του, η κατασκευή του και η ανάλυση και προσομοίωσή του).



Το τελευταίο αποτελεί και ένα από τα σημαντικότερα πλεονεκτήματά του αφού με αυτόν τον τρόπο δίνει ολοκληρωμένες λύσεις στον οποιοδήποτε χρήστη του. Ένα άλλο βασικό πλεονέκτημα είναι το ενιαίο υπόβαθρο πάνω στο οποίο είναι κατασκευασμένο το CATIA με αποτέλεσμα την εύκολη συνεργασία τμημάτων του που εξειδικεύονται σε διαφορετικά στάδια του κύκλου ζωής όπως μηχανολογικού σχεδιασμού, ψηφιακού ελέγχου, εκτέλεσης κατεργασιών, ανάλυσης και προσομοίωσης κ.λπ. Επίσης, η προσαρμογή του στις εκάστοτε ανάγκες της εφαρμογής μέσω προγραμματιστικών περιβάλλοντων ενσωματωμένων στο λογισμικό (Fortran και C στην έκδοση 4, Visual Basic στην έκδοση 5) καθιστά αποτελεσματικότερη τη χρήση του ενώ τέλος, η δυνατότητα επανάκτησης

πληροφοριών και συμπερασμάτων που έχουν προκύψει από προηγούμενες εφαρμογές διευκολύνουν την χρήση του επιταχύνοντας ταυτόχρονα τη συνολική διαδικασία.

Η συμβολή του στην ανάπτυξη της παγκόσμιας βιομηχανίας αντικατοπτρίζεται από τη συνεχώς αυξανόμενη εκμετάλλευσή του από πολλές και μεγάλες εταιρίες διαφόρων κλάδων: αεροπλοΐα (Boeing, Airbus, Bombardier Aerospace), αυτοκινητοβιομηχανία (BMW, Porsche, Daimler Chrysler, Audi, Volkswagen, Volvo, Fiat, Gestamp Automocion, Benteler AG, PSA Peugeot Citroën, Renault, Toyota, Honda, Ford, Scania, Hyundai, Proton, Tata motors and Mahindra), ναυσιπλοΐα (GD Electric Boat για σχεδιασμό υποβρυχίων και Northrop Grumman Newport News για τον σχεδιασμό αεροπλανοφόρων του αμερικανικού ναυτικού), κατασκευή βιομηχανικού εξοπλισμού, ηλεκτρικών και ηλεκτρονικών εφαρμογών, σχεδιασμού εργοστασιακών εγκαταστάσεων, καταναλωτικών αγαθών κ.α. [Dgt01]

Η συνεχής εξέλιξή του από την κατασκευάστρια εταιρία έχει σαν αποτέλεσμα μία πληθώρα εκδόσεων του προγράμματος. Παρακάτω περιγράφεται η έκδοση CATIA V5R18 (version 5 release 18) η οποία και χρησιμοποιήθηκε για την υλοποίηση των στόχων που τέθηκαν στη συγκεκριμένη διπλωματική.

3.2 Τμήματα του CATIA

Όπως κάθε ολοκληρωμένο λογισμικό πακέτο, έτσι και το CATIA, είναι διαχωρισμένο σε τμήματα κάθε ένα από τα οποία έχει τον δικό του ρόλο και το δικό του περιβάλλον εντός του συνόλου. Στην επόμενη σελίδα αναφέρονται τα σημαντικότερα από αυτά, βάσει του διαχωρισμού που έχει γίνει από την ίδια την κατασκευάστρια εταιρία [Ibm00].

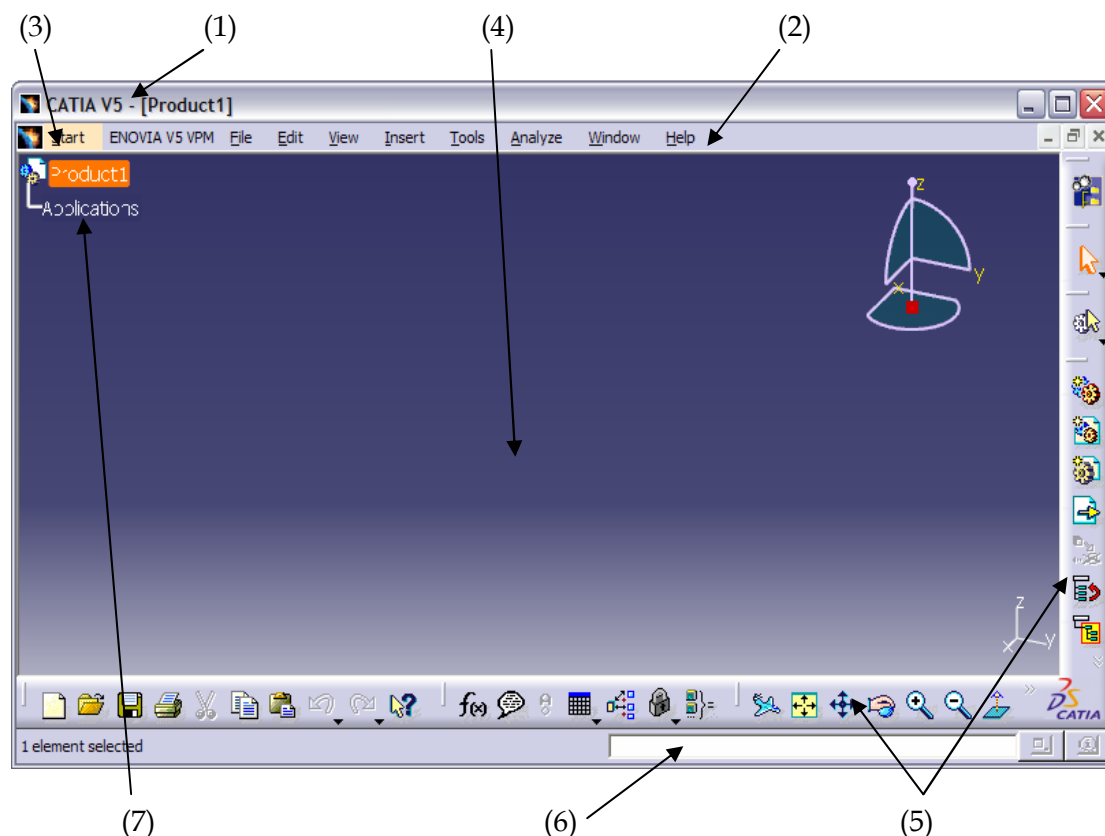
Πίνακας 3.1: Τμήματα CATIA

Ονομασία	Σύντομη περιγραφή
Μηχανολογικός Σχεδιασμός (Mechanical Design)	Προσφέρει μοντελοποίηση βάσει προδιαγραφών για σχεδιασμό ενός τεμαχίου, πληθώρα αυτών σε συνεργασία (assembly) και μηχανολογικού σχεδίου.
Σχεδιασμός Μορφής και Σχήματος (Shape Design and Styling)	Επιτρέπει την δημιουργία και τροποποίηση πολύπλοκων επιφανειών
Σύνθεση Προϊόντων (Product Synthesis)	Προσφέρει αυτοματοποίηση και έλεγχο σχεδίων και κατασκευαστικών δεδομένων
Μηχανική Εξοπλισμού και Συστημάτων (Equipment and System Engineering)	Πραγματοποιεί σχεδιασμό και ολοκλήρωση ηλεκτρικών, υδραυλικών και μηχανολογικών συστημάτων
Ανάλυση (Analysis)	Επιτρέπει την κατασκευαστική ανάλυση και προσομοίωση προϊόντων
Κατεργασίες (Machining)	Προσφέρει πληθώρα γνώσεων και πληροφοριών βασισμένη σε ολοκληρωμένη υποδομή, η οποία καλύπτει όλες τις εφαρμογές μηχανουργικών κατεργασιών
Υποδομή (Infrastructure)	Περιέχει το σύνολο των πληροφοριών και των στοιχείων που απαιτούνται για τη συνεργασία των επιμέρους τμημάτων

3.3 Το περιβάλλον του CATIA

Η ανάγκη συνεργασίας των επιμέρους τμημάτων απαιτεί την ύπαρξη μίας βασικής δομής, πάνω στην οποία “χτίστηκε” το συγκεκριμένο λογισμικό. Σε αυτή την δομή ανήκει και το γραφικό περιβάλλον του CATIA (GUI - Graphics Users

Interface) το οποίο επιτρέπει την επικοινωνία του λογισμικού με τον χρήστη και μέσω του οποίου εκτελούνται όλες οι λειτουργίες του. Κύριος εκφραστής του GUI είναι το κεντρικό παράθυρο του CATIA, το οποίο φέρει ορισμένα γενικά χαρακτηριστικά, ανεξαρτήτως επιμέρους περιβάλλοντος και παρουσιάζεται στο σχήμα 3.1.

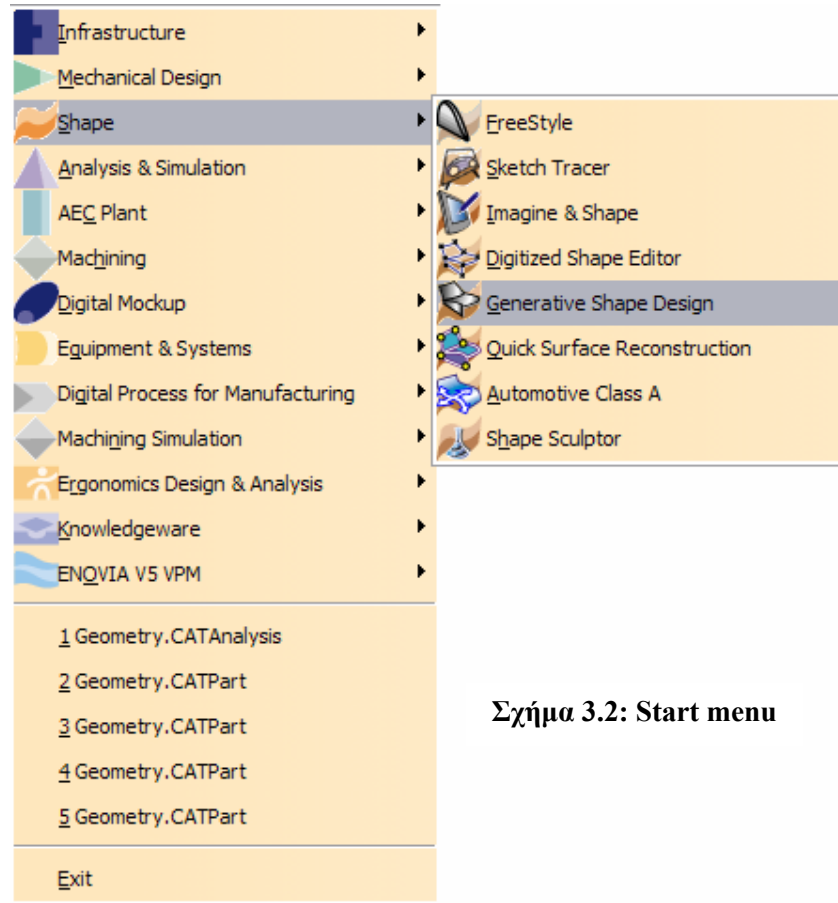


Σχήμα 3.1: Κεντρικό παράθυρο CATIA

Πρόκειται για ένα κλασικό παράθυρο μορφής Windows στο πάνω μέρος του οποίου φαίνεται η ονομασία του προγράμματος στο οποίο ανήκει, δηλαδή αναφέρεται η ονομασία και η έκδοση του CATIA (1), ακολουθούμενη από την ονομασία και τον τύπο του προς επεξεργασία αρχείου

Ακριβώς κάτω από αυτό βρίσκεται η μπάρα των βασικών menu (2) μέσω της οποίας εκτελούνται λειτουργίες όπως άνοιγμα και αποθήκευση αρχείων, εκτύπωση, ρυθμίσεις που αφορούν το λογισμικό, καθορισμός προτιμήσεων κ.α. Η μπάρα αυτή χρησιμοποιείται σε όλα τα λογισμικά που λειτουργούν σε περιβάλλον Windows και θεωρείται αρκετά γνωστή, αν και εμπλουτισμένη με εξειδικευμένες λειτουργίες του CATIA.

Στην αρχή της μπάρας αυτής εμφανίζεται ένα menu με την ονομασία Start (3). Μέσω αυτού γίνονται προσβάσιμα τα διάφορα επιμέρους τμήματα του CATIA και είναι εφικτή η συνεργασία τους, χωρίς πολύπλοκες διαδικασίες, αφού η διαμεταγωγή δεδομένων γίνεται αυτόματα. Στην ουσία το menu αυτό καταγράφει τα διάφορα περιβάλλοντα που ανήκουν στο κάθε τμήμα και επιτρέπει, μέσω της επιλογής τους, την μεταβίβαση από το ένα στο άλλο.

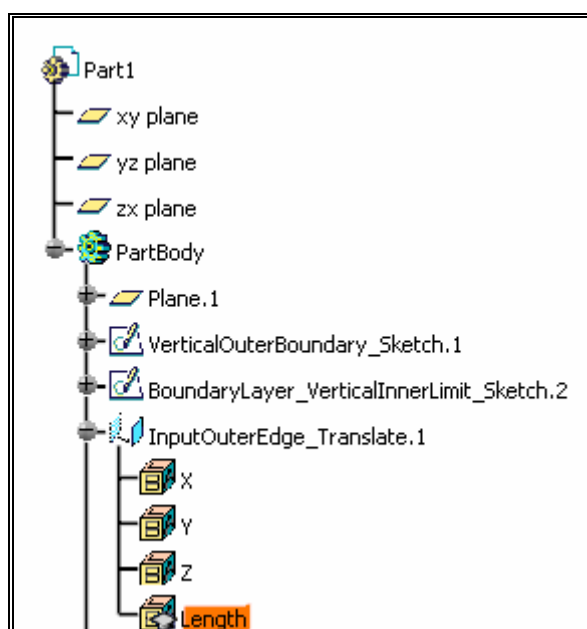


Σχήμα 3.2: Start menu

Όπως είναι φυσικό, το μεγαλύτερο μέρος του παραθύρου καταλαμβάνεται από το ενεργό περιβάλλον του λογισμικού (4) το οποίο έχει επιλεγεί από το Start menu. Σε αυτό γίνεται η απεικόνιση των προς επεξεργασία προϊόντων - σχεδίων με, κατά προεπιλογή, μπλε φόντο. Περιμετρικά αυτού βρίσκονται οι γραμμές εργαλείων (5) οι οποίες μεταβάλλονται δυναμικά, ανάλογα με το ενεργό περιβάλλον, και περιέχουν τις συντομεύσεις για όλα τα εργαλεία και τις εντολές που ανήκουν στο τελευταίο.

Στο κάτω μέρος του παραθύρου βρίσκεται η γραμμή εντολών (6) η οποία προσφέρει τη δυνατότητα καταγραφής εντολών σε μορφή κειμένου και εκτέλεσής τους.

Τέλος, βασικό γνώρισμα του GUI του CATIA είναι η εμφάνιση ενός δέντρου (7) συγκεκριμένης μορφής (σχήμα 3.3), το οποίο δημιουργείται κατά τον σχεδιασμό των προϊόντων. Σε αυτό καταγράφονται όλα τα χαρακτηριστικά που προκύπτουν από την εκτέλεση εντολών συμπεριλαμβανομένων οντοτήτων σχεδιασμού (επιφάνειες, γραμμές κ.λπ.) και αριθμητικών τιμών παραμέτρων (δυνάμεις, διαστάσεις κ.λπ.) καθώς και των σχέσεων και αλληλοεξαρτήσεων αυτών. Αποτελεί πολύ χρήσιμο χαρακτηριστικό διότι επιτρέπει την αποτύπωση της λογικής που υιοθετήθηκε κατά το σχεδιασμό και την ανασκόπηση πληροφοριών κατά την περαιτέρω επεξεργασία του προϊόντος.

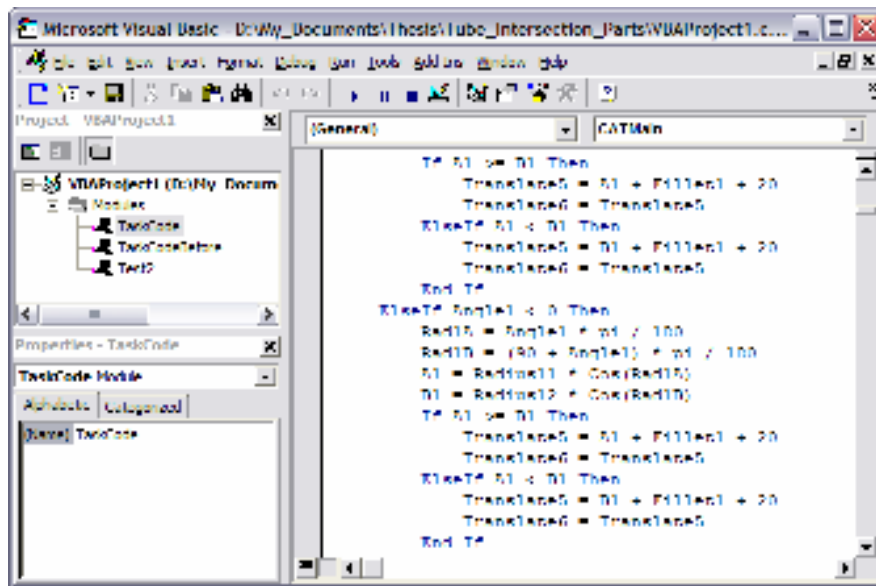


Σχήμα 3.3: Τμήμα δέντρου χαρακτηριστικών σχεδιασμού για το προϊόν Part1

Προφανώς, η περιγραφή που προηγήθηκε αφορά τη μορφή του λογισμικού όπως αυτή έχει προκαθοριστεί από την κατασκευάστρια εταιρία. Παρόλ' αυτά υπάρχει και η δυνατότητα επεξεργασίας και προσαρμογής στοιχείων αυτής στις προτιμήσεις του χρήστη, μέσω της εντολής *Tools > Options*. [Cat00]

3.4 Visual Basic Editor

Όπως προαναφέρθηκε, βασικό χαρακτηριστικό και πλεονέκτημα του CATIA είναι η δυνατότητα προσαρμοσμένης χρήσης ανά εφαρμογή, μέσω του ενσωματωμένου Visual Basic Editor. Το περιβάλλον του φαίνεται στο σχήμα 3.4 και είναι προσβάσιμο από το menu "*Tools*" της μπάρας βασικών menu εκτελώντας την εντολή *Macros > Visual Basic Editor*.



Σχήμα 3.4: Προγραμματιστικό περιβάλλον Visual Basic Editor

Οι δυνατότητες που προσφέρει ένα τέτοιο εργαλείο είναι απεριόριστες αφού πραγματοποιείται συνεργασία του CATIA με τη γλώσσα προγραμματισμού Visual Basic (VB). Αποτέλεσμα αυτού είναι η αυτοματοποίηση λειτουργιών του CATIA αλλά και ο εμπλουτισμός του με νέες, εκτελέσιμες από την VB. Παράγωγο της χρήσης του είναι οι λεγόμενες μακροεντολές (macros) οι οποίες έχουν ως στόχο την αυτόματη εκτέλεση επαναλαμβανόμενης σειράς ενεργειών εντός του CATIA έτσι ώστε να αποφεύγεται η ασύμφορη χειροκίνητη χρήση. Με απλά λόγια, πρόκειται για καταγραφή αλληλουχίας εντολών με σκοπό την μελλοντική εκτέλεση του συνόλου αυτών με το πάτημα ενός κουμπιού.

Επίσης, η χρήση του Visual Basic Editor επιτρέπει την κατασκευή ξεχωριστών interfaces, τα οποία έχουν το δικό τους, κατάλληλα προσαρμοσμένο GUI, μέσω του οποίου δίνουν τη δυνατότητα εκτέλεσης προεπιλεγμένων εντολών και μακροεντολών χρήσιμων για την εκάστοτε εφαρμογή.

Η δημιουργία των ανωτέρω προϋποθέτει γνώση της συγκεκριμένης γλώσσας προγραμματισμού και πραγματοποιείται με την πληκτρολόγηση των επιθυμητών γραμμών εντολών εντός ενός συνολικού κώδικα VB. Όσον αφορά τις λειτουργίες του CATIA, εκτός από την πληκτρολόγηση, υπάρχει και η δυνατότητα αυτόματης καταγραφής τους μέσω της εντολής *Tools > Macros > Start Recording...* η οποία μετατρέπει σε κείμενο κώδικα όσες από τις εντολές εκτελούνται χειροκίνητα από την επιλογή της έως και την διακοπή της [Cat00].

3.5 Γραμμή εντολών DOS

Πέραν του Visual Basic Editor, το CATIA δίνει τη δυνατότητα αυτοματοποίησης των λειτουργιών του και μέσω περιβάλλοντος εντολών DOS, το λεγόμενο *command prompt*. Από το παράθυρο αυτό πληκτρολόγησης εντολών είναι εφικτή η ενεργοποίηση του προγράμματος CATIA, κάτι το οποίο επιτρέπει το λογισμικό Windows, με την εκτέλεση του κατάλληλου αρχείου. Αυτό όμως που προσφέρεται από το CATIA, σε αυτή την περίπτωση, είναι μία λίστα από παραμέτρους που μπορούν να προστεθούν στην εκτέλεση αυτού του αρχείου οι οποίες συμπληρώνουν την διαδικασία έναρξης του προγράμματος με επιπρόσθετες ενέργειες, όπως αυτόματη εκτύπωση κ.λπ.

Από το ίδιο το λογισμικό παρέχονται έτοιμα πακέτα λειτουργιών, επεξεργάσιμα μέσω του περιβάλλοντος *Tools > Utility...* του CATIA, το οποίο ονομάζεται Batch Monitor.

Στη συνέχεια αναφέρονται δύο πολύ σημαντικές, για την παρούσα διπλωματική εργασία, παράμετροι που μπορούν να ακολουθήσουν την εντολή εκτέλεσης DOS του CATIA [Cat00]:

-macro “τοποθεσία και ονομασία της μακροεντολής”: επιτρέπει την εκτέλεση μίας μακροεντολής αμέσως μετά το άνοιγμα του CATIA.

-batch: υποχρεώνει τις υπόλοιπες λειτουργίες που συμπεριλαμβάνονται στην εντολή DOS να εκτελεστούν στην μνήμη του υπολογιστή, χωρίς παραθυρικό περιβάλλον και γραφική απεικόνιση, εξοικονομώντας έτσι χρόνο και υπολογιστικούς πόρους.

3.6 Απαιτήσεις

Σε συστήματα με λειτουργικό τα Microsoft Windows XP, το λογισμικό CATIA απαιτεί, ως ελάχιστα όρια, την ύπαρξη επεξεργαστή Pentium 4 και ανώτερου, σκληρού δίσκου 4 GB, 256 MB μνήμης RAM, κάρτας γραφικών με επιταχυντή 3D OpenGL και ανάλυση 1024 x 768. Για την εφαρμογή που περιγράφεται στην παρούσα διπλωματική εργασία προτείνεται η ύπαρξη διπύρηνου επεξεργαστή, μνήμης τουλάχιστον 2 GB και κάρτας γραφικών με δυνατότητα ανάλυσης 1280 x 1024 ενώ θεωρείται απαραίτητη η προεγκατάσταση του Service Pack 2 του λειτουργικού λογισμικού [Ibm00] .

4. Το CATIA στη Βελτιστοποίηση

Στο κεφάλαιο 3 παρουσιάστηκε λεπτομερώς το λογισμικό CATIA καθώς και οι πολλές δυνατότητες που έχει και προσφέρει στους διάφορους τομείς εφαρμογής του. Στο παρόν κεφάλαιο θα οριστεί η συμβολή του στη διαδικασία της βελτιστοποίησης, θα καθοριστούν τα τμήματά του που θα συμμετέχουν σε αυτή καθώς και οι λειτουργίες που αυτά θα επιτελέσουν ενώ τέλος θα ακολουθήσει μία πλήρης περιγραφή των όσων υλοποιήθηκαν στα πλαίσια της παρούσας διπλωματικής.

4.1 Ο Ρόλος του CATIA

Επειδή το CATIA είναι ένα σχεδιαστικό πακέτο, γίνεται κατανοητό ότι με τον όρο βελτιστοποίηση, στην προκείμενη περίπτωση, εννοείται η εύρεση μίας βέλτιστης γεωμετρίας, η οποία πρέπει να ικανοποιεί κάποιες συνθήκες και προϋποθέσεις που έχουν προκαθοριστεί από το πρόβλημα. Η περιγραφή της βέλτιστης αυτής λύσης γίνεται μέσω των τιμών ορισμένων μεταβλητών τις οποίες διαχειρίζεται ο εξελικτικός αλγόριθμος.

Βάσει των παραπάνω, αυτό που ζητείται να υλοποιηθεί από το CATIA συνοψίζεται στα εξής:

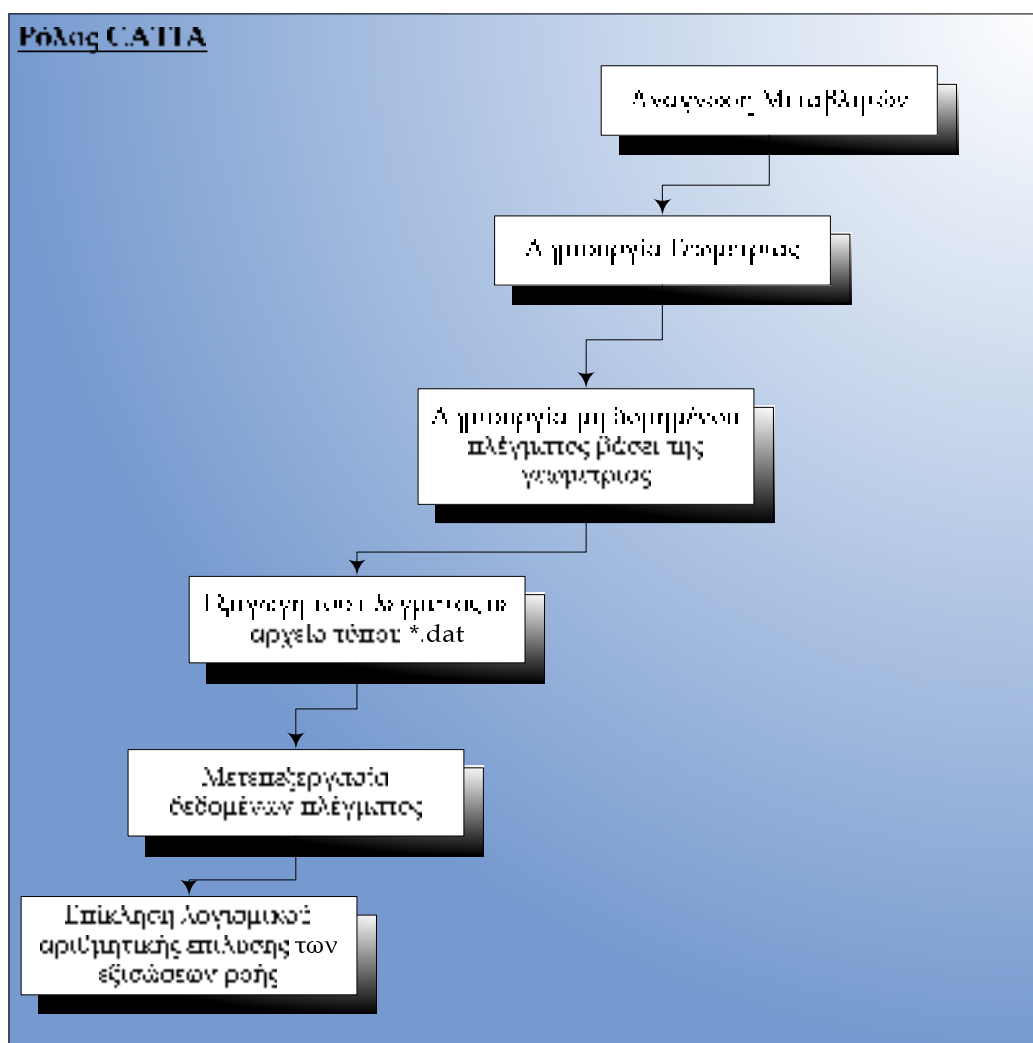
1. Αναγνώριση των τιμών που διαχειρίζεται ο εξελικτικός αλγόριθμος (το σύνολο των οποίων αποτελεί τη βάση για τη δημιουργία γονέων και απογόνων από αυτόν) και μετατροπή τους στην, προς εξέταση, αντίστοιχα ορισμένη γεωμετρία.
2. Δημιουργία ενός μη-δομημένου πλέγματος, βασισμένο σε αυτή τη γεωμετρία και παραμετροποιημένο αναλόγως με τις εκάστοτε ανάγκες της επίλυσης.
3. Εξαγωγή του πλέγματος αυτού σε κατάλληλο/α αρχείο/α για περαιτέρω επεξεργασία.
4. Επίκληση κάποιου επιπλέον λογισμικού όπου κρίνεται απαραίτητο (π.χ. λογισμικό αριθμητικής επίλυσης εξισώσεων Navier – Stokes)

Σημειώνεται επίσης ότι βασικός στόχος είναι η εφαρμογή των ανωτέρω σε διαδικασία βελτιστοποίησης μορφής σώματος στην ρευστοδυναμική, αεροδυναμική κ.λπ. που αποτελεί και αντικείμενο ερευνών του ΕΘΣ. Για τον λόγο αυτό, το βήμα 3

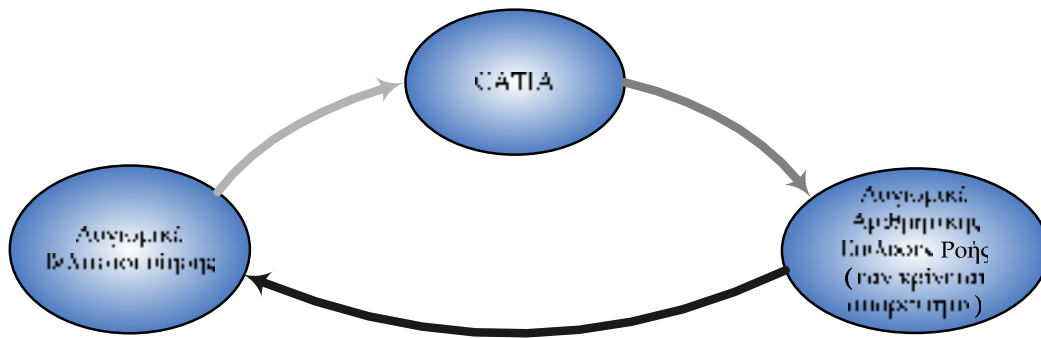
περιλαμβάνει και μορφοποίηση των δεδομένων του πλέγματος έτσι ώστε να είναι συμβατά με τον τρόπο με τον οποίο δέχεται δεδομένα ο κώδικας επίλυσης των εξισώσεων ροής του ΕΘΣ.

Φυσικά, όπως και πριν, έτσι και μετά την συμπλήρωση της συνολικής διαδικασίας, η εκτέλεσή της πρέπει να γίνεται χωρίς την παρέμβαση χρήστη. Αυτό συνεπάγεται ότι και οι λειτουργίες του CATIA θα πρέπει να εκτελούνται αυτόματα την κατάλληλη στιγμή σε κάθε επανάληψη.

Ακολουθούν τα σχήματα 5.1 και 5.2 τα οποία δείχνουν σε μορφή εικόνων και διαγραμμάτων την συνολική διαδικασία και τον ρόλο - λειτουργία του CATIA.



Σχήμα 4.1: Λειτουργία προσθήκης CATIA στη βελτιστοποίηση



Σχήμα 4.2: Διαδικασία Βελτιστοποίησης – θέση CATIA επί του συνόλου

4.2 Υλοποίηση και Αυτοματοποίηση της Διαδικασίας

Έχει γίνει κατανοητό μέχρι στιγμής το τι ακριβώς απαιτείται να πραγματοποιηθεί μέσω του λογισμικού CATIA για να επιτευχθεί ο αρχικός στόχος. Στη συνέχεια παρουσιάζεται το πώς υλοποιήθηκαν αυτές οι απαιτήσεις μέσω εντολών του CATIA και γλωσσών προγραμματισμού.

Αρχικά, αναφέρεται η δυνατότητα, που παρέχεται από το συγκεκριμένο λογισμικό CAD, πλήρους παραμετροποίησης της οποιασδήποτε γεωμετρίας σχεδιαστεί σε αυτό, μέσω ενός αρχείου τύπου *.txt ή *.xls. Το αρχείο αυτό χαρακτηρίζεται από το ίδιο το πρόγραμμα ως *design table* και εμπεριέχει, σε μορφή κειμένου, όλες από τις μεταβλητές που περιγράφουν τη γεωμετρία επιθυμεί ο χρήστης. Η αξία του design table είναι η αυτόματη προσαρμογή της γεωμετρίας στις τιμές των μεταβλητών που εμπεριέχονται εντός του αντίστοιχου αρχείου [Cat00]. Με τη βοήθεια αυτού, δηλαδή, είναι εφικτή η αναβάθμιση της γεωμετρίας βάσει του συνδυασμού των μεταβλητών που ορίζει ο εξελικτικός αλγόριθμος σε κάθε επανάληψή του, γεγονός μέγιστης σημασίας για την διευκόλυνση της συνολικής διαδικασίας βελτιστοποίησης μέσω του CATIA.

Η ύπαρξη του design table, προϋποθέτει την εκ των προτέρων δημιουργία της γεωμετρίας στα κατάλληλα αρχεία, με τυχαίες τιμές μεταβλητών οι οποίες εν τέλει θα αποθηκευτούν στο αρχείο *.txt (ή *.xls). Από το σημείο αυτό και μετά, οι τιμές μπορούν να αλλαχθούν μόνο μέσω αυτού του αρχείου και είναι απευθείας συνδεδεμένες με το αρχείο της γεωμετρίας το οποίο και αναπροσαρμόζουν κατάλληλα.

PartBody\Sketch.1\Radius.6\Radius (mm)	100.703458540977
PartBody\Sketch.1\Radius.8\Radius (mm)	50.57410534555
PartBody\Sketch.1\Angle.9\Angle (deg)	65.3571571388087
PartBody\Translate.1\Length (mm)	342.406656455511
PartBody\Sketch.2\Offset.13\Offset (mm)	72.5701401794616
PartBody\Sketch.2\Radius.14\Radius (mm)	50.1695744019996
PartBody\Sketch.2\Radius.16\Radius (mm)	40.00792369184713
PartBody\Sketch.2\Angle.17\Angle (deg)	71.5740109858059
PartBody\Translate.2\Length (mm)	305.261898761271
PartBody\Fillet.1\Radius (mm)	5.62862279239713

Σχήμα 4.3: Ενδεικτική μορφή design table αρχείου

Εάν παραδείγματος χάριν, το αρχικό μήκος ενός άξονα κατά τη σχεδίασή του ορίστηκε ίσο με 1 m και συμπεριλήφθηκε στα περιεχόμενα του design table, πιθανή αλλαγή της τιμής αυτής σε 2 m, με το άνοιγμα του αρχείου της γεωμετρίας από το CATIA, θα προκαλέσει αυτόματη ανακατασκευή του άξονα με την ίδια διάμετρο (και λοιπές παραμέτρους) αλλά με μήκος 2 m.

4.2.1 Γεωμετρία

Με βάση τα όσα προηγήθηκαν, η δημιουργία της γεωμετρίας και η κατάλληλη παραμετροποίηση και αποθήκευσή της είναι το πρώτο βήμα της υλοποίησης. Το CATIA, όπως έχει προαναφερθεί, περιέχει πολλά διαφορετικά περιβάλλοντα, καθένα από τα οποία μπορεί να συντελέσει στην κατασκευή της γεωμετρίας. Ωστόσο, επειδή παίζει σημαντικό ρόλο η μορφή του αρχείου αποθήκευσης της γεωμετρίας για τη λειτουργικότητα της διαδικασίας, τα τμήματα του λογισμικού που θα χρησιμοποιηθούν θα είναι συγκεκριμένα, μιας και αυτά είναι που καθορίζουν τον τύπο του αρχείου αυτού.

Σε πρώτο στάδιο, αναφερόμαστε σε γεωμετρίες πρωτογενών τεμαχίων (ο όρος πρωτογενές αναφέρεται σε τεμάχια που δεν μπορούν να αναλυθούν σε μικρότερα κομμάτια). Για το λόγο αυτό, η κατάληξη του αρχείου αποθήκευσης θα είναι ***.CATPart** - σε αντίθεση με τα αρχεία που περιέχουν σύμπλεγμα τεμαχίων (assemblies), των οποίων η κατάληξη είναι *.CATProduct, ή μηχανολογικό σχέδιο (*.CATDrawing) - [Cat00]. Μέσω των εντολών *Start > Mechanical Design* και *Start > Shape* παρέχεται πρόσβαση σε πληθώρα εργαλείων κατάλληλων για την δημιουργία οποιασδήποτε γεωμετρίας της προκαθορισμένης μορφής. Από το σύνολο αυτών προτείνονται τα:

- Part Design

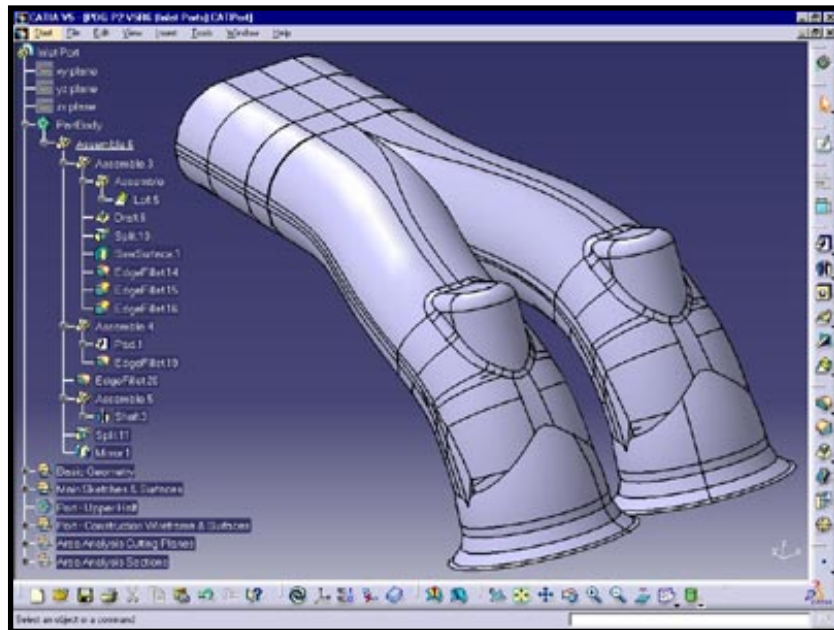
- Generative Shape Design

διότι εμπεριέχουν τα πλέον πιο χρήσιμα εργαλεία, χωρίς αυτό να σημαίνει ότι η εκμετάλλευση και άλλων τμημάτων (όπως το Sheet Metal Design κ.α.) δεν είναι επιτρεπτή.

Σε αυτό το στάδιο είναι πολύ σημαντικό το να ληφθεί υπόψη η μορφή του μη-δομημένου πλέγματος που θα δημιουργηθεί στη συνέχεια. Ο τρόπος με τον οποίο θα κατασκευαστεί η γεωμετρία επηρεάζει τη μορφή του πλέγματος με διάφορους τρόπους. Με άλλα λόγια, το δέντρο (βλέπε παράγραφο 3.2) το οποίο θα προκύψει από την κατασκευή της γεωμετρίας είναι άμεσα συνδεδεμένο με αυτό της ανάλυσης και του πλέγματος γιατί τα τμήματα του δεύτερου θα προκύπτουν μετά από επιλογές τμημάτων του πρώτου. Για παράδειγμα, δεν είναι εφικτός ο ορισμός επιφανειακού πλέγματος, εάν στη γεωμετρία υπάρχουν μόνο γραμμές και όχι επιφάνειες.

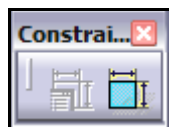
Μέσα από τις γραμμές εργαλείων και την χρήση εντολών τους εντός του λογισμικού επιτυγχάνεται η σχεδίαση ενός συνόλου σημείων, γραμμών, επιφανειών και όγκων, τα οποία εν τέλει σχηματίζουν το αντικείμενο που θα δοθεί προς βελτιστοποίηση.

Σχήμα 4.4:
Διακλάδωση
σωληνώσεων
σχεδιασμένη στο
CATIA

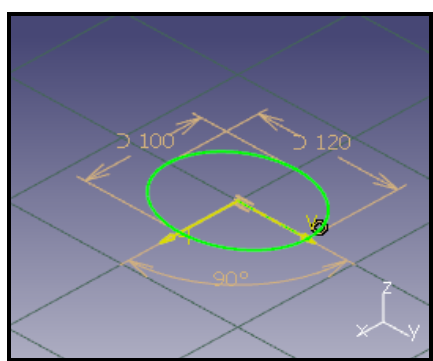


Δεν αρκεί όμως μόνο η σχεδίαση. Όπως προαναφέρθηκε, πρέπει να γίνει πλήρης παραμετροποίηση της γεωμετρίας για τη δημιουργία του design table. Ορισμένες από τις μεταβλητές που περιγράφουν τη γεωμετρία και θα χρησιμοποιηθούν ως παράμετροι στη συνέχεια έχουν δημιουργηθεί κατά τη διάρκεια

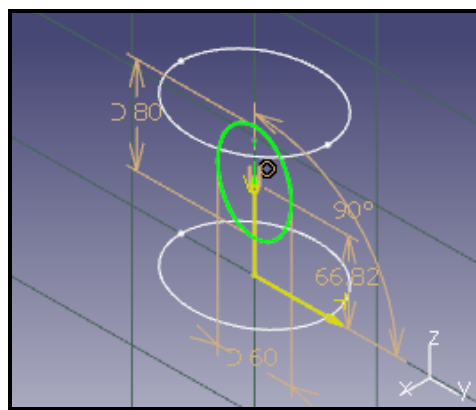
της σχεδίασης. Για όλες τις υπόλοιπες, που δεν περιλαμβάνονται στο δέντρο του σχεδίου αλλά είναι επιθυμητή η ύπαρξή τους, χρησιμοποιείται η εντολή *Constraint* ή *Constraint Defined in dialog Box* που βρίσκεται στη γραμμή εργαλείων *Constraints*.



Με τη χρήση αυτής της εντολής μπορούν να οριστούν διαστάσεις, γωνίες κ.λπ. και οι τιμές τους, είτε αυτές έχουν ενταχθεί ήδη στο σχέδιο είτε όχι. Για λόγους πληρότητας και οργάνωσης, θα γίνει χρήση της εντολής για όλες τις μεταβλητές που περιγράφουν το αντικείμενο. Στη συνέχεια ακολουθεί σχήμα με κάποιες από τις παραμέτρους (constraints) της γεωμετρίας που χρησιμοποιείται στο κεφάλαιο 6 (στην εικόνα παρουσιάζονται με το πορτοκαλί χρώμα) :



(α)



(β)

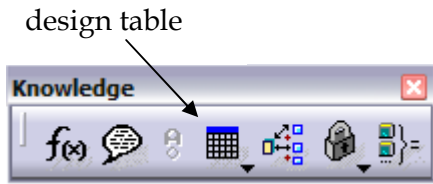
Σχήμα 4.5: Παράμετροι (constraints) γεωμετρίας σωλήνωσης, όπως αυτές φαίνονται εντός του περιβάλλοντος του CATIA. α) ελλειπτική διατομή κατακόρυφου σωλήνα, β) ελλειπτική διατομή οριζόντιου σωλήνα και θέση αυτού ως προς τον κατακόρυφο

Με την ολοκλήρωση των ανωτέρω, το αρχείο αποθηκεύεται με την ονομασία Geometry.CATPart εντός ενός φακέλου, ο οποίος και θα χρησιμοποιηθεί για την αποθήκευση **όλων** των αρχείων που θα προκύψουν από τη διαδικασία αυτή και θα χρησιμοποιούνται κατά τη διάρκεια της βελτιστοποίησης. Η ονομασία, όπως και η κατάληξη, επηρεάζει τη λειτουργικότητα του συνόλου.

4.2.2 Design Table

Το επόμενο βήμα, είναι η δημιουργία του design table. Από τη γραμμή εργαλείων *Knowledge*, με την επιλογή του εικονιδίου που φαίνεται στο σχήμα 4.6, εμφανίζεται το παράθυρο “Creation of design table” του σχήματος 4.7, απ’ όπου

μπορούν να καθοριστούν τα χαρακτηριστικά του αρχείου που απαιτείται. Αυτά τα χαρακτηριστικά επηρεάζουν τη μορφή και τα δεδομένα του αρχείου και έχουν προφανώς σημασία στην μετέπειτα εξέλιξη της διαδικασίας.

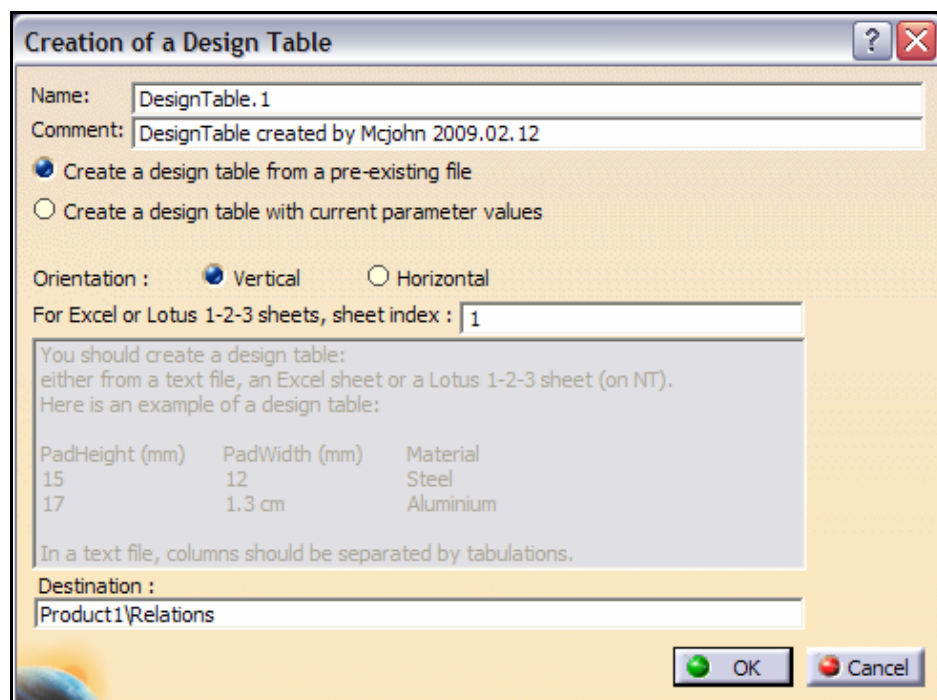


Σχήμα 4.6: Knowledge toolbar

Για τον λόγο αυτό ως τύπος αρχείου επιλέχθηκε το *.txt (έναντι του *.xls) διότι θεωρείται το πιο εύχρηστο από τα δύο ενώ δόθηκε η ονομασία "design" - συνολικά design.txt -.

Πιο συγκεκριμένα, στο παράθυρο που προέκυψε, πρέπει να καταχωρηθούν τα εξής:

1. Η ονομασία "*design*" στην γραμμή εισαγωγής δεδομένων "Name".
2. Η επιλογή "*Create a design table with current parameter values*" η οποία επιτρέπει, στη συνέχεια, επιλογή μεταβλητών από το σύνολο αυτών που έχουν ενταχθεί στο σχέδιο της γεωμετρίας.
3. Η επιλογή "*Horizontal*" που καθορίζει την θέση των τιμών σε σχέση με τις ονομασίες των μεταβλητών εντός του αρχείου. Η συγκεκριμένη επιλογή τοποθετεί τις τιμές δίπλα στις ονομασίες και θεωρείται η πιο ευανάγνωστη και ευκολότερα επεξεργάσιμη αφού οι τελευταίες διαφοροποιούνται σε κάθε επανάληψη.



Σχήμα 4.7: Παράθυρο δημιουργίας Design table

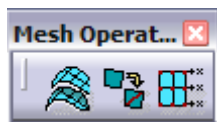
Στη συνέχεια γίνεται εκλογή των παραμέτρων που θα καταχωρηθούν στο design table και θα ελέγχουν, μέσω αυτού, τη μορφή της γεωμετρίας ενώ τέλος αποθηκεύεται το αρχείο στην κατάλληλη τοποθεσία και με την προκαθορισμένη ονομασία. Στο τελευταίο αυτό βήμα, επιλέγεται και ως τύπος αρχείου το “αρχείο κειμένου (text file - *.txt) ”.

Από αυτό το σημείο και έπειτα, το Geometry.CATPart είναι άμεσα συνδεδεμένο με το design.txt, γεγονός που πραγματοποιείται αυτόματα από το CATIA με τη δημιουργία του δευτέρου και που μπορεί να ελεγχθεί ανά πάσα στιγμή με την εντολή *Edit > Links...* [Cat00].

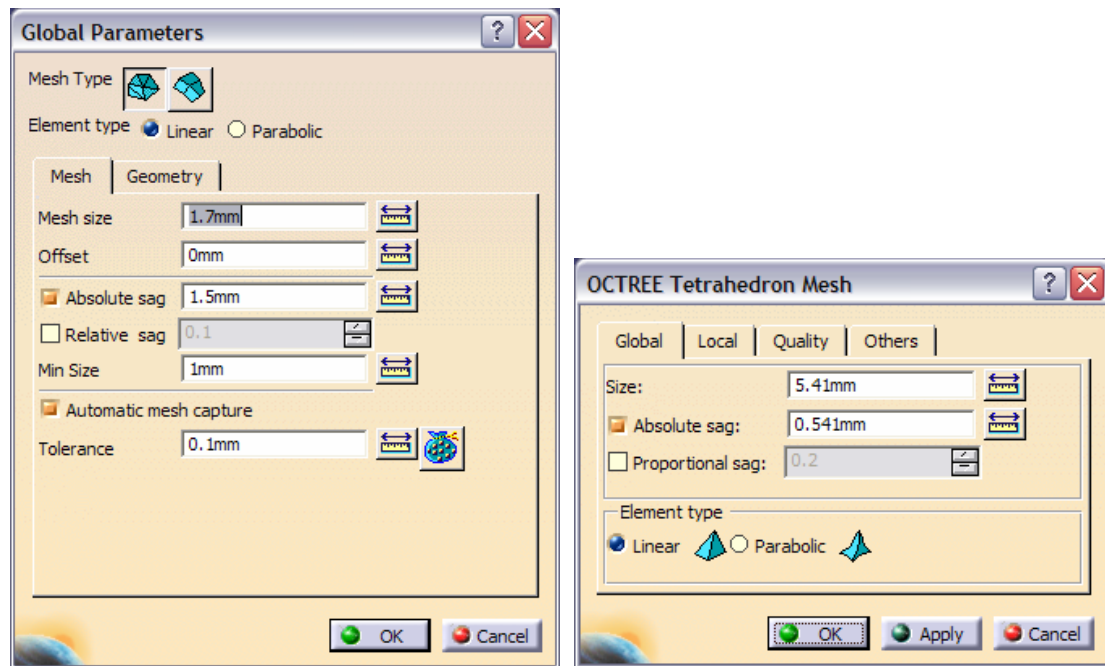
4.2.3 Μη-Δομημένο Πλέγμα

Όπως έχει προαναφερθεί, η προσαρμογή της γεωμετρίας σε ένα συνδυασμό τιμών και η οπτική απεικόνιση αυτής είναι ένα από τα ζητούμενα, καθώς είναι προαπαιτούμενο για τη συνέχεια. Το βασικότερο απ’ όλα όμως, είναι η δημιουργία του μη-δομημένου πλέγματος, στην επίλυση του οποίου στηρίζεται η απόδοση τιμής καταλληλότητας ή κόστους και η επιλογή της(των) βέλτιστης(ων) λύσεων που είναι και ο απώτερος σκοπός.

Εντός του CATIA, είναι εφικτό μέσω του περιβάλλοντος *Advanced Meshing Tools*. Οι αντίστοιχες γραμμές εργαλείων έχουν όλα τα απαραίτητα εργαλεία για την κατασκευή, ανάλυση, μορφοποίηση και εξαγωγή του επιθυμητού πλέγματος. Οι πιο χρήσιμες από αυτές είναι οι γραμμές εργαλείων *Meshing Methods*, *Mesh Operators*, *Mesh Transformations* μέσα από τις οποίες καθορίζονται η μορφή και τα χαρακτηριστικά του πλέγματος [Cat00].



Σημειώνεται όμως ότι τα χαρακτηριστικά του δεν μεταβάλλονται κατά τη διάρκεια της διαδικασίας βελτιστοποίησης. Συνεπώς, σε αυτό το σημείο πρέπει να γίνει προσεκτική κατασκευή του πλέγματος στηριζόμενη στην εμπειρία του χρήστη, το είδος της μαθηματικής επίλυσης που θα ακολουθήσει, τυχόν απαιτήσεις ακριβείας που μπορεί να προκύπτουν από διάφορους παράγοντες κ.α.



(α)

(β)

Σχήμα 4.8: Παράθυρο καθορισμού παραμέτρων πλέγματος που προκύπτει από την επιλογή του εργαλείου α) Advanced Surface Mesher και β) OCTREE Tetrahedron Mesher

Οι παράμετροι που θα επιλεγούν θα ορίζουν τη μορφή του, ασχέτως τιμών μεταβλητών. Σε αυτές τις παραμέτρους περιλαμβάνονται: ο τύπος του πλέγματος, ο τύπος των στοιχείων του πλέγματος (element type), το μέγεθος αυτών (mesh size) κ.α. ενώ ενδεικτικά περιγράφονται στη συνέχεια οι σημαντικότερες από αυτές [Cat00]:

Mesh Type (τύπος πλέγματος): Ορίζει το είδος του πλέγματος αλλά και το σχήμα των στοιχείων που το απαρτίζουν. Από το εργαλείο που θα επιλεχθεί για τη δημιουργία του θα επηρεαστεί η μορφή του (γραμμικό, επιφανειακό, όγκου) ενώ εντός του menu του κάθε εργαλείου ορίζεται το σχήμα των στοιχείων του (τρίγωνο, τετράπλευρο).

Element Type (τύπος στοιχείων): Ορίζει τον αριθμό των κόμβων που έχει κάθε στοιχείο μέσω του χαρακτηρισμού του ως γραμμικό ή παραβολικό.

Mesh Size (μέγεθος στοιχείων): Έχει αριθμητική τιμή και καθορίζει την απόσταση του ενός κόμβου από τον άλλο. Η τιμή αυτή μεταβάλλεται εντός κάποιων ορίων για ορισμένα στοιχεία κατά την βελτίωση ποιότητας του πλέγματος – optimization.

Mesh Sag: Έχει αριθμητική τιμή και προσδιορίζει τη μέγιστη απόκλιση του κάθε στοιχείου από την πραγματική γεωμετρία στην οποία βασίζεται.

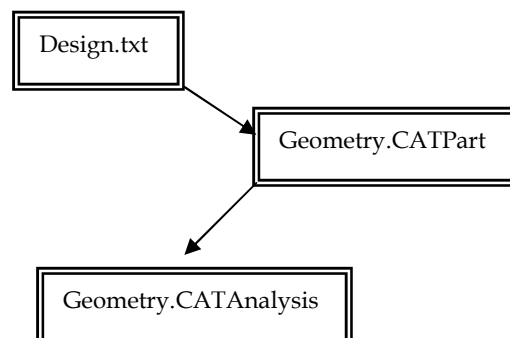
Min Size: Όπως προαναφέρθηκε, κατά το optimization του πλέγματος, μεταβάλλεται το μέγεθος των στοιχείων αυτομάτως από το CATIA. Η παράμετρος αυτή ορίζει την ελάχιστη επιτρεπτή αριθμητική τιμή μεγέθους που μπορεί να προκύψει κατά την διαδικασία αυτή.

Automatic Mesh Capture Tolerance: Έχει ομοίως αριθμητική τιμή και προσδιορίζει την ανοχή κατά την ένωση δύο πλεγμάτων σε συγκεκριμένες περιοχές έτσι ώστε οι κοινοί κόμβοι τους να θεωρηθούν ως ένας από το λογισμικό

Έχοντας “ανοιχτό” το Geometry.CATPart, και εκτελώντας την εντολή *Start > Analysis & Simulation > Advanced Meshing Tools* εισάγεται αυτομάτως η γεωμετρία στο επιθυμητό περιβάλλον του CATIA, για περαιτέρω επεξεργασία με τα εργαλεία κατασκευής πλέγματος. Αποτέλεσμα της χρήση αυτών και του καθορισμού όλων των απαραίτητων στοιχείων που περιγράφονται προηγουμένως είναι το επιθυμητό πλέγμα το οποίο αποτελεί ξεχωριστή οντότητα για το λογισμικό σχεδιασμού. Αυτό σημαίνει ότι πρέπει να αποθηκευτεί σε ξεχωριστό αρχείο, κατάλληλο να καταχωρήσει όλα τα δεδομένα για την απεικόνιση και την περιγραφή του. Το αρχείο αυτό έχει κατάληξη *.CATAnalysis και για ευνότητους λόγους ονομάστηκε Geometry.CATAnalysis.

Όπως και τα προηγούμενα, έτσι και αυτό συνδέεται αυτομάτως με τα υπόλοιπα. Έτσι, το Geometry.CATAnalysis προσαρμόζεται στο Geometry.CATPart με το οποίο είναι άμεσα συνδεδεμένο ενώ το τελευταίο στο design.txt. Τελικά, το αποτέλεσμα είναι η αυτόματη προσαρμογή του μη-δομημένου πλέγματος της γεωμετρίας στον συνδυασμό των τιμών των μεταβλητών που περιγράφονται στο design table.

Σχήμα 4.9: Αλληλοεξάρτηση αρχείων



Επειδή όμως το Geometry.CATAnalysis είναι αναγνώσιμο μόνο από το CATIA, κάτι που δυσχεραίνει την επιπλέον ανάλυση και επεξεργασία των δεδομένων του, κρίνεται αναγκαία η εξαγωγή των στοιχείων του πλέγματος σε ένα αρχείο κατάλληλης μορφής. Η δυνατότητα αυτή προσφέρεται απ' ευθείας, από το ίδιο το λογισμικό, μέσω της εντολής *Mesh Export* της γραμμής εργαλείων *Import/Export* [Cat00].

Οι πιθανές μορφές του είναι bulk data αρχείο (*.dat) ή CATIA V4 (*.model) από τις οποίες επιλέγεται η πρώτη διότι ξεπερνάει τον περιορισμό που προαναφέρθηκε, αφού πρόκειται για απλό αρχείο κειμένου. Η ονομασία που του ανατέθηκε είναι ExportedMesh_Temp.dat και η εσωτερική δομή του είναι αυστηρώς καθορισμένη - αρχικά αναγράφεται το σύνολο των κόμβων του πλέγματος με τις συντεταγμένες του καθενός και στη συνέχεια περιγράφονται τα τμήματά του με τη σειρά που έχουν τοποθετηθεί στο τμήμα Nodes and Elements του αντίστοιχου δέντρου - (βλέπε παράγραφο 4.2.5.2). Η τυποποίηση αυτή αποδεικνύεται πολύ χρήσιμη γιατί καθιστά εφικτή την αυτόματη ανάγνωση και τροποποίηση των δεδομένων από κατάλληλους κώδικες.

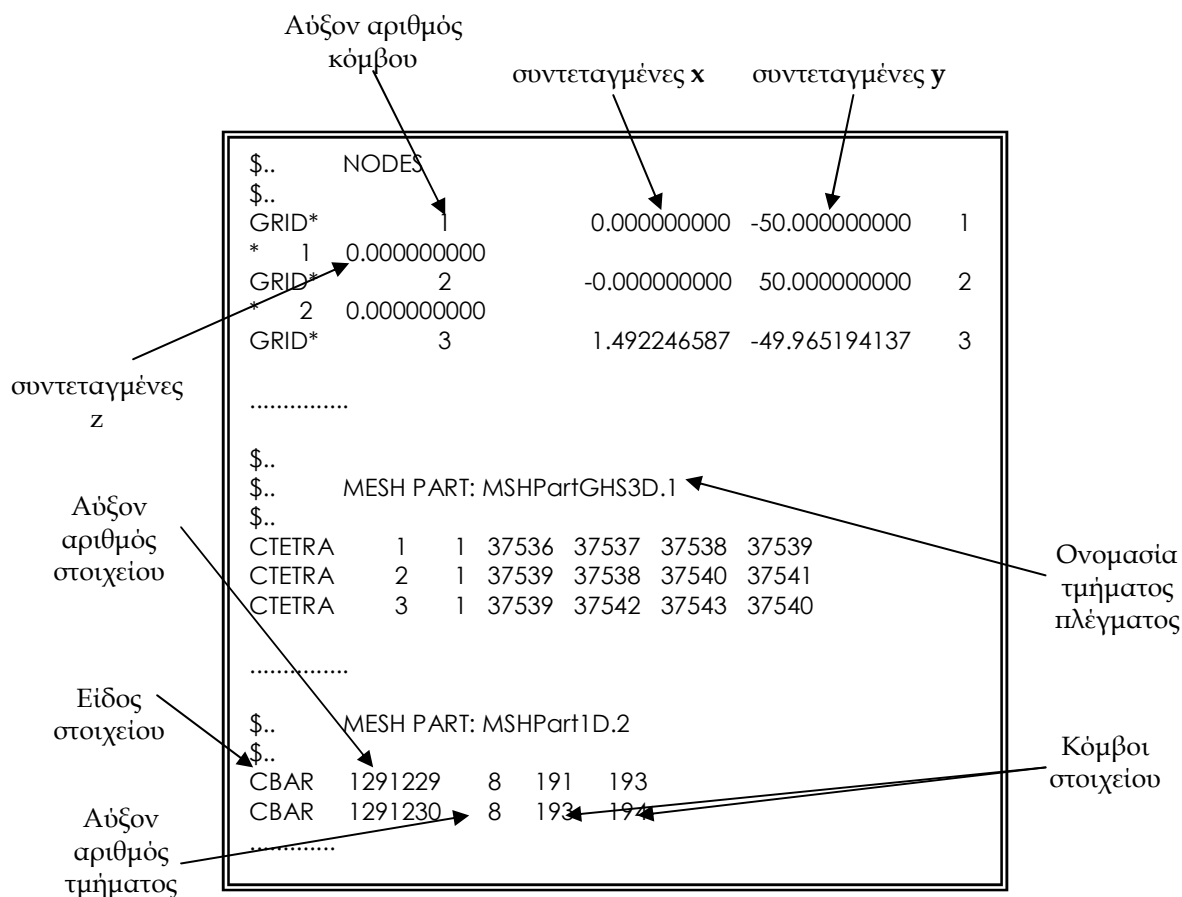
Για λόγους λειτουργικούς αλλά και οργάνωσης, τα τμήματα του πλέγματος που αφορούν όγκους τοποθετούνται στην κορυφή της λίστας Nodes and Elements του δέντρου. Αυτό επιτυγχάνεται αμέσως μετά την δημιουργία του μη-δομημένου πλέγματος με ένα δεξί κλικ του ποντικιού πάνω στην ονομασία Nodes and Elements και επιλογή της εντολής *Reorder Children*, απ' όπου η σειρά εμφάνισης των τμημάτων είναι επεξεργάσιμη [Cat00].

Στο σχήμα 5.10 παρουσιάζεται εικονικά η εσωτερική δομή του συγκεκριμένου αρχείου, για την ευκολότερη κατανόηση των όσων περιγράφονται στις επόμενες παραγράφους αλλά και του τρόπου λειτουργίας του κάθε κώδικα που επεξεργάζεται δεδομένα αυτού του αρχείου.

4.2.4 Macros

Όπως έχει επισημανθεί, η προσθήκη του CATIA στη διαδικασία βελτιστοποίησης είναι η αυτόματη εκτέλεση των επιθυμητών λειτουργιών. Χωρίς το χαρακτηριστικό αυτό, θα ήταν απαραίτητη η επέμβαση του χρήστη σε κάθε επανάληψη, γεγονός μη αποδεκτό για πρακτικούς και λειτουργικούς λόγους.

Μέχρι στιγμής, όσα έχουν περιγραφεί θεωρούνται “χειροκίνητα”. Μπορεί η προσαρμογή του πλέγματος στις τιμές των μεταβλητών να γίνεται αυτόματα, όμως για την επίτευξη αυτού πρέπει πρώτα να ανοιχτεί το αρχείο Geometry.CATAnalysis από το λογισμικό και στη συνέχεια να γίνει αποδεκτός ο συγχρονισμός του με το design.txt από τον χρήστη του λογισμικού. Τέλος, ο ίδιος θα πρέπει να εξαγάγει το πλέγμα πληκτρολογώντας την ονομασία ExportedMesh_Temp.dat στο κατάλληλο παράθυρο αποθήκευσης.



Σχήμα 4.10: Περιγραφή δομής ExportedMesh_Temp.dat αρχείου

Όπως έχει αναφερθεί στην παράγραφο 3.4, το CATIA δίνει τη δυνατότητα να ξεπεραστεί το εμπόδιο αυτό μέσω του Visual Basic Editor. Με καταγραφή των κατάλληλων γραμμών εντολών σε γλώσσα προγραμματισμού Visual Basic [Mic02] εντός του Editor θα παραχθεί μία μακροεντολή της οποίας σκοπός θα είναι η εκτέλεση όλων των ενεργειών που μέχρι στιγμής απαιτούν επέμβαση ανθρώπινου παράγοντα.

Πιο συγκεκριμένα, η αλληλουχία των εντολών που πρέπει να εκτελείται μέσω αυτής είναι η ακόλουθη:

1. Άνοιγμα του αρχείου Geometry.CATPart.
2. Συγχρονισμός του με τις τιμές του design.txt.
3. Εκ νέου αποθήκευσή του.
4. Άνοιγμα του αρχείου Geometry.CATAnalysis.
5. Προσαρμογή του πλέγματος στην νέα γεωμετρία.
6. Αποθήκευσή του αρχείου.
7. Εξαγωγή του πλέγματος με αυτόματη πληκτρολόγηση του ονόματος και του φακέλου τοποθέτησής του.
8. Κλείσιμο των αρχείων και του CATIA.

Τα βήματα 1, 2, 3, 4, 5, 6 και 8 πραγματοποιούνται από την ίδια την μακροεντολή, με την εκτέλεση των κατάλληλων γραμμών κώδικα της γλώσσας προγραμματισμού Visual Basic που καταγράφονται σε αυτή. Οι περισσότερες από αυτές, επειδή αφορούν εξειδικευμένες λειτουργίες του CATIA, είναι εκτελέσιμες μόνο μέσω του συγκεκριμένου Editor και δεν εντάσσονται στο γενικότερο πλαίσιο των εντολών αυτής της γλώσσας. Τμήμα του κώδικα παρουσιάζεται στο σχήμα 5.10.

Όσον αφορά το βήμα 7, υιοθετείται μία διαφορετική προσέγγιση. Αν και η εξαγωγή του πλέγματος είναι εφικτή μέσω αυτού του κώδικα, ωστόσο η αυτόματη πληκτρολόγηση των στοιχείων απαιτεί τη χρήση της εντολής "Sendkeys" της VB [Mic02]. Για λόγους που δεν αναλύονται διότι ξεφεύγουν του αντικείμενου της διπλωματικής, η εκτέλεση της εντολής αυτής γίνεται μέσω ενός νέου αρχείου, τύπου Visual Basic Script. Είναι ένα αρχείο κειμένου με κατάληξη *.vbs του οποίου η εκτέλεση συνεπάγεται την εκτέλεση του κώδικα που είναι καταγεγραμμένος εντός του και το οποίο έχει, επί του παρόντος, ονομαστεί Sendkeys_ExportMesh.vbs.

Συνολικά, κλήση του τελευταίου από τον αρχικό κώδικα, αμέσως μετά την εκτέλεση της εντολής εξαγωγής του πλέγματος (Export Mesh), έχει ως αποτέλεσμα την αυτόματη πληκτρολόγηση της τοποθεσίας και της ονομασίας αποθήκευσης του εξαγόμενου πλέγματος στο κατάλληλο παράθυρο και εν τέλει τη δημιουργία του αρχείου ExportedMesh_temp.dat.

Για να είναι εκτελέσιμος ο βασικός κώδικας της μακροεντολής, πρέπει να είναι αποθηκευμένος ως τμήμα μίας βιβλιοθήκης μακροεντολών, η οποία με τη σειρά της θα πρέπει να έχει ενταχθεί¹ εκ των προτέρων στη λίστα με τις βιβλιοθήκες του CATIA [Cat00]. Το τμήμα του συγκεκριμένου κώδικα ονομάστηκε *TaskCode* ενώ για την βιβλιοθήκη διατηρήθηκε η προεπιλεγμένη ονομασία VBAProject1.catvba

```
.....

'Update the file
part1.Update

Set partDocument1 = CATIA.ActiveDocument

'Save the updated file
partDocument1.Save

Set documents1 = CATIA.Documents

'Open the analysis file
Dim analysisDocument1 As AnalysisDocument
Set analysisDocument1 = documents1.Open("path\Geometry.CATAnalysis")

.....

cmd = "wscript path\sendkeys_ExportMesh.vbs"
Call Shell(cmd)

'#####
'Close documents and CATIA
'#####

analysisDocument1.Close
partDocument1.Close
CATIA.Quit
```

Σχήμα 4.11: Τμήμα της μακροεντολής *TaskCode* της βιβλιοθήκης VBAProject1.catvba

¹ Η ένταξη της VBAProject1.catvba βιβλιοθήκης στην λίστα πραγματοποιείται με την εξής διαδικασία: εκτέλεση της εντολής *Tools > Macro > Macros* εντός του CATIA και επιλογή του κουμπιού *Macros Libraries* στο εμφανιζόμενο παράθυρο. Στο νέο παράθυρο επιλέγεται το κουμπί *Add Existing Library* ενώ η εύρεση του *.catvba αρχείου ολοκληρώνει την διαδικασία.

4.2.5 Συμπληρωματικοί κώδικες

Ο ρόλος του CATIA, αυτός καθαυτός, έχει ολοκληρωθεί και υλοποιηθεί με τα όσα προηγήθηκαν στην παράγραφο 5.2. Η ένταξή του όμως στο σύνολο της βελτιστοποίησης δεν περιορίζεται μόνο στη λειτουργία του CATIA αλλά σχετίζεται και με την ομαλή συνεργασία του με τα υπόλοιπα λογισμικά που συμμετέχουν, τα οποία για το αντικείμενο της συγκεκριμένης διπλωματικής είναι, προφανώς ο εξελικτικός αλγόριθμος, αλλά και το λογισμικό επίλυσης ροής.

Όσον αφορά τον εξελικτικό αλγόριθμο, το CATIA πρέπει να αναγνωρίζει την τρέχουσα γενιά και να προσαρμόζεται στις τιμές των αντίστοιχων μεταβλητών, ενώ μετά το πέρας των ενεργειών που εκτελεί θα πρέπει τα δεδομένα που εξάγει να τροποποιούνται έτσι ώστε να είναι συμβατά με τον τρόπο με τον οποίο τα δέχεται ο αλγόριθμος επίλυσης των εξισώσεων ροής του ΕΘΣ.

Για την επίτευξη αυτών συμπληρώνονται δύο τμήματα κώδικα, γραμμένα σε γλώσσα Visual Basic [Mic02], τα οποία αναλύονται ξεχωριστά παρακάτω. Θεωρήθηκε επαρκής η περιγραφή της λειτουργίας του κάθε κώδικα και της λογικής που τους διέπει ενώ παραλήφθηκε η λεπτομερής ανάλυση των γραμμών εντολών που χρησιμοποιήθηκαν ως πλεονάζουσα για την παρούσα διπλωματική εργασία.

4.2.5.1 Ανανέωση Τιμών των Μεταβλητών

Είναι γνωστό ότι ο εξελικτικός αλγόριθμος που χρησιμοποιείται στο ΕΘΣ για τη διαδικασία βελτιστοποίησης χρησιμοποιεί για την καταγραφή των τιμών των μεταβλητών το αρχείο task.dat. Στην πρώτη γραμμή του αναγράφεται το πλήθος n των μεταβλητών και στις επόμενες n γραμμές οι τρέχουσες τιμές. Οπότε, η ανάγνωση του αρχείου και η αντιγραφή των τιμών στο design table της γεωμετρίας είναι ο σκοπός αυτού του τμήματος κώδικα.

Στο design table όμως που σχεδιάστηκε στην παράγραφο 5.2.2, αναγράφονται όλες οι μεταβλητές που περιγράφουν τη γεωμετρία της εφαρμογής είτε αυτές ανήκουν στις ελεγχόμενες από τον εξελικτικό αλγόριθμο είτε όχι. Επειδή η δημιουργία του πλέγματος από το CATIA είναι αρκετά απαιτητική σε υπολογιστικούς πόρους, έχει μεγάλη σημασία ο σωστός προσδιορισμός **όλων των τιμών** έτσι ώστε να αποφευχθεί η σπατάλη πόρων και να περιοριστεί ο συνολικός χρόνος που απαιτείται για το πέρας της διαδικασίας. Για το λόγο αυτό το συγκεκριμένο τμήμα του κώδικα θα υπολογίζει και τις τιμές παραμέτρων που δεν

καθορίζονται από τον εξελικτικό αλγόριθμο, βάσει μαθηματικών εξισώσεων. Οι μαθηματικές εξισώσεις που θα χρησιμοποιηθούν υπαγορεύονται από την ίδια την εφαρμογή και τις απαιτήσεις αυτής, με γνώμονα τον περιορισμό του μεγέθους του πλέγματος στο ελάχιστο δυνατό, χωρίς όμως την αλλοίωση των αποτελεσμάτων.

Διαδοχικά, ο κώδικας που δημιουργήθηκε, εκτελεί τα ακόλουθα:

1. Ανοίγει το αρχείο task.dat στη μνήμη του υπολογιστή και διαβάζει την αριθμητική τιμή που αναγράφεται σε κάθε σειρά του, αποθηκεύοντάς την ταυτόχρονα σε μία αντίστοιχη μεταβλητή με κατάλληλη ονομασία.
2. Κλείνει το συγκεκριμένο αρχείο διατηρώντας τα επιθυμητά δεδομένα στις μεταβλητές που δημιούργησε προηγουμένως.
3. Προσδιορίζει κάποιους συντελεστές και σταθερές που θα χρησιμοποιηθούν στη συνέχεια.
4. Εάν κριθεί απαραίτητο, υπολογίζει με κατάλληλες μαθηματικές εξισώσεις τις τιμές των μεταβλητών που υπολείπονται για την πλήρη περιγραφή της γεωμετρίας, αποθηκεύοντας και αυτές σε νέες μεταβλητές.
5. Ανοίγει το design.txt στη μνήμη του υπολογιστή και καταγράφει σε κάθε γραμμή αυτού το όνομα της μεταβλητής (όπως αυτό έχει οριστεί από το CATIA κατά την δημιουργία του design table) και ακολούθως, την αριθμητική τιμή της μεταβλητής που αντιστοιχεί σε αυτή την ονομασία, διαχωρίζοντας τα δύο τμήματα της γραμμής με ένα [TAB] (βλέπε σχήμα 4.3).
6. Αποθηκεύει και κλείνει το διαφοροποιημένο πλέον design.txt

```
.....  
  
Input #1, Fillet1  
  
'Close the file  
Close #1  
  
'Definition of constants  
pi = 3.14159265  
  
'Calculate the depended variables  
InnerRadius11 = Radius11 - 10  
InnerRadius12 = Radius12 - 10  
  
.....
```

Σχήμα 4.12: Τμήμα κώδικα VarUpdate

Αυτό το κομμάτι του κώδικα έχει ονομαστεί *VarUpdate* και έχει τοποθετηθεί στην αρχή του συνολικού *TaskCode* αφού αποτελεί το πρώτο βήμα της διαδικασίας.

4.2.5.2 Τροποποίηση Δεδομένων

Για την επίλυση της ροής από το αντίστοιχο λογισμικό του ΕΘΣ απαιτείται η ανάγνωση και αναγνώριση από το τελευταίο, του μη-δομημένου πλέγματος και των χαρακτηριστικών του.

Ο τρόπος με τον οποίο επιτυγχάνεται κάτι τέτοιο είναι προκαθορισμένος και υλοποιείται μέσω δύο αρχείων ASCII, συγκεκριμένου τύπου και εσωτερικής δομής. Τα αρχεία αυτά, πέραν των δεδομένων που ήδη περιέχονται στο *ExportedMesh_Temp.dat*, θα πρέπει να κατηγοριοποιούν και τους κόμβους ανάλογα με την θέση του καθενός στη γεωμετρία, οριοθετώντας έτσι την ροή και καθιστώντας ικανή την επίλυσή της.

Οι κατηγορίες αυτές είναι 5 και περιγράφονται από την τιμή *logfr*. Η ανάθεση μίας τέτοιας τιμής σε έναν κόμβο ορίζει το είδος του κόμβου και προσδιορίζει ορισμένες συνθήκες που επικρατούν στη θέση αυτού. Η αντιστοιχία τιμών και κατηγοριών παρουσιάζεται στον πίνακα 5.1:

Τιμή <i>logfr</i>	Θέση κόμβου
0	Εσωτερικός (In)
1	Periodicity
2	Symmetry
3	Τοίχωμα (Wall)
4	Είσοδος Ροής (Inlet)
5	Έξοδος Ροής (Outlet)

Πίνακας 4.1: Περιγραφή *logfr*

Σε περίπτωση που κάποιος κόμβος ανήκει σε δύο από αυτές τις κατηγορίες, του ανατίθεται διψήφιο *logfr* (π.χ. εάν *logfr* = 34 τότε ο κόμβος αυτός ανήκει στην ακμή του τοιχώματος που βρίσκεται στην είσοδο της ροής στη γεωμετρία).

Οι καταλήξεις των αρχείων αυτών είναι *.nod και *.ele ενώ η ονομασία τους πρέπει να είναι κοινή. Η δομή τους είναι η εξής:

Στο *.nod καταγράφονται:

1η γραμμή: Ο αριθμός των κόμβων του πλέγματος

2η γραμμή: οι τιμές logfr των κόμβων διαχωρισμένες με ένα κενό σε αλληλουχία αντιστοιχία του αύξοντα αριθμού των κόμβων.

3η γραμμή: Οι συντεταγμένες x των κόμβων σε αλληλουχία όμοια με προηγουμένως

4η γραμμή: Οι y συντεταγμένες των κόμβων

5η γραμμή: Οι z συντεταγμένες των κόμβων

Στο *.ele καταγράφονται:

1η γραμμή: πλήθος τετραέδρων, πλήθος πυραμίδων, πλήθος πρισμάτων, πλήθος εξαέδρων

2η γραμμή: 4άδες κόμβων που ανήκουν σε κάθε τετραέδρο. Καταγράφονται η μία μετά την άλλη βάσει του αύξοντα αριθμού των ιδίων των τετραέδρων.

3η γραμμή: 5άδες κόμβων που ανήκουν σε κάθε πυραμίδα.

4η γραμμή: 6άδες κόμβων που ανήκουν σε κάθε πρίσμα

5η γραμμή: 8άδες κόμβων που ανήκουν σε κάθε εξαέδρο

Με σκοπό την μετατροπή των δεδομένων στις ανωτέρω μορφές και την συμπλήρωση αυτών με τα logfr, ο κώδικας εκτελεί βηματικά τα παρακάτω :

1. Εντοπίζει το τμήμα του ExportedMesh_Temp.dat αρχείου στο οποίο περιγράφονται οι κόμβοι και κάνει καταμέτρηση του συνόλου των γραμμών που απαιτούνται για αυτήν την περιγραφή. Έπειτα, παρατηρώντας ότι για την περιγραφή ενός κόμβου απαιτούνται δύο γραμμές, διαιρείται ο αρχικός αριθμός με το 2 για να υπολογιστεί τελικά το συνολικό πλήθος των κόμβων του πλέγματος. Η τιμή αυτή αποθηκεύεται στην μεταβλητή NumNodes.

2. Εντοπίζει, στο ίδιο αρχείο, τα τμήματα του πλέγματος που αναφέρονται σε όγκους και τα διαχωρίζει ανάλογα με το σχήμα των στοιχείων τους. Για κάθε ένα γίνεται καταμέτρηση του συνόλου των γραμμών που απαιτούνται για την περιγραφή του. Μία γραμμή όμως αντιστοιχεί σε ένα και μοναδικό στοιχείο του πλέγματος, συνεπώς από την καταμέτρηση προκύπτει ο αριθμός των στοιχείων ανά είδος. Οι αριθμοί αυτοί αποθηκεύονται στις μεταβλητές NumTet (σύνολο τετραέδρων), NumPyr (σύνολο πυραμίδων), NumPrism (σύνολο πρισμάτων), NumHex (σύνολο εξαέδρων)
3. Δημιουργεί έναν πίνακα διαστάσεων NumNodes x 1 με την ονομασία Nodes, στον οποίο θα αποθηκευτούν οι τιμές logfr. Η αρχικοποίηση του πίνακα τοποθετεί αυτομάτως σε όλα τα κελιά την τιμή "0".
4. Εντοπίζει στο ExportedMesh_Temp.dat τα τμήματα του πλέγματος που αναφέρονται σε γεωμετρίες αντίστοιχες με τα logfr, όπως π.χ. επιφανειακά πλέγματα τοιχωμάτων της γεωμετρίας ή επιφανειακά πλέγματα εισόδων ροής κ.λπ. "Διαβάζει" διαδοχικά τους κόμβους που ανήκουν σε αυτά και αποθηκεύει για κάθε έναν, στο κελί της αντίστοιχης με τον αριθμό του κόμβου γραμμής του πίνακα Nodes , την κατάλληλη τιμή logfr (ανάλογα με το τμήμα του πλέγματος στο οποίο ανήκουν). Φροντίζεται, η διαδικασία αυτή να μην εκτελείται για κόμβους που βρίσκονται στο εσωτερικό της γεωμετρίας, αφού η τιμή logfr που πρέπει να τους ανατεθεί συμπίπτει με την αρχικοποιημένη τιμή των κελιών του πίνακα. Με αυτόν τον τρόπο μειώνεται ο χρόνος εκτέλεσης του κώδικα, αφού οι κόμβοι αυτοί αποτελούν την πλειοψηφία του συνόλου.
5. Συγκρίνει τα πλέγματα που τέμνονται μεταξύ τους στη γεωμετρία, έτσι ώστε να βρεθούν οι κόμβοι που ανήκουν σε περισσότερα από ένα τμήματα και να ανανεωθεί η τιμή logfr σε αυτούς. Ο εντοπισμός πραγματοποιείται με κατάλληλες δομές κώδικα ενώ από το είδος των τμημάτων του πλέγματος που συμμετέχουν στη διαδικασία σύγκρισης προσδιορίζεται η πρόποσα τιμή logfr. Αμέσως μετά την εύρεση κάθε τέτοιου κόμβου, αντικαθίσταται αυτομάτως στον πίνακα Nodes η τιμή του κελιού με αριθμό ίδιο με αυτόν του κόμβου με τη νέα τιμή logfr.

6. Δημιουργεί έναν δεύτερο πίνακα διαστάσεων NumNodes x 3 για την αποθήκευση των συντεταγμένων x, y, z κάθε κόμβου. Προφανώς, ο αριθμός κάθε γραμμής θα αντιστοιχεί στον αριθμό του κόμβου τον οποίο περιγράφουν οι συντεταγμένες που έχουν καταγραφεί στη γραμμή αυτή.
7. Εκτελεί μία επαναληπτική διαδικασία κατά την οποία διαβάζει τις δύο γραμμές του ExportedMesh_temp.dat αρχείου που αναγράφουν τις συντεταγμένες ενός κόμβου, τις αποθηκεύει στις μεταβλητές XCoor, YCoor, ZCoor και έπειτα καταγράφει στα κελιά της κατάλληλης γραμμής του πίνακα Coor τις τιμές αυτών των μεταβλητών (συντεταγμένη x στην πρώτη στήλη, y στην δεύτερη και z στην τρίτη). Η διαδικασία αυτή εκτελείται για όλους τους κόμβους και μάλιστα βάσει του αύξοντα αριθμού αυτών, αφού έτσι είναι καταγεγραμμένοι στο αρχικό αρχείο.
8. Δημιουργεί το αρχείο K.nod και καταγράφει σε αυτό:

1η γραμμή: Την τιμή της μεταβλητής NumNodes

2η γραμμή: Τις τιμές των κελιών του πίνακα Nodes σε σειρά και διαχωρισμένες με ένα κενό.

3η γραμμή: Τις τιμές των κελιών της πρώτης στήλης του πίνακα Coor σε σειρά και διαχωρισμένες με ένα κενό.

4η γραμμή: Ομοίως με προηγουμένως, τις τιμές της δεύτερης στήλης του πίνακα Coor.

5η γραμμή: Τις τιμές της τρίτης στήλης του ίδιου πίνακα.

9. Δημιουργεί το αρχείο K.ele και καταγράφει σε αυτό, εφόσον υπάρχουν στο πλέγμα:

1η γραμμή: Την τιμή των μεταβλητών NumTet, NumPyr, NumPrism, NumHex.

2η γραμμή: 4άδες κόμβων που ανήκουν σε κάθε τετράεδρο. Καταγράφονται η μία μετά την άλλη βάσει του αύξοντα αριθμού των ιδίων των τετραέδρων.

3η γραμμή: 5άδες κόμβων που ανήκουν σε κάθε πυραμίδα.

4η γραμμή: 6άδες κόμβων που ανήκουν σε κάθε πρίσμα

5η γραμμή: 8άδες κόμβων που ανήκουν σε κάθε εξάεδρο

Όσον αφορά τον προσδιορισμό των κόμβων κάθε τρισδιάστατου στοιχείου του πλέγματος, δηλαδή την 2η, 3η, 4η, και 5η γραμμή του K.ele αρχείου σημειώνεται ότι, λόγω της μορφής του ExportedMesh_Temp.dat, συμφέρει να γίνει παράλληλα με την καταγραφή τους ως εξής: “διαβάζεται” από τον κώδικα η γραμμή που περιγράφει ένα στοιχείο και αποθηκεύεται ο αριθμός κάθε κόμβου που ανήκει σε αυτό σε κατάλληλη μεταβλητή. Αμέσως μετά καταγράφονται στην αντίστοιχη γραμμή του K.ele οι τιμές αυτών των μεταβλητών. Επαναλαμβάνοντας την ίδια διαδικασία για όλα τα στοιχεία προκύπτει η επιθυμητή μορφή των γραμμών.

```

.....

If text1 = "$.." MESH PART: MSHPartGHS3D.6" Then
  Input #1, Text2
  C = Input(6, #1)
  Do Until C <> "CTETRA"
    Input #1, D, E, F, G, H, I
    If C = "CTETRA" Then
      k = k + 1
    End If
    C = Input(6, #1)
  Loop
End If
Loop
NumTet = k

Close #1

'#####
'Logfr assignment
'#####

'Create array named Nodes() with the values of logfr for each node
Dim Nodes() As Integer
ReDim Nodes(1 To NumNodes)

.....

```

Σχήμα 4.13: Τμήμα κώδικα Processing_of_Data

Το κομμάτι αυτό του κώδικα ονομάστηκε *Processing_of_Data* και έχει τοποθετηθεί στο τέλος του συνολικού κώδικα TaskCode αφού επιτελεί τις λειτουργίες που απαιτούνται στο τέλος της συνολικής διαδικασίας.

4.2.6 Ολοκλήρωση αυτοματοποίησης

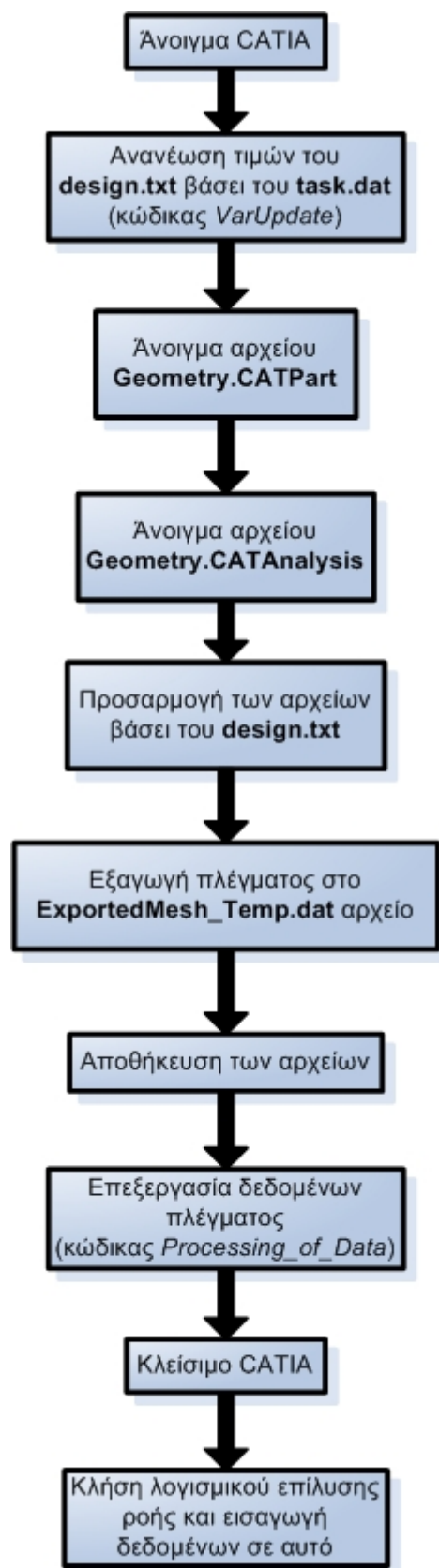
Υπάρχει ένα ακόμα βήμα, με το οποίο ολοκληρώνεται η αυτοματοποίηση της όλης διαδικασίας: η χρήση εντολής DOS (βλέπε παράγραφο 3.5) με κατάλληλες παραμέτρους. Η μέθοδος αυτή επιτρέπει την αποφυγή χειροκίνητης εκτέλεσης του CATIA και στη συνέχεια των επιθυμητών μακροεντολών. Αντ' αυτού, παρέχει την δυνατότητα διαδοχικής εκτέλεσης αυτών με την ενεργοποίηση μίας και μόνο γραμμής κειμένου DOS.

Η εντολή αυτή καταγράφεται σε ένα αρχείο τύπου *.bat [Mic01], η εκτέλεση του οποίου αντιστοιχεί με την πληκτρολόγηση και εκτέλεση της ίδιας εντολής στο περιβάλλον Command Prompt. Το αρχείο αυτό ονομάστηκε task.bat και το κείμενο που εμπεριέχει είναι το εξής [Cat00]:

```
installation path\CNEXT.exe -macro  
path\VBAProject1.catvba TaskCode -batch
```

με το οποίο ενεργοποιείται το λογισμικό CATIA και ακολούθως εκτελείται η μακροεντολή TaskCode της βιβλιοθήκης μακροεντολών VBAProject1.catvba χωρίς ενεργό παραθυρικό περιβάλλον – γραφική απεικόνιση.

Συνολικά, με την εκτέλεση ενός και μόνο αρχείου εκτελούνται όλα όσα έχουν περιγραφεί προηγουμένως στο παρόν κεφάλαιο και εμπεριέχονται στον κώδικα της μακροεντολής TaskCode.



Σχήμα 4.14: Διάγραμμα ροής της διαδικασίας

5. CFD – Επίλυτης Navier – Stokes

Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε ο κώδικας επίλυσης των εξισώσεων Navier - Stokes του Εργαστηρίου Θερμικών Στροβιλομηχανών του Ε.Μ.Π.

Η αριθμητική επίλυση των εξισώσεων ροής γίνεται μέσω μίας μεθόδου χρονοπροέλασης, με την τεχνική των πεπερασμένων όγκων και κεντροκομβική διατύπωση (τα ροϊκά μεγέθη αποθηκεύονται στους κόμβους). Το λογισμικό διαχειρίζεται μη-δομημένα/ υβριδικά πλέγματα (αποτελούμενα από τετράεδρα, πενταεδρικές πυραμίδες, πρίσματα τριγωνικής βάσης και εξάεδρα) ενώ είναι εφικτή η εφαρμογή του και σε δομημένα πλέγματα τα οποία αντιμετωπίζονται ως υβριδικά αποτελούμενα αποκλειστικά από εξάεδρα. Ο υπολογισμός των ατρίβων όρων γίνεται κάνοντας χρήση του σχήματος του Roe [Roe00], ενώ αύξηση της ακρίβειας του σχήματος συνοδεύεται με χρήση συναρτήσεων περιορισμού. Για τη μοντελοποίηση της τύρβης χρησιμοποιούνται μοντέλα μιας ή δύο διαφορικών εξισώσεων, για παράδειγμα Spalart-Allmaras, k-e, k-ω. Ο κώδικας εκτελείται σε πολυεπεξεργαστικό περιβάλλον με την τεχνική των πολλαπλών υποχωρίων (μοντέλο συντονιστή - εργάτη) χρησιμοποιώντας τα πρωτόκολλα επικοινωνίας PVM ή MPI.

Ο επίλυτης αυτός καταλαμβάνει, στη συνολική διαδικασία, τη θέση του λογισμικού αριθμητικής επίλυσης όπως φαίνεται και από το σχήμα 4.2. Αναφέρεται ότι για την εφαρμογή που ακολουθεί (παράγραφος 6.6.2)επιλύθηκαν μόνιμες ροές ενώ χρησιμοποιήθηκε το μοντέλο του SA (μοντέλο μίας εξίσωσης) για την μοντελοποίηση της τύρβης και η συνάρτηση περιορισμού των Van-Leer και Albada για την αύξηση της ακρίβειας κατά τον υπολογισμό των συνεκτικών όρων.

6. Πρακτική Εφαρμογή

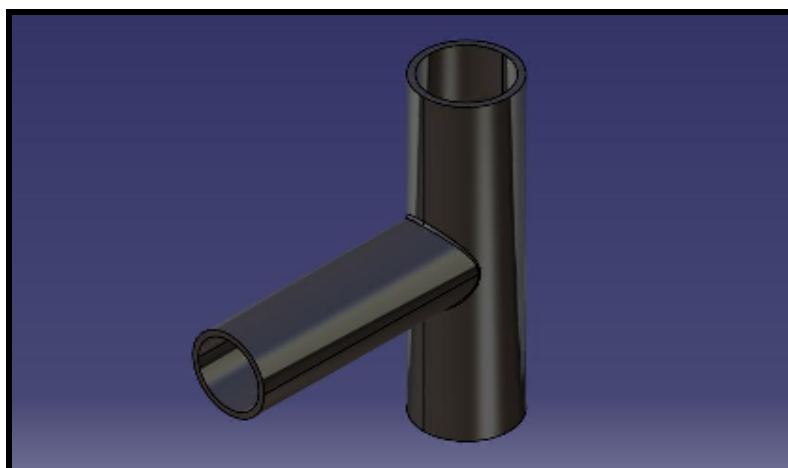
Τα όσα προηγήθηκαν αποτελούν τη θεωρία, τη λογική αλλά και τη βάση για την επίτευξη των στόχων, όπως αυτοί περιγράφονται στο 1^ο Κεφάλαιο (παράγραφος 1.1)

Προφανώς, θεωρείται αναγκαία η εφαρμογή αυτών σε κάποιο πλήρως παραμετροποιημένο, πρακτικό πρόβλημα έτσι ώστε να εξακριβωθεί η αποτελεσματικότητα και αποδοτικότητα του συνόλου. Ταυτόχρονα, από τα αποτελέσματα της εφαρμογής θα μπορούν να διεξαχθούν και ορισμένα πρώτα συμπεράσματα όσον αφορά τον τρόπο κατασκευής και λειτουργίας του ενώ θα παρουσιαστούν και τυχόν “ αδυναμίες ” οι οποίες με τη σειρά τους θα τροφοδοτήσουν εκ νέου προσπάθειες για την βελτίωσή του.

Στη συνέχεια περιγράφεται η πρακτική αυτή εφαρμογή ακολουθούμενη από τα αποτελέσματα που προέκυψαν...

6.1 Γενική περιγραφή

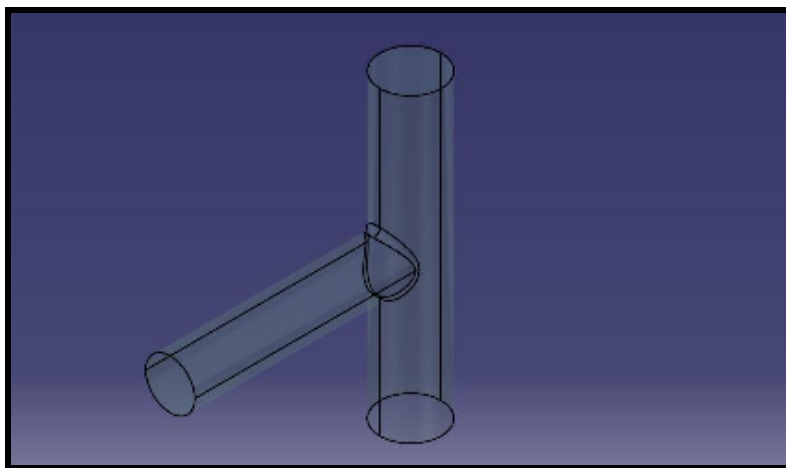
Αφού ο απώτερος σκοπός είναι η εύρεση μίας βέλτιστης γεωμετρίας βάσει κριτηρίων, είναι προφανές ότι οι υποψήφιες επιλογές για την δοκιμή του interface είναι αμέτρητες. Εν τέλει, επιλέχθηκε, η διακλάδωση δύο σωληνώσεων υπό γωνία 90 μοιρών, δηλαδή μία διακλάδωση μορφής T, όπως φαίνεται στο σχήμα 6.1.



Σχήμα 6.1: Σωλήνωση μορφής T

Ως κριτήριο προς βελτιστοποίηση επιλέχθηκε το σύνολο των απωλειών της σωλήνωσης αυτής. Η γεωμετρία με τις ελάχιστες απώλειες, θα αποτελεί και την τελική λύση του προβλήματος.

Το πλέγμα που θα δημιουργείται για κάθε συνδυασμό μεταβλητών πρέπει να αντιστοιχεί στη ροή του ρευστού εντός της σωλήνωσης. Η γεωμετρία, λοιπόν, που θα σχεδιαστεί εντός του CATIA είναι αυτή του όγκου του ρευστού. Από τη μαθηματική επίλυση του πλέγματος της τελευταίας θα προκύπτει σε κάθε επανάληψη και η τιμή αξιολόγησης της συγκεκριμένης υποψήφιας λύσης. Στο εξής, ο όρος “ γεωμετρία ” θα αναφέρεται στον όγκο του ρευστού.



Σχήμα 6.2: Όγκος ρευστού εντός της σωλήνωσης

6.2 Κατασκευή Γεωμετρίας

Καταρχάς, κρίνεται απαραίτητη η παρουσίαση ορισμένων βασικών χαρακτηριστικών του επιλεγμένου γεωμετρικού μοντέλου. Ισχύουν τα εξής:

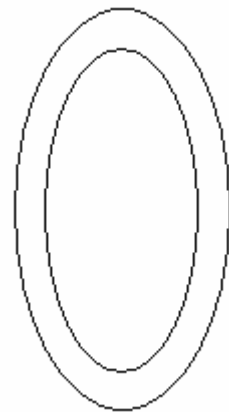
- Οι διατομές των αγωγών θεωρούνται ελλειπτικές
- Οι δύο αγωγοί διακλαδώνονται υπό γωνία 90 μοιρών, γεγονός που θα διατηρηθεί σταθερό καθ' όλη τη διαδικασία
- Ο διαμήκης άξονας του οριζώντιου αγωγού και ο διαμήκης άξονας του κατακόρυφου βρίσκονται πάνω στο επίπεδο xz και είναι κάθετοι στο επίπεδο yz και xy αντίστοιχα

- Υπάρχει μία είσοδος ρευστού και βρίσκεται στο ένα άκρο του κατακόρυφου αγωγού, πάνω στο επίπεδο xy

Πέρα από τα προηγούμενα, όπως αναφέρθηκε και στο κεφάλαιο 4, τη μορφή της γεωμετρίας επηρεάζει και η μορφή του μη-δομημένου πλέγματος που απαιτείται για την επίλυση. Στην προκειμένη περίπτωση, οι εξισώσεις Navier - Stokes επιβάλλουν τον διαχωρισμό της περιοχής που αντιστοιχεί, κατά προσέγγιση, στο οριακό στρώμα της ροής από την υπόλοιπη γεωμετρία, έτσι ώστε το πλέγμα που αντιστοιχεί σε αυτό να είναι αρκετά πυκνό, χωρίς παράλληλα να πυκνώνει τόσο και το υπόλοιπο πλέγμα, γεγονός που θα αύξανε υπερβολικά τις απαιτήσεις σε υπολογιστικούς πόρους και κυρίως σε χρόνο.

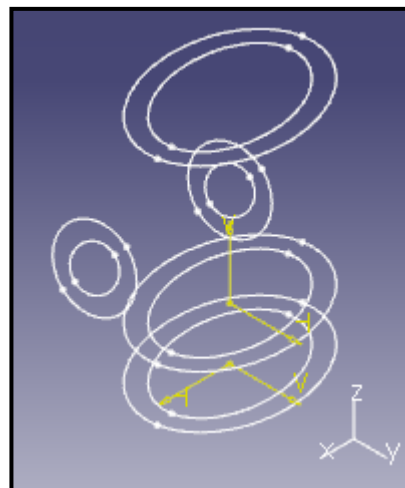
Ο όγκος του πλέγματος θα χωριστεί τελικά σε δύο μέρη: σε ένα δακτυλιοειδή σωλήνα μικρού πάχους ο οποίος εκτείνεται παράλληλα με τα τοιχώματα της σωληνώσεως και σε έναν ελλειψοειδή σωλήνα, ο οποίος καλύπτει το εσωτερικό του προηγούμενου.

Αρχικά δημιουργείται το σχέδιο της διατομής του κατακόρυφου σωλήνα. Η εντολή *Start > Shape > Generative Shape Design* ενεργοποιεί το κατάλληλο περιβάλλον του CATIA [Cat00]. Με την επιλογή του εικονιδίου “Plane” δημιουργείται ένα νέο επίπεδο σχεδιασμού, 30 mm κάτω από το xy και παράλληλο με αυτό. Επιλέγεται το νέο επίπεδο και αμέσως μετά το εικονίδιο “Sketch”, το οποίο περιορίζει τον σχεδιασμό στο προεπιλεγμένο επίπεδο και εμφανίζει νέα εργαλεία. Από αυτά επιλέγεται το εικονίδιο “Ellipse” για την δημιουργία της πρώτης έλλειψης. Με κέντρο το σημείο (0,0,0) ορίζονται οι δύο άξονες της έλλειψης που αποτελεί την εξωτερική ακμή της διατομής του κατακόρυφου σωλήνα. Για τον ορισμό της εσωτερικής ακμής που διαχωρίσει το οριακό στρώμα από τον υπόλοιπο όγκο ρευστού επιλέγεται εκ νέου το επίπεδο και ακολουθείται η ίδια διαδικασία με αποτέλεσμα μία μικρότερη έλλειψη εντός της πρώτης και με παράλληλους άξονες. Απαραίτητη προϋπόθεση για τη σωστή λειτουργία στη συνέχεια είναι η διατήρηση αυτής της παραλληλότητας των αξόνων. Επιλέγοντας τις δύο έλλειψεις, από το παράθυρο σχεδιασμού του CATIA, και στη συνέχεια το εικονίδιο “Constraints defined in a Dialog Box” από τη γραμμή εργαλείων *Constraint* ορίζεται αυτή η παραλληλότητα και εξαναγκάζεται η διατήρησή της στη



συγκεκριμένη γεωμετρία, με αποτέλεσμα, η εσωτερική έλλειψη να περιστρέφεται αυτόματα σε περίπτωση περιστροφής της εξωτερικής.

Η αντιγραφή και μεταφορά αυτών των σχεδίων σε κατάλληλα επίπεδα θα ορίσει τις διατομές εισόδου και εξόδου του κατακόρυφου αγωγού. Επιλέγεται το εικονίδιο *“Translate”* και στη συνέχεια το σχέδιο της εξωτερικής ακμής. Ως άξονας μεταφοράς ορίζεται ο άξονας z ενώ η απόσταση μεταφοράς είναι 30mm έτσι ώστε η είσοδος του αγωγού να συμπίσει με το επίπεδο xy. Εκτελούνται οι ίδιες εντολές για την εσωτερική ακμή και επαναλαμβάνεται η διαδικασία με νέα απόσταση μεταφοράς 100 mm.

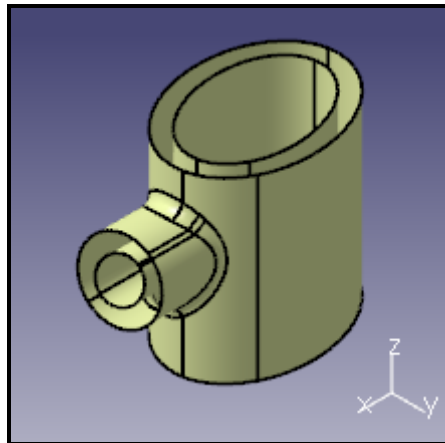


Επειδή και ο οριζόντιος άξονας είναι ίδιας μορφής με τον κατακόρυφο, πρέπει να δημιουργηθούν τα ίδια σχέδια και για αυτόν. Αυτή τη φορά επιλέγεται το επίπεδο yz ενώ κέντρο των ελλείψεων είναι το (0,0,50). Η εντολή *Translate* δημιουργεί μία νέα διατομή σε απόσταση 100 mm από την πρώτη.

Ακολουθεί η δημιουργία των επιφανειών, η οποία θα οριοθετήσει την γεωμετρία του όγκου του ρευστού εντός της σωλήνωσης. Η εντολή *“MultiSections Surface”* σχεδιάζει επιφάνειες μεταξύ δύο γραμμικών ορίων, όπως οι ελλείψεις των διατομών. Επιλέγεται το αντίστοιχο εικονίδιο από την γραμμή εργαλείων *Surfaces* ενώ μετά την εμφάνιση του νέου παραθύρου ορίζονται τα δύο γραμμικά όρια που θα αποτελέσουν τα άκρα της επιφάνειας. Αποτέλεσμα αυτού είναι η εμφάνιση ελλειψοειδών κυλίνδρων που ενώνουν τις εξής δυάδες ελλείψεων:

- Εξωτερική έλλειψη διατομής εισόδου κατακόρυφου αγωγού – εξωτερική έλλειψη διατομής εξόδου κατακόρυφου αγωγού (E1)
- Εσωτερική έλλειψη διατομής εισόδου κατακόρυφου αγωγού – εσωτερική έλλειψη διατομής εξόδου κατακόρυφου αγωγού (E2)
- Εξωτερική έλλειψη διατομής επιπέδου yz οριζόντιου αγωγού – εξωτερική έλλειψη διατομής εξόδου οριζόντιου αγωγού (E3)
- Εσωτερική έλλειψη διατομής επιπέδου yz οριζόντιου αγωγού – εσωτερική έλλειψη διατομής εξόδου οριζόντιου αγωγού (E4)

Σε αυτό το σημείο πρέπει να συμπληρωθεί η καμπυλότητα των τομών αυτών των επιφανειών. Το εικονίδιο *“Fillet”* από την γραμμή εργαλείων *Operations* ορίζει την ακτίνα καμπυλότητας που θα περιγράφει την τομή των προεπιλεγμένων επιφανειών. Η εντολή εκτελείται δύο φορές για τα ζευγάρια επιφανειών E1 – E3 και E2 – E4 με ακτίνες 5 mm και 7 mm αντίστοιχα. Το γεωμετρικό μοντέλο έχει την μορφή που φαίνεται στο σχήμα 6.3...

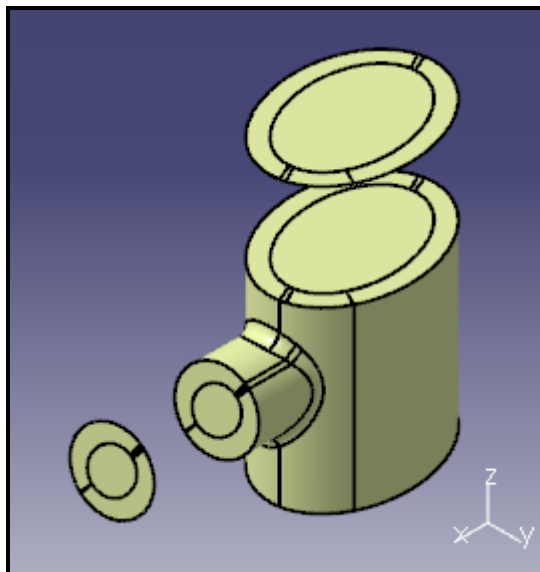


Σχήμα 6.3: Μορφή γεωμετρίας μετά την εκτέλεση των εντολών fillet

Το μόνο που λείπει για την ολοκλήρωση του γεωμετρικού μοντέλου είναι η δημιουργία των επιφανειών στις διατομές εισόδου και εξόδου του ρευστού στη σωλήνωση. Με την εντολή *“Blend”* δημιουργούνται οι ελλειψοειδής δακτύλιοι στα όρια της περιοχής που, λίγο-πολύ, αντιπροσωπεύει το οριακό στρώμα της ροής ενώ με την εντολή *“Fill”* από την γραμμή εργαλείων *Surface* δημιουργούνται οι ελλειπτικές επιφάνειες εισόδου και εξόδου της βασικής ροής [Cat00].

Η κατασκευή της διακλάδωσης των αγωγών έχει ολοκληρωθεί. Το μήκος όμως των σωληνώσεων αυτού του τμήματος θα διατηρηθεί στο ελάχιστο δυνατό, βάσει απαιτήσεων επίλυσης των εξισώσεων ροής, ενώ κατά μήκος του υπόλοιπου τμήματος των αγωγών, το πλέγμα που θα κατασκευαστεί θα αραιώνει σταδιακά. Δημιουργούνται νέες διατομές των δύο αγωγών σε μεγαλύτερη απόσταση από τις προηγούμενες οι οποίες ορίζουν και τις τελικές εξόδους ροής της σωλήνωσης. Τελικά η γεωμετρία πάνω στην οποία θα βασιστεί το μη δομημένο πλέγμα έχει τη μορφή του σχήματος 6.4.

Οι τιμές που έχουν δοθεί μέχρι στιγμής για τον καθορισμό των μεταβλητών είναι τυχαίες και δεν επηρεάζουν σε καμία περίπτωση το σύνολο της διαδικασίας. Δίνουν απλά την επιθυμητή μορφή στο γεωμετρικό μοντέλο. Στη συνέχεια, όμως, θα υπολογίζονται κατά τη διάρκεια της βελτιστοποίησης, είτε από τον εξελικτικό αλγόριθμο είτε από μαθηματικές εξισώσεις βάσει κριτηρίων και θα μεταβάλλονται κατά βούληση, μεταβάλλοντας αντίστοιχα και τη γεωμετρία της σωλήνωσης.



Σχήμα 6.4: Τελική γεωμετρία όγκου ρευστού

6.3 Παραμετροποίηση της γεωμετρίας

Βάσει των όσων έχουν προηγηθεί, η γεωμετρία αυτή ορίζεται πλήρως από 17 μεταβλητές: άξονες ελλείψεως εσωτερικής διατομής κατακόρυφου αγωγού (2), άξονες ελλείψεως εσωτερικής διατομής οριζόντιου αγωγού (2), μήκος βασικού τμήματος κατακόρυφου αγωγού (1), μήκος βασικού τμήματος οριζόντιου αγωγού (1), γωνία του μεγάλου άξονα των ελλείψεων με τον άξονα x στο επίπεδο xy για τον κατακόρυφο αγωγό (1) και με τον άξονα y στο επίπεδο yz για τον οριζόντιο αγωγό (1), θέση οριζόντιου αγωγού κατά μήκος του κατακόρυφου (1), εσωτερική ακτίνα καμπυλότητας της ένωσης των δύο αγωγών (1), άξονες ελλείψεως διαχωρισμού του οριακού στρώματος στον κατακόρυφο αγωγό (2), άξονες ελλείψεως διαχωρισμού του οριακού στρώματος στον οριζόντιο αγωγό (2), ακτίνα καμπυλότητας περιοχής διαχωρισμού του οριακού στρώματος στην διακλάδωση (1), απόσταση τελικής

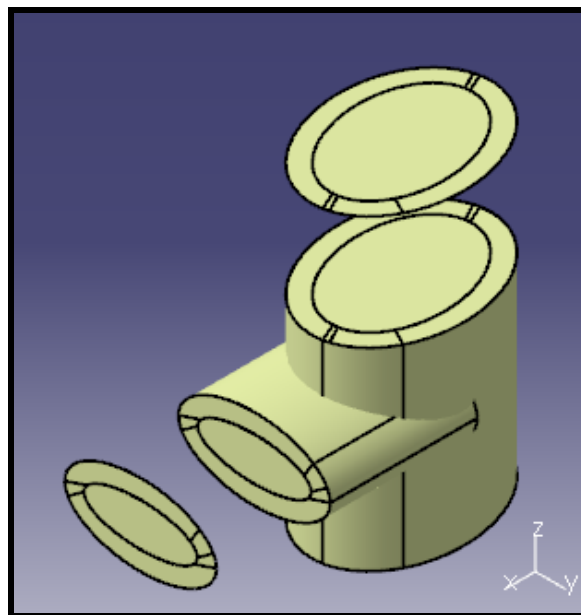
εξόδου οριζόντιου αγωγού από το βασικό τμήμα της διακλάδωσης (1), απόσταση τελικής εξόδου οριζόντιου αγωγού από το βασικό τμήμα της διακλάδωσης (1).

Ο προσδιορισμός αυτών, εντός της γεωμετρίας, γίνεται με την εντολή “Constraints” της γραμμής εργαλείων Constrain [Cat00] η οποία δίνει τη δυνατότητα προσδιορισμού διαστάσεων και λοιπών παραμέτρων, όπως περιγράφεται και στην παράγραφο 4.2.1.

Από αυτές, τα μήκη του βασικού τμήματος της διακλάδωσης αλλά και τα συνολικά μήκη των αγωγών που έχουν σχεδιαστεί θα υπολογίζονται από μαθηματικές εξισώσεις και δεν θα ανήκουν στις παραμέτρους σχεδιασμού του προβλήματος (το τελευταίο είναι προφανές διότι θεωρείται ότι έχουν σχεδιαστεί τμήματα αγωγών και όχι ολόκληρο το μήκος αυτών) - επέκταση του τελευταίου είναι και ο μαθηματικός υπολογισμός της απόστασης του οριζόντιου αγωγού από την είσοδο του κατακόρυφου - . Το ίδιο θα ισχύει και για τους άξονες των ελλείψεων διαχωρισμού του οριακού στρώματος (εσωτερικές ελλείψεις) για κάθε αγωγό, οι οποίοι βάσει εμπειρίας θα υπολογίζονται περί τα 8/10 των αντίστοιχων των εξωτερικών ελλείψεων. Τέλος, η ακτίνα καμπυλότητας διαχωρισμού του οριακού στρώματος στην διακλάδωση θα υπολογίζεται επίσης εμπειρικά.

Συνολικά, καταλήγουμε σε 7 ανεξάρτητες μεταβλητές (βλέπε πίνακα 6.1 σελ. 57) των οποίων τις τιμές θα διαχειρίζεται ο εξελικτικός αλγόριθμος σε κάθε επανάληψη. Η εύρεση του κατάλληλου συνδυασμού των τιμών αυτών αποτελεί και τον στόχο της βελτιστοποίησης.

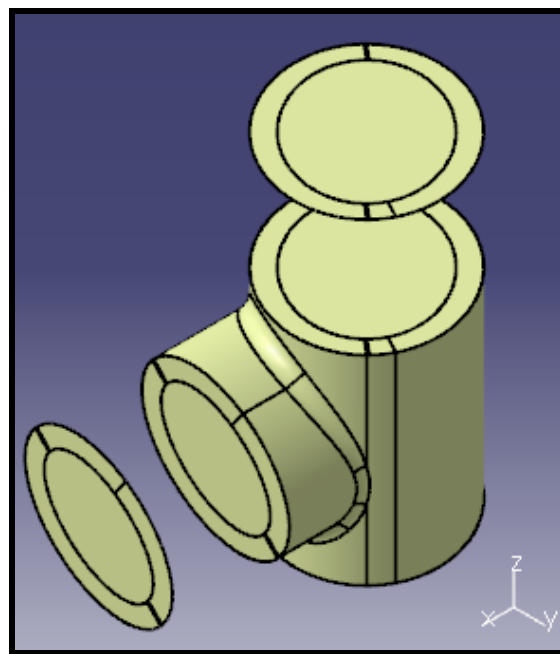
Προφανώς όμως, δεν αρκεί απλά ο προσδιορισμός των ανεξάρτητων παραμέτρων του προβλήματος. Παράλληλα είναι αναγκαία η τοποθέτηση κάθε παραμέτρου εντός συγκεκριμένων αριθμητικών ορίων για να αποφευχθούν απειρισμοί ή μηδενισμοί αυτών κατά την επίλυση αλλά και για να είναι εφικτή η δημιουργία της γεωμετρίας από το CATIA. Περίπτωση παράλειψης ή επιλογής λανθασμένων τιμών των ορίων θα μπορούσε να οδηγήσει π.χ. στη δημιουργία ενός μη επιτρεπτού γεωμετρικού μοντέλου όπως αυτό του σχήματος 6.5 όπου η διατομή του οριζόντιου αγωγού υπερέχει των ορίων του κατακόρυφου.



Σχήμα 6.5: Λανθασμένη γεωμετρία λόγω υπέρβασης επιτρεπτών ορίων

1. Μεγάλος άξονας διατομής ελλείψεως κατακόρυφου αγωγού: Τα όρια της συγκεκριμένης μεταβλητής θα επιλεχθούν σχεδόν αυθαίρετα και θα καθορίσουν σε μεγάλο βαθμό τις τιμές των υπολοίπων, όπως περιγράφεται στη συνέχεια. Μοναδικό κριτήριο περιορισμού των τιμών αυτών αποτελεί η προϋπάρχουσα εμπειρία που σχετίζει τη μεταβλητή με την πολυπλοκότητα της εφαρμογής και τις συνακόλουθες απαιτήσεις σε πόρους. Ο άξονας θα κυμανθεί μεταξύ των τιμών (85mm – 120mm], διαστάσεις που συναντώνται συχνά σε σωληνώσεις.
2. Μικρός άξονας διατομής ελλείψεως κατακόρυφου αγωγού: Προφανώς, δεν υφίσταται λόγος οι τιμές του μικρού άξονα να υπερβαίνουν την ελάχιστη του μεγάλου. Αντιθέτως, όταν οι άξονες αυτοί είναι ίσοι οδηγούμαστε σε κυκλική διατομή. Οπότε καταλήγουμε στα εξής όρια: [60mm – 85mm). Το ελάχιστο όριο επιλέχθηκε ομοίως με την μεταβλητή (1).
3. Μεγάλος άξονας διατομής ελλείψεως οριζόντιου αγωγού: Η μέγιστη τιμή του δεν πρέπει να ξεπερνά την ελάχιστη της μεταβλητής (2) διότι πιθανώς να καταλήγουμε σε μοντέλο όμοιο με αυτό του σχήματος 6.5. Άρα, επιλέγουμε τα όρια (45mm – 60mm].

4. Μικρός άξονας διατομής ελλειψεως οριζόντιου αγωγού: Ισχύει η ίδια λογική με αυτή που αναφέρθηκε κατά τον προσδιορισμό της μεταβλητής (2). Όρια τιμών: [30mm – 45mm).
5. Γωνία κατακόρυφου αγωγού: Είναι η γωνία που σχηματίζεται μεταξύ του μεγάλου άξονα της έλλειψης και του άξονα των x (βλέπε πίνακα 5.1). Λόγω συμμετρίας της διατομής θα κυμανθεί μεταξύ $[(-90)^0 - (+90)^0]$.
6. Γωνία οριζόντιου αγωγού: Είναι η γωνία που σχηματίζεται μεταξύ του μεγάλου άξονα της έλλειψης και του άξονα των y (βλέπε πίνακα 5.1). Τα όρια της είναι όμοια με αυτά της μεταβλητής (5): $[0^0 - 180^0]$.
7. Εσωτερική ακτίνα καμπυλότητας της διακλάδωσης των αγωγών: Η ελάχιστη τιμή της εξαρτάται από την ίδια την κατεργασία της ένωσης, δηλαδή την μέθοδο συγκολλήσεως και βάσει θεωρίας δεν μπορεί να είναι μικρότερη από 2mm [Mak00]. Προφανώς, η μέθοδος συγκόλλησης μπορεί να εξασφαλίσει και μηδενική ακτίνα καμπυλότητας (δηλαδή 90^0 γωνία στην ένωση) περίπτωση που δεν εξετάζεται στην παρούσα εφαρμογή. Όρια τιμών [2mm – 7mm].



Σχήμα 6.6: Ενδεικτική μορφή γεωμετρίας για τιμές μεταβλητών (1) = 110, (2) = 85, (3) = 85, (4) = 60, (5) = 45, (6) = 140, (7) = 7

Όσον αφορά τις υπόλοιπες μεταβλητές, μετά από ανάλυση των αναγκών της επίλυσης αλλά και βάσει εμπειρίας, προέκυψαν τα εξής:

- ✓ Το κάτω άκρο του οριζόντιου αγωγού θα απέχει 30 mm από την διατομή εισόδου της ροής.
- ✓ Η διατομή εξόδου του κατακόρυφου αγωγού θα απέχει 20 mm από το άνω άκρο του οριζόντιου αγωγού.
- ✓ Η διατομή εξόδου του οριζόντιου αγωγού θα απέχει 20 mm από τον κατακόρυφο αγωγό.
- ✓ Η άξονες των εσωτερικών ελλείψεων θα είναι 10 mm μικρότερες από τις αντίστοιχες των εξωτερικών και για τους δύο αγωγούς.
- ✓ Η ακτίνα καμπυλότητας διαχωρισμού του οριακού στρώματος στην περιοχή της διακλάδωσης θα είναι 5 mm μεγαλύτερη από την αντίστοιχη του εξωτερικού ορίου της ροής.

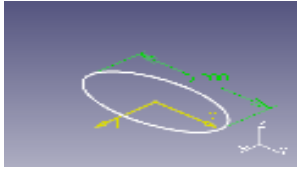
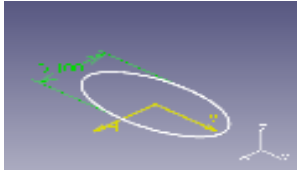
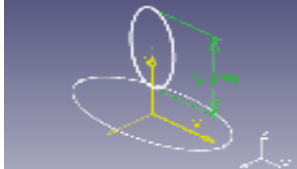
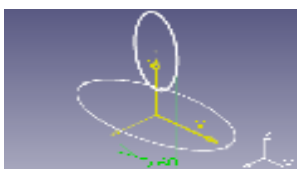
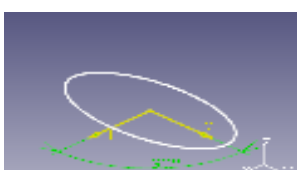
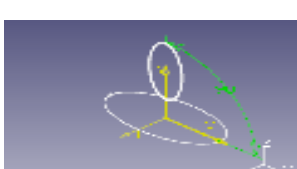
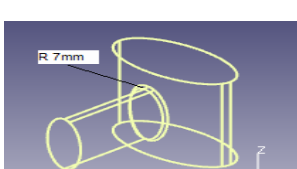
Οι παραπάνω τιμές δεν είναι αυστηρώς καθορισμένες και θα μπορούσαν να διαφοροποιηθούν ποικιλοτρόπως. Ωστόσο, η λογική που διέπει τον προσδιορισμό τους είναι μείζονος σημασίας και θα πρέπει να τηρείται.

Ενδεικτικά αναφέρεται ότι μία σωληνώση αυτής της μορφής, με διαστάσεις και χαρακτηριστικά όπως αυτά που προηγήθηκαν, θα μπορούσε κάλλιστα να αποτελέσει τμήμα σωληνώσεων μίας τυπικής εγκαταστάσεως κλιματισμού ενός γραφείου [Dgt00] ή τμήμα σωληνώσεως παροχής νερού σε δίκτυο πυρόσβεσης μεγάλων εγκαταστάσεων υψηλής επικινδυνότητας.

Στη συνέχεια, γίνεται μία συγκεντρωτική παρουσίαση των ανεξάρτητων παραμέτρων υπό τη μορφή του πίνακα 6.1.

Η επτά μεταβλητές αυτού του πίνακα αλλά και οι υπόλοιπες 10 θα ορίσουν το περιεχόμενο του design table της συγκεκριμένης γεωμετρικής εφαρμογής. Επιλέγεται, εντός του CATIA, το εικονίδιο "*Design Table*" από τη γραμμή εργαλείων Knowledge και, από ένα σύνολο παραμέτρων που εμφανίζεται υπό μορφή λίστας, επιλέγονται και ταξινομούνται οι ανωτέρω 17 μεταβλητές [Cat00]. Επειδή όμως έξι από αυτές έχουν διπλή έκφραση (απαιτούνται δηλαδή 2 τιμές για τον καθορισμό τους στη γεωμετρία λόγω της μορφής της τελευταίας) συνολικά στο design table θα περιλαμβάνονται 23 παράμετροι ακολουθούμενες από τις αντίστοιχες τιμές.

Πίνακας 6.1: Ανεξάρτητες μεταβλητές γεωμετρίας

A/A	Μεταβλητή	Όρια τιμών	Απεικόνιση
1	Μεγάλος άξονας έλλειψης διατομής κατακόρυφου αγωγού	85mm - 120mm	
2	Μικρός άξονας έλλειψης διατομής κατακόρυφου αγωγού	60mm - 85mm	
3	Μεγάλος άξονας έλλειψης διατομής οριζόντιου αγωγού	45mm - 60mm	
4	Μικρός άξονας έλλειψης διατομής οριζόντιου αγωγού	30mm - 45mm	
5	Γωνία κατακόρυφου αγωγού	$(-90)^{\circ} - (+90)^{\circ}$	
6	Γωνία οριζόντιου αγωγού	$0^{\circ} - 180^{\circ}$	
7	Εσωτερική ακτίνα καμπυλότητας διακλάδωσης	2mm - 7mm	

6.4 Δημιουργία Πλέγματος

Έχει ήδη αναφερθεί ότι η επίλυση των εξισώσεων Navier – Stokes επιβάλλει μεγαλύτερη πυκνότητα πλέγματος στην περιοχή του οριακού στρώματος, συγκριτικά με τον υπόλοιπο όγκο της ροής. Το γεγονός αυτό έχει ήδη ληφθεί υπόψη κατά την κατασκευή της γεωμετρίας, με τον διαχωρισμό του “οριακού στρώματος” με ένα σύνολο ακμών και επιφανειών, κατάλληλα τοποθετημένων στα όριά του. Τα μέρη αυτού του συνόλου θα αποτελέσουν και τον συνδετικό κρίκο ανάμεσα στα δύο βασικά μέρη του πλέγματος, αυτό του οριακού στρώματος και αυτό του υπόλοιπου όγκου του ρευστού της ροής.

Για το σωστό προσδιορισμό του μη δομημένου πλέγματος πρέπει για κάθε τμήμα αυτού που θα δημιουργηθεί να είναι ενεργή η επιλογή *Automatic Mesh Capture* η οποία θα εξασφαλίσει την συνέχεια και συνοχή του συνολικού πλέγματος, ασχέτως χαρακτηριστικών επιμέρους τμημάτων [Cat00].

Αρχικά, με την επιλογή του εικονιδίου “*Beam Mesher*” από την γραμμή εργαλείων *Meshing Methods*, δημιουργούνται γραμμικά πλέγματα στις ελλείψεις των διατομών εξόδου ροής του βασικού τμήματος αλλά και στην εξωτερική έλλειψη της διατομής εισόδου ροής. Για τις εξωτερικές ελλείψεις, το μέγεθος των στοιχείων είναι 1,5 mm ενώ για τις εσωτερικές 1,7 mm.

Στη συνέχεια, περνώντας στα επιφανειακά πλέγματα, χρησιμοποιείται το εργαλείο “*Advanced Surface Mesher*” από την ίδια γραμμή εργαλείων. Επαναλαμβανόμενη χρήση αυτού με κατάλληλες επιλογές επιφανειών έχουν ως αποτέλεσμα τη δημιουργία των πλεγμάτων που φαίνονται στον πίνακα 6.2.

Ο πίνακας 6.3 παρουσιάζει συνοπτικά τα υπόλοιπα επιφανειακά πλέγματα που δημιουργήθηκαν μέσω του εργαλείου “*Extrude Mesher with Translation*” το οποίο δίνει τη δυνατότητα μετατροπής ενός γραμμικού πλέγματος σε επιφανειακό με τη δημιουργία πολλαπλών στρωμάτων όμοιων με το πρώτο, κατά μήκος ενός άξονα και για συγκεκριμένη απόσταση, διαμερισμένων είτε βάσει αριθμητικής είτε γεωμετρικής προόδου. Κατά τη δημιουργία αυτού του πλέγματος ορίζεται επίσης και η σχέση μεγέθους των στοιχείων του τελευταίου στρώματος, συγκριτικά με το αρχικό (πολλαπλασιαστής μεγέθους).

Α/Α	Ονομασία	Είδος Στοιχείων		Μέγεθος Στοιχείου ν	Constraint Sag	Ειδικά Χαρακτηριστικά
		Δ	□			
1	Επιφανειακό πλέγμα εξωτερικού ορίου ροής	✓		1,5 mm	0,3 mm	1,5 mm (στην ένωση των δύο αγωγών)
2	Επιφανειακό πλέγμα εσωτερικού ορίου διαχωρισμού του οριακού στρώματος	✓		1,7 mm	0,34 mm	1,5 mm (στην ένωση των δύο αγωγών)
3	Δακτύλιος οριακού στρώματος στη διατομή εισόδου	✓		2 mm	0,4 mm	✗
4	Δακτύλιος οριακού στρώματος διατομής εξόδου βασικού τμήματος κατακόρυφου αγωγού	✓		2 mm	0,4 mm	✗
5	Δακτύλιος οριακού στρώματος διατομής εξόδου βασικού τμήματος οριζώντιου αγωγού	✓		2 mm	0,4 mm	✗
6	Ελλειπτικό τμήμα διατομής εισόδου ροής	✓		4 mm	0,8 mm	✗
7	Ελλειπτικό τμήμα διατομής εξόδου βασικού τμήματος κατακόρυφου αγωγού	✓		5 mm	1 mm	✗
8	Ελλειπτικό τμήμα διατομής εξόδου βασικού τμήματος οριζώντιου αγωγού	✓		5 mm	1 mm	✗
9	Δακτύλιος οριακού στρώματος τελικής διατομής εξόδου κατακόρυφου αγωγού	✓		4 mm	0,8 mm	✗
10	Ελλειπτικό τμήμα τελικής διατομής εξόδου κατακόρυφου αγωγού	✓		10 mm	2 mm	✗
11	Δακτύλιος οριακού στρώματος τελικής διατομής εξόδου οριζώντιου αγωγού	✓		4 mm	0,8 mm	✗
12	Ελλειπτικό τμήμα τελικής διατομής εξόδου οριζώντιου αγωγού	✓		8 mm	1,6 mm	✗

Πίνακας 6.2: Σύνολο επιφανειακών πλεγμάτων παραγόμενων με το εργαλείο Advanced Surface Mesher

Πίνακας 6.3: Σύνολο επιφανειακών πλεγμάτων παραγόμενων με το εργαλείο Extrude Mesher with Translation

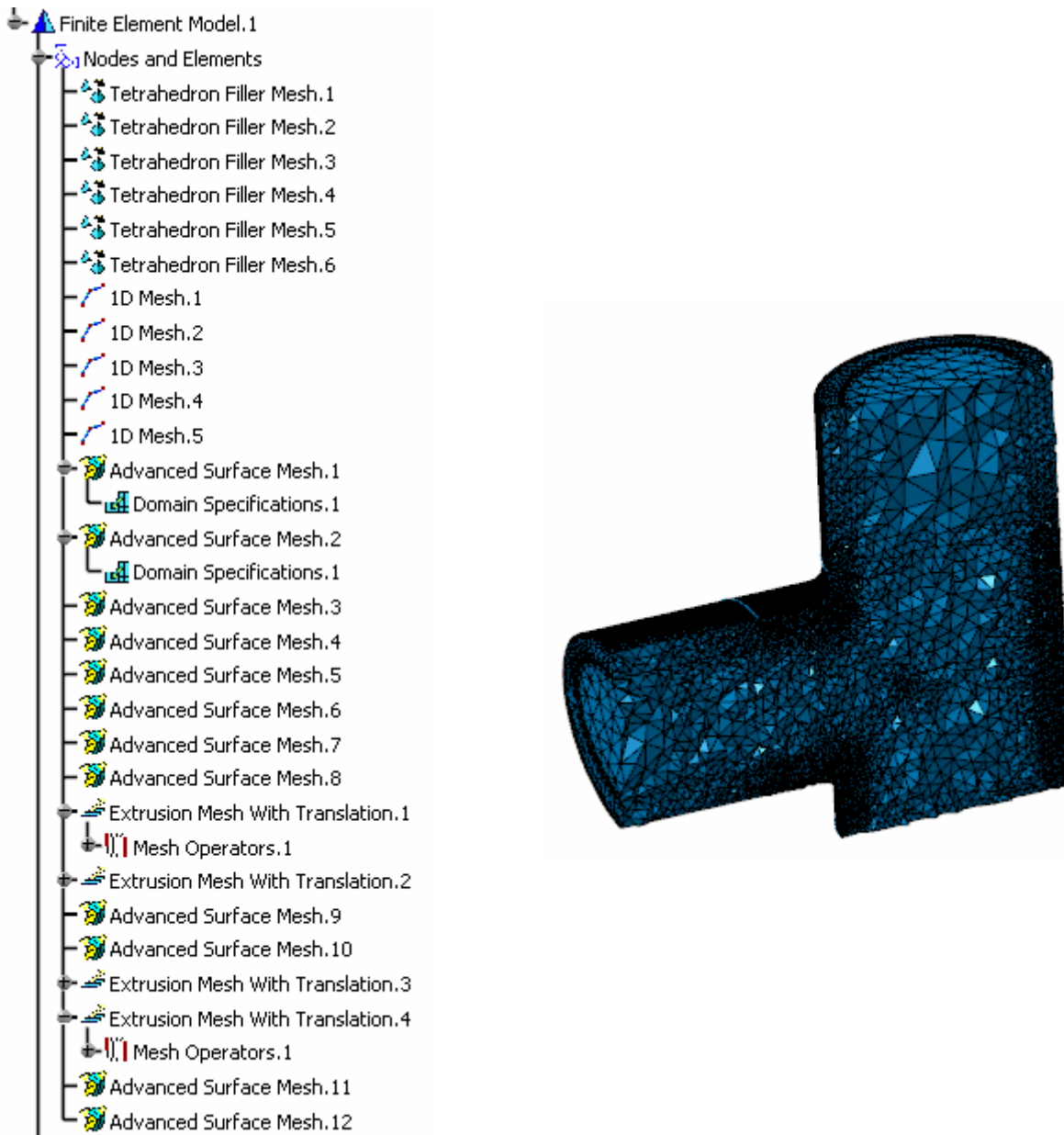
Α/Α	Ονομασία	Είδος Στοιχείων	Πρόσδος Διανομή		Πλήθος στρωμάτων	Πολλαπλασιαστής μεγέθους στοιχείων
			Γεωμετρική	Αριθμητική		
1 3	Επιφανειακό πλέγμα εξωτερικού ορίου ροής μεταξύ βασικής και τελικής εξόδου κατακόρυφου αγωγού	Γραμμικά τρίγωνα	✓		17	4
1 4	Επιφανειακό πλέγμα εσωτερικού διαχωρισμού του οριακού στρώματος μεταξύ βασικής και τελικής εξόδου ροής κατακόρυφου αγωγού	Γραμμικά τρίγωνα	✓		17	4
1 5	Επιφανειακό πλέγμα εξωτερικού ορίου ροής μεταξύ βασικής και τελικής εξόδου οριζόντιου αγωγού	Γραμμικά τρίγωνα	✓		17	4
1 6	Επιφανειακό πλέγμα εσωτερικού διαχωρισμού του οριακού στρώματος μεταξύ βασικής και τελικής εξόδου ροής οριζόντιου αγωγού	Γραμμικά τρίγωνα	✓		17	4

Τα επιφανειακά πλέγματα που δημιουργήθηκαν ήδη, θα χρησιμοποιηθούν για την κατασκευή των πλεγμάτων όγκου με ταυτόχρονη χρήση του εργαλείου “*Tetrahedron Filler*” της γραμμής εργαλείων Mesh Methods. Μέσω αυτής της εντολής επιτυγχάνεται η πλήρωση όγκου, περιβαλλόμενου πλήρως από επιφανειακά πλέγματα, με τετράεδρα [Cat00]. Στην γεωμετρία της συγκεκριμένης πρακτικής εφαρμογής υπάρχουν έξι τέτοιοι όγκοι: οριακό στρώμα στο βασικό τμήμα της διακλάδωσης, κύρια ροή βασικού τμήματος διακλάδωσης, οριακό στρώμα μεταξύ εξόδου ροής του βασικού τμήματος του κατακόρυφου αγωγού και τελικής εξόδου ροής αυτού, κύρια ροή ρευστού στο ίδιο τμήμα και τα αντίστοιχα για τον οριζόντιο αγωγό.

Η πλήρωση του πρώτου γίνεται με συντελεστή μεγέθους 1. Ομοίως και του όγκου των οριακών στρωμάτων μεταξύ βασικών και τελικών εξόδων. Για τον όγκο

της κύριας ροής του βασικού τμήματος ορίστηκε συντελεστής 1,15 ενώ για τα υπόλοιπα συντελεστής 2.

Στο τέλος, έγινε ανακατάταξη της σειράς των πλεγμάτων με την εντολή *Reorder Children* (βλέπε παράγραφο 4.2.3) έτσι ώστε το δέντρο που περιγράφει το συνολικό πλέγμα να έχει την μορφή του επόμενου σχήματος...



Σχήμα 6.7: Δέντρο παραγόμενου πλέγματος και κάθετη τομή αυτού

6.5 Προσαρμογή Τμημάτων Κώδικα “VarUpdate” και “Processing_of_Data” στην Εφαρμογή

Προφανώς, οι γραμμές εντολών που εντάσσονται στα τμήματα VarUpdate και Processing_of_Data του συνολικού κώδικα TaskCode εξαρτώνται από τα δεδομένα της συγκεκριμένης πρακτικής εφαρμογής, όπως το πλήθος των μεταβλητών που περιγράφουν το προς βελτιστοποίηση πρόβλημα, τον υπολογισμό των υπόλοιπων μεταβλητών εάν κρίνεται απαραίτητο και τη μορφή του ExportedMesh_Temp.dat αρχείου, η οποία εξαρτάται από το εκάστοτε πλέγμα.

Αρχικά, απομονώνοντας το τμήμα VarUpdate του συνολικού κώδικα, με κατάλληλη επεξεργασία των γραμμών του, εξασφαλίζεται η ανάγνωση των 7 και μόνο τιμών των μεταβλητών που μας ενδιαφέρουν και περιέχονται στο αρχείο task.dat. Στη συνέχεια προστίθενται στον κώδικα οι μαθηματικές εξισώσεις που απαιτούνται για τον υπολογισμό των υπόλοιπων παραμέτρων της γεωμετρίας και καθορίζονται οι τιμές όσων από τις μεταβλητές θα θεωρηθούν σταθερές. Δεν γίνεται εκτενέστερη αναφορά στις μαθηματικές αυτές λειτουργίες που ορίστηκαν στον κώδικα, αφού δεν αφορούν το αντικείμενο της παρούσας διπλωματικής εργασίας. Τονίζεται μόνο ότι βάσει γεωμετρίας και απλών μαθηματικών πράξεων τηρούνται οι παραδοχές που περιγράφονται στην παράγραφο 6.3 σελ. 56. Τέλος, στην έξοδο του VarUpdate στο design.txt προστίθενται οι ονομασίες όλων των μεταβλητών που περιέχονται σε αυτό ακολουθούμενες από τις τιμές τους.

Αν και το κεντρικό τμήμα του κώδικα TaskCode δεν χρήζει προσαρμογής ανά εφαρμογή, ωστόσο, η τροποποίηση του συνόλου δεν τελειώνει εδώ. Το τελευταίο κομμάτι του (Processing_of_Data) είναι αυτό που απαιτεί την εκτενέστερη επεξεργασία. Από το ExportedMesh_Temp.dat παρατηρείται ο χαρακτηρισμός κάθε τετραέδρου ως “CTETRA”, κάθε τριγώνου “CTRIA3” και κάθε γραμμής “CBAR”. Επίσης, έχει ανατεθεί και μία μοναδική ονομασία σε κάθε τμήμα του πλέγματος: για τα πλέγματα όγκων “MSHPartGHS3D.i”, για τα επιφανειακά πλέγματα “MSHPartSmartSurf.i” και “MSHPartExtrTranslation.i” ενώ για τα γραμμικά “MSHPart1D.i” όπου i ο αύξων αριθμός του τμήματος, όπως αυτός φαίνεται στο δέντρο του σχήματος 6.7.

Χρησιμοποιείται η παράμετρος CTETRA για την καταμέτρηση του συνολικού αριθμού τετραέδρων στο πλέγμα (NumTet). Τοποθετούνται οι ονομασίες των πλεγμάτων στα κατάλληλα σημεία του κώδικα έτσι ώστε να είναι εφικτός ο

εντοπισμός των τμημάτων αυτών εντός του ExportedMesh_Temp.dat και η αυτόματη ανάγνωση των κόμβων που ανήκουν σε κάθε τμήμα. Προσαρμόζονται οι τιμές logfr ανάλογα με το είδος του καθενός (3, 4, 5). Για τον εντοπισμό των κόμβων που έχουν διψήφιο logfr προτιμήθηκε η δημιουργία τεσσάρων βοηθητικών αρχείων, για την καταγραφή των κόμβων των τμημάτων πλέγματος με αύξοντα αριθμό (βάσει του πίνακα 6.1, 6.2) 9 (αρχείο VOutputRingMesh.txt), 11 (αρχείο HOutputRingMesh.txt), 13 (αρχείο Translation1Mesh.txt) και 15 (αρχείο Translation2Mesh.txt). Από διαδοχική σύγκριση των κόμβων στα ζεύγη αρχείων VOutputRingMesh.txt - Translation1Mesh.txt και HOutputRingMesh.txt - Translation2Mesh.txt προκύπτουν οι κόμβοι με logfr = 35. Παράλληλα όλοι οι κόμβοι του γραμμικού πλέγματος MSHPart1D.1 πρέπει να έχουν logfr = 34.

Τέλος, τροποποιείται ο κώδικας δημιουργίας του αρχείου K.ele έτσι ώστε να καταγράφεται σε αυτό μόνο η μεταβλητή NumTet στην πρώτη γραμμή ακολουθούμενη από τα στοιχεία της δεύτερης γραμμής και μόνο.

Σημειώνεται ότι οι ολοκληρωμένοι κώδικες που χρησιμοποιήθηκαν για την υλοποίηση της πρακτικής εφαρμογής του παρόντος κεφαλαίου παρουσιάζονται στα παραρτήματα I, II, III που βρίσκονται στο τέλος του συγγράμματος.

6.6 Δοκιμή Εφαρμογής σε Διαδικασία Βελτιστοποίησης

Η λειτουργικότητα της εφαρμογής που περιγράφεται είναι πολύ εύκολο να ελεγχθεί, απλά και μόνο με την εκτέλεση του αρχείου task.dat (παράγραφος 4.2.6). Προφανώς όμως δεν αρκεί μόνο αυτού του τύπου ο έλεγχος.

Αφού σκοπός είναι η ένταξη του συνόλου σε υπάρχουσα μέθοδο βελτιστοποίησης, θεωρείται απαραίτητη η δοκιμή της εφαρμογής αυτής ως τμήμα της ολοκληρωμένης διαδικασίας βελτιστοποίησης. Αποτέλεσμα αυτού θα είναι η συνεργασία του CATIA και των όσων υλοποιήθηκαν, στο πλαίσιο της διπλωματικής εργασίας, με εξελικτικό αλγόριθμο αλλά και με κάποιο λογισμικό επίλυσης του μη-δομημένου πλέγματος, σε πρακτικό πλέον επίπεδο.

Αν και τελικός στόχος είναι η χρήση του επιλύτη ροής των εξισώσεων Navier-Stokes ως το λογισμικό επίλυσης του πλέγματος, εν τούτοις, οι απαιτήσεις ενός τέτοιου εγχειρήματος είναι τεράστιες σε υπολογιστική ισχύ και χρόνο. Για τον λόγο αυτό, κρίθηκε σκόπιμη η δοκιμή του συνόλου σε δύο φάσεις:

1. Εισαγωγή του CATIA σε διαδικασία βελτιστοποίησης με χρήση του λογισμικού EASY (βλέπε παράγραφο 2.3) και ενός ελάχιστα απαιτητικού λογισμικού αντί του επιλύτη (παράγραφος 6.6.1)
2. Συνεργασία του CATIA και των όσων υλοποιήθηκαν, με τον επιλύτη των εξισώσεων Navier – Stokes για τυχαίες, μεμονωμένες υποψήφιες λύσεις, χωρίς επαναληπτική διαδικασία (παράγραφος 6.6.2)

6.6.1 Διαδικασία Βελτιστοποίησης με χρήση του EASY

Σε αυτό το σημείο, αρκεί να καθοριστεί το λογισμικό που θα αντικαταστήσει τον επιλύτη. Το λογισμικό αυτό πρέπει να εκμεταλλεύεται τα δεδομένα που προκύπτουν από τη δημιουργία του πλέγματος και μάλιστα να δέχεται δεδομένα ακριβώς με τον ίδιο τρόπο που τα δέχεται και ο επιλύτης έτσι ώστε να είναι εφικτή η αξιολόγηση ολόκληρης της διαδικασίας συμπεριλαμβανομένης και της επεξεργασίας δεδομένων από το τμήμα του κώδικα `Processing_of_Data`.

Με βάση τα παραπάνω, κατασκευάστηκε ένα εκτελέσιμο αρχείο με την ονομασία Volume.exe το οποίο διαβάζει τα δεδομένα των αρχείων `K.nod` και `K.ele` και υπολογίζει τον όγκο του πλέγματος σε mm^3 .

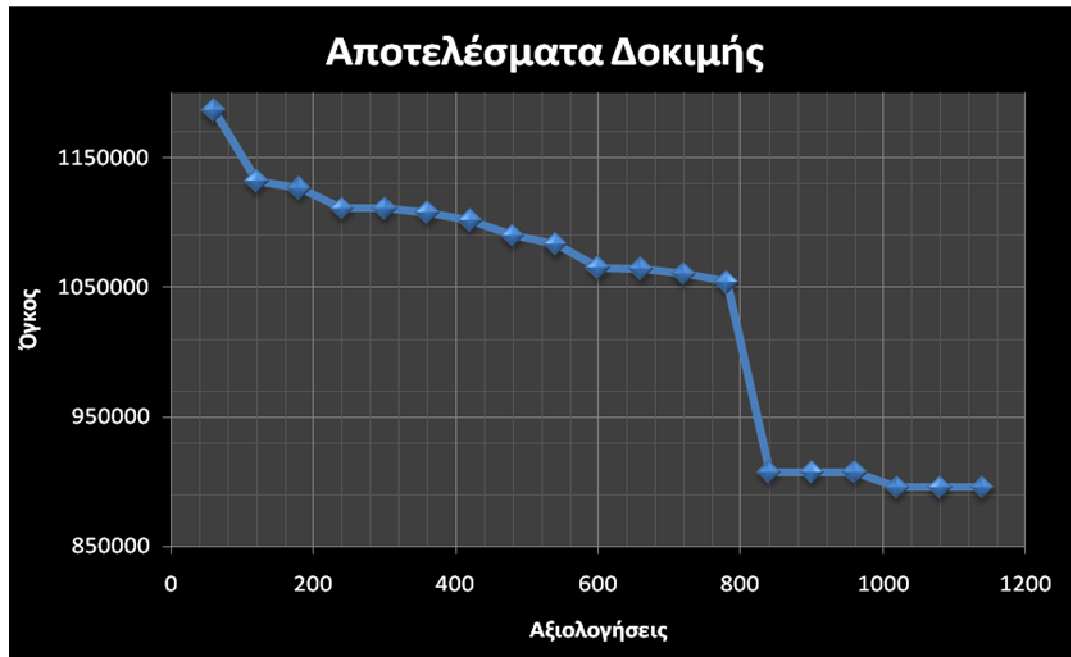
Επειδή όμως, η λύση που θα προέκυπτε, εκτελώντας την διαδικασία με κριτήριο την ελαχιστοποίηση/μεγιστοποίηση του όγκου του γεωμετρικού μοντέλου, είναι προφανής (όλες οι μεταβλητές θα έτειναν στο ελάχιστο/μέγιστο όριό τους), προστέθηκε και ένας επιπλέον περιορισμός στο σύνολο: οι απώλειες πίεσεως του κατακόρυφου αγωγού. Η δομή του εξελικτικού αλγορίθμου επιτρέπει τον έλεγχο ενός τέτοιου περιορισμού διαμέσου ενός αρχείου task.cns και βάσει ενός προκαθορισμένου ανώτατου επιτρεπτού ορίου.

Δημιουργήθηκε εν τέλει ένας κώδικας VB υπό την μορφή εκτελέσιμου αρχείου ονόματι Losses.exe το οποίο υπολογίζει τις απώλειες πίεσεως του κατακόρυφου αγωγού σε bar και αποθηκεύει την τιμή αυτή στο αρχείο `task.cns`. Σημειώνεται ότι για τον υπολογισμό των απωλειών θεωρήθηκε μόνιμη ροή ασυμπίεστου, μη συνεκτικού ρευστού πυκνότητας $\rho = 1293 \text{ kg/m}^3$ και παροχής $V = 0.05 \text{ m}^3/\text{s}$ [Tsa00], [Kos00], [Mat00].

Επειδή με την μείωση της διατομής ενός αγωγού αυξάνονται ταυτόχρονα οι απώλειες πίεσεως κατά μήκος του, συμπεραίνουμε ότι, τυχόν επιθυμία ελαχιστοποίησης του όγκου της γεωμετρίας επιφέρει αύξηση των απωλειών.

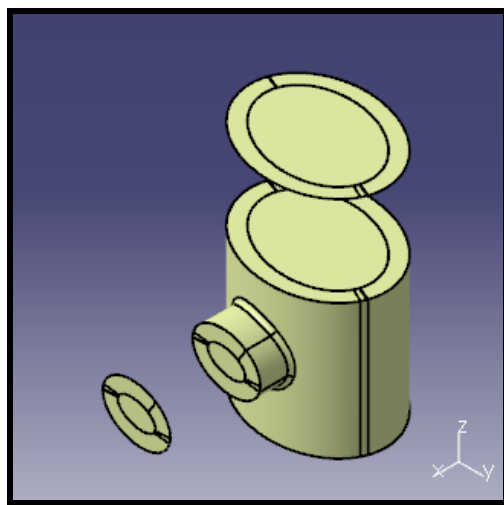
Συνολικά, απαιτήθηκε η εύρεση γεωμετρίας διακλάδωσης μορφής T με τον ελάχιστο όγκο, για την οποία οι απώλειες πιέσεως του κατακόρυφου αγωγού δεν υπερβαίνουν την τιμή 0,3 bar.

Τα αποτελέσματα που προέκυψαν παρουσιάζονται στη συνέχεια υπό τη μορφή γραφήματος:

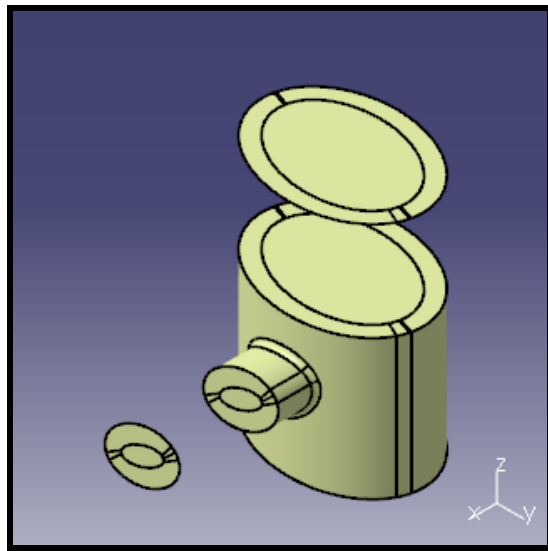


Σχήμα 6.8: Γράφημα τιμών όγκου συναρτήσει του πλήθους των αξιολογήσεων

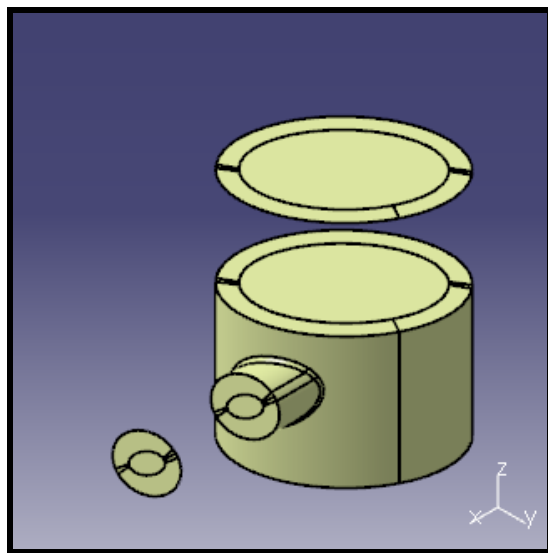
Ενδεικτικά, παρουσιάζονται και οι μορφές ορισμένων γεωμετρικών μοντέλων που προέκυψαν μετά την αξιολόγηση 60, 420 και 840 υποψήφιων λύσεων:



Σχήμα 6.9: Γεωμετρικό μοντέλο (μετά από 60 αξιολογήσεις)



Σχήμα 6.10: Γεωμετρικό μοντέλο (μετά από 420 αξιολογήσεις)



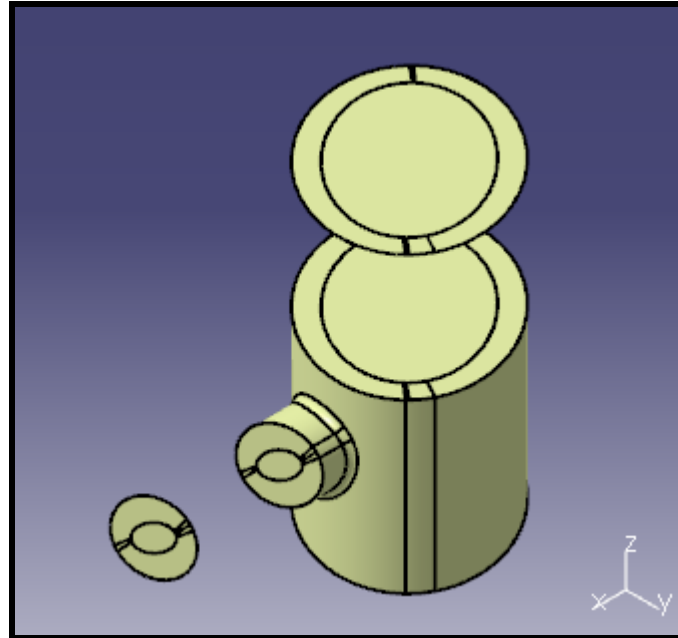
Σχήμα 6.11: Γεωμετρικό μοντέλο (μετά από 840 αξιολογήσεις)

Ως τελική λύση προέκυψε ο εξής συνδιασμός μεταβλητών:

- Μεγάλος άξονας έλλειψης διατομής κατακόρυφου αγωγού = **55.25936630**
- Μικρός άξονας έλλειψης διατομής κατακόρυφου αγωγού = **39.86575092**
- Γωνία κατακόρυφου αγωγού = **43.64850482**
- Μεγάλος άξονας έλλειψης διατομής οριζόντιου αγωγού = **22.60866142**
- Μεγάλος άξονας έλλειψης διατομής οριζόντιου αγωγού = **15.52795276**

- Γωνία οριζόντιου αγωγού = **33.26123528**
- Εσωτερική ακτίνα καμπυλότητας διακλάδωσης = **2.00000000**

ο οποίος αντιστοιχεί σε όγκο **895469.26698925 mm³**.



Σχήμα 6.12: Απεικόνιση βέλτιστης γεωμετρίας

Επίσης παρατηρούνται τα εξής: α) ο αλγόριθμος συγκλίνει σταδιακά προς μία ελάχιστη τιμή όγκου, β) οι διαστάσεις του οριζόντιου αγωγού, του οποίου οι απώλειες δεν έχουν ληφθεί υπόψη στην διαδικασία, τείνουν προς τα κατώτερα όριά τους, δηλαδή στον ελάχιστο όγκο του αγωγού και γ) ο τελικός συνδιασμός μεταβλητών έχει σαν αποτέλεσμα οι απώλειες του κατακόρυφου αγωγού να υπολογίζονται στα **0.297 bar**, οριακά κάτω από το επιτρεπτό όριο.

6.6.2 Δοκιμές Επίλυσης Ροών / Συνεργασία CATIA –Επιλύτη

Η τελική φάση της δοκιμής είναι η επίλυση των εξισώσεων Navier – Stokes από τον επιλύτη του ΕΘΣ (βλέπε κεφάλαιο 5) για ροές ρευστού που περιγράφονται από το μη-δομημένο πλέγμα που εξάγει το CATIA, δηλαδή από τα αρχεία K.nod και K.ele που δημιουργούνται με το πέρας τις διαδικασίας του κεφαλαίου 4.

Θεωρήθηκε μόνιμη ροή αέρα με διαφορές πιέσεως 10 Pa από την διατομή εισόδου του ρευστού σε κάθε διατομή εξόδου. Αναφέρεται επίσης ότι για την αρχικοποίηση της επίλυσης ορίστηκαν σε όλους τους κόμβους πολύ χαμηλές ταχύτητες, σχεδόν μηδενικές.

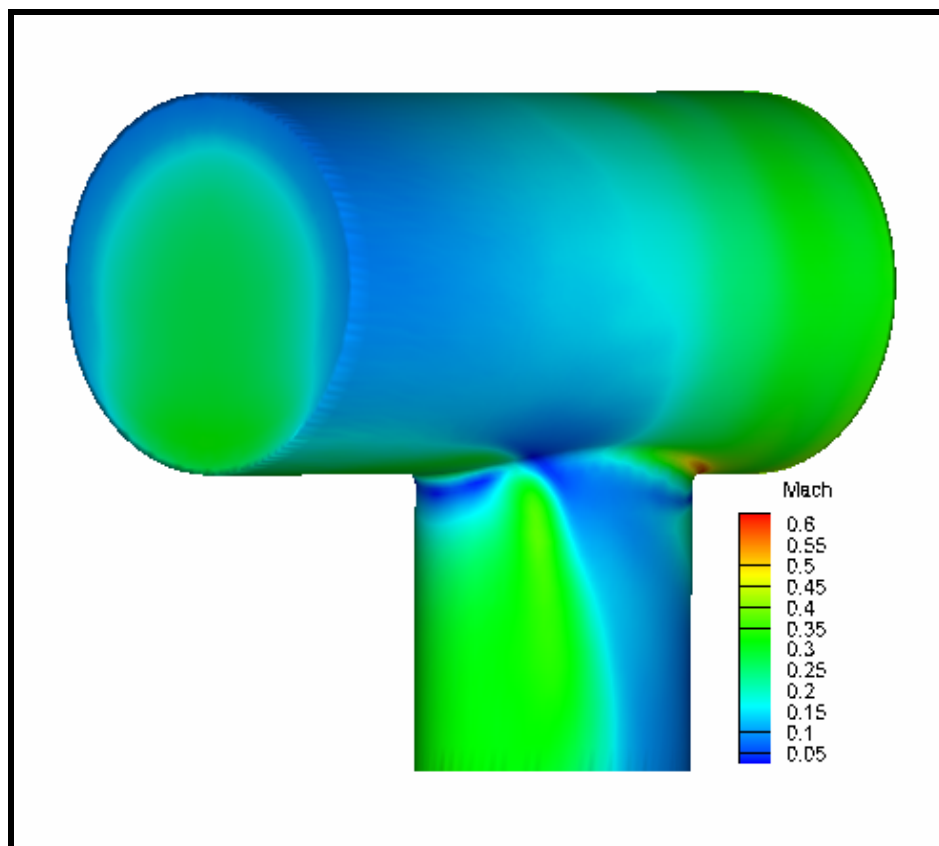
Έγιναν συνολικά 2 δοκιμές για δύο διαφορετικές και τυχαίες γεωμετρίες, τα αποτελέσματα των οποίων παρουσιάζονται στη συνέχεια υπό μορφή εικόνων:

1η Δοκιμή:

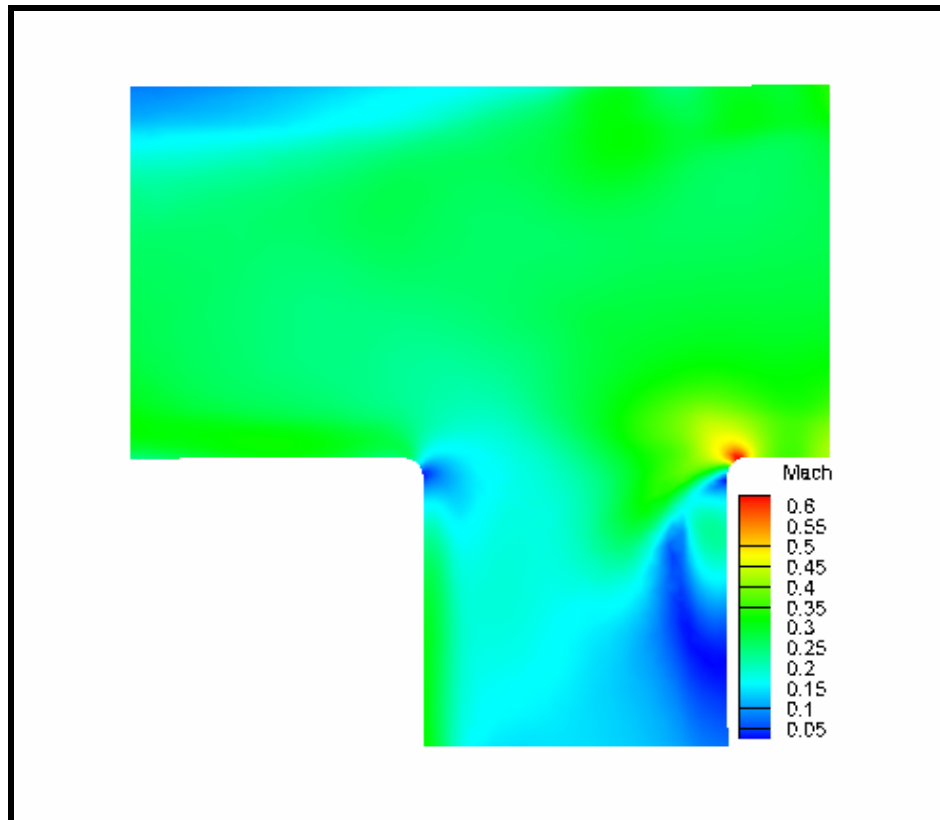
Η γεωμετρία που επιλέχθηκε περιγράφεται από τις τιμές μεταβλητών:

Μεγάλος άξονας έλλειψης διατομής κατακόρυφου αγωγού	40.6
Μικρός άξονας έλλειψης διατομής κατακόρυφου αγωγού	32.4
Γωνία κατακόρυφου αγωγού	48.7
Μεγάλος άξονας έλλειψης διατομής οριζόντιου αγωγού	27.3
Μεγάλος άξονας έλλειψης διατομής οριζόντιου αγωγού	22.3
Γωνία οριζόντιου αγωγού	125.4
Εσωτερική ακτίνα καμπυλότητας διακλάδωσης	6.9

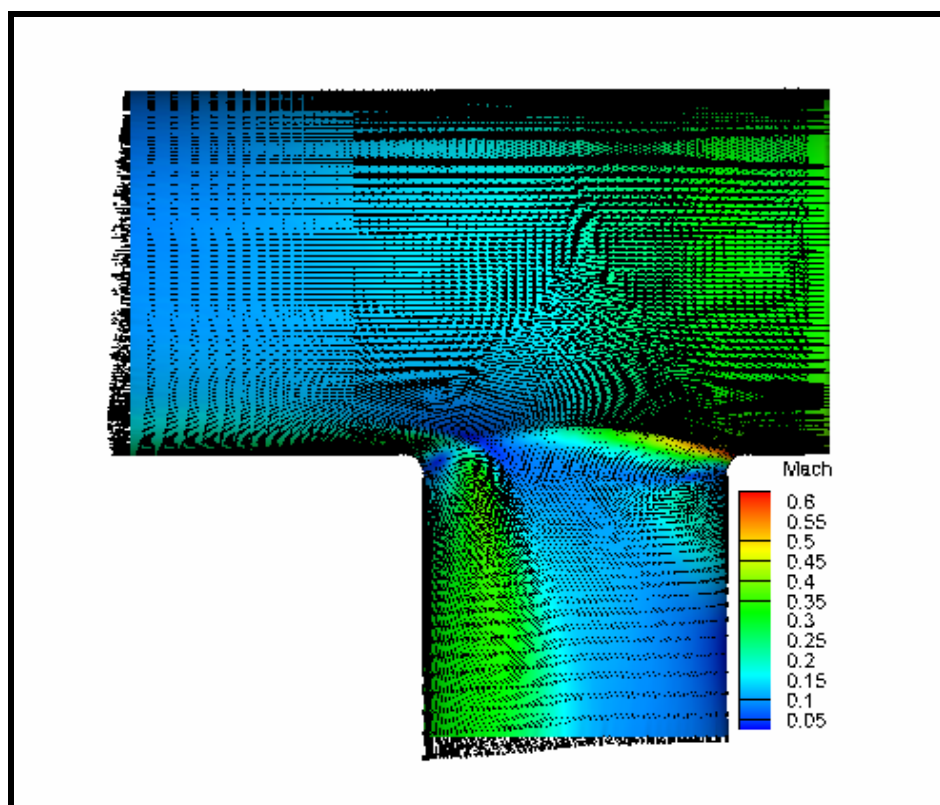
Ενώ τα αποτελέσματα φαίνονται στα σχήματ 6.13, 6.14 και 6.15...



Σχήμα 6.13: Απεικόνιση αριθμού Mach



Σχήμα 6.14: Κάθετη τομή ($y=0$) ροής με απεικόνιση του αριθμού Mach



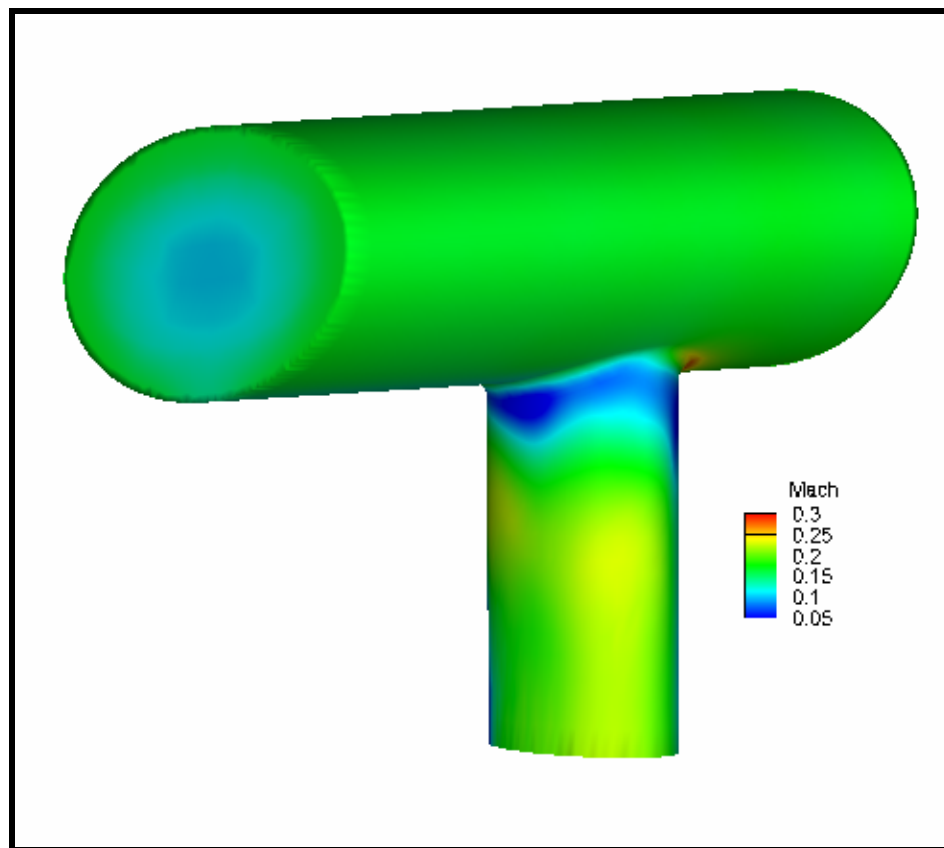
Σχήμα 6.15: Διανύσματα ταχυτήτων της ροής

2η Δοκιμή:

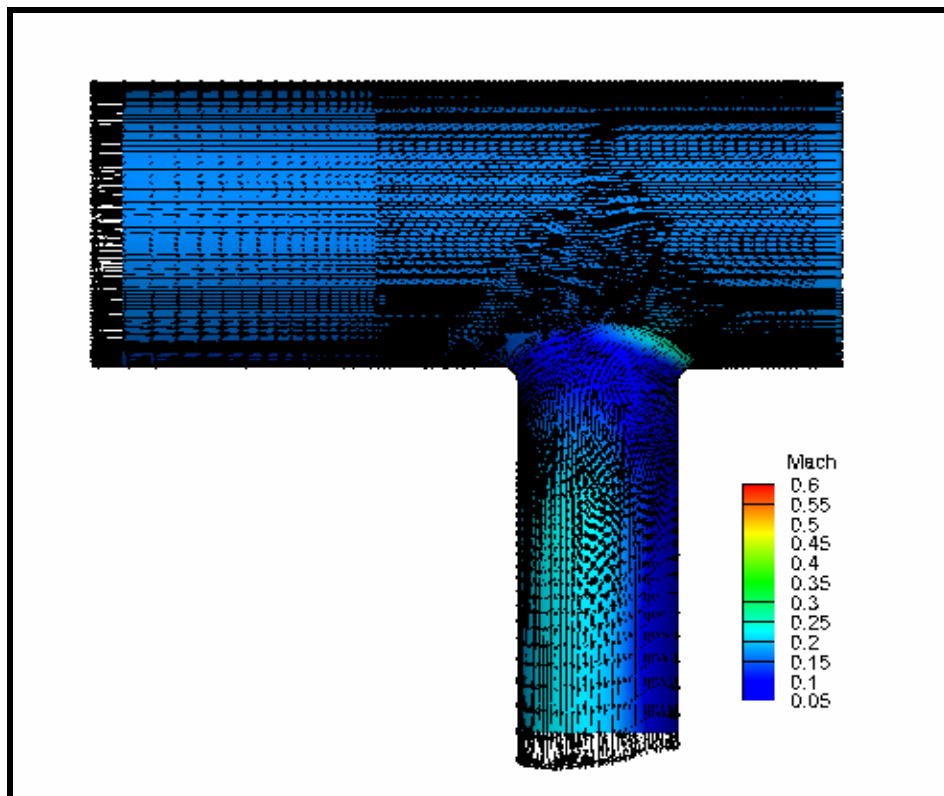
Η γεωμετρία που επιλέχθηκε περιγράφεται από τις τιμές μεταβλητών:

Μεγάλος άξονας έλλειψης διατομής κατακόρυφου αγωγού	59.6
Μικρός άξονας έλλειψης διατομής κατακόρυφου αγωγού	31.1
Γωνία κατακόρυφου αγωγού	-79.4
Μεγάλος άξονας έλλειψης διατομής οριζόντιου αγωγού	29.8
Μεγάλος άξονας έλλειψης διατομής οριζόντιου αγωγού	17.2
Γωνία οριζόντιου αγωγού	0.2
Εσωτερική ακτίνα καμπylότητας διακλάδωσης	5.3

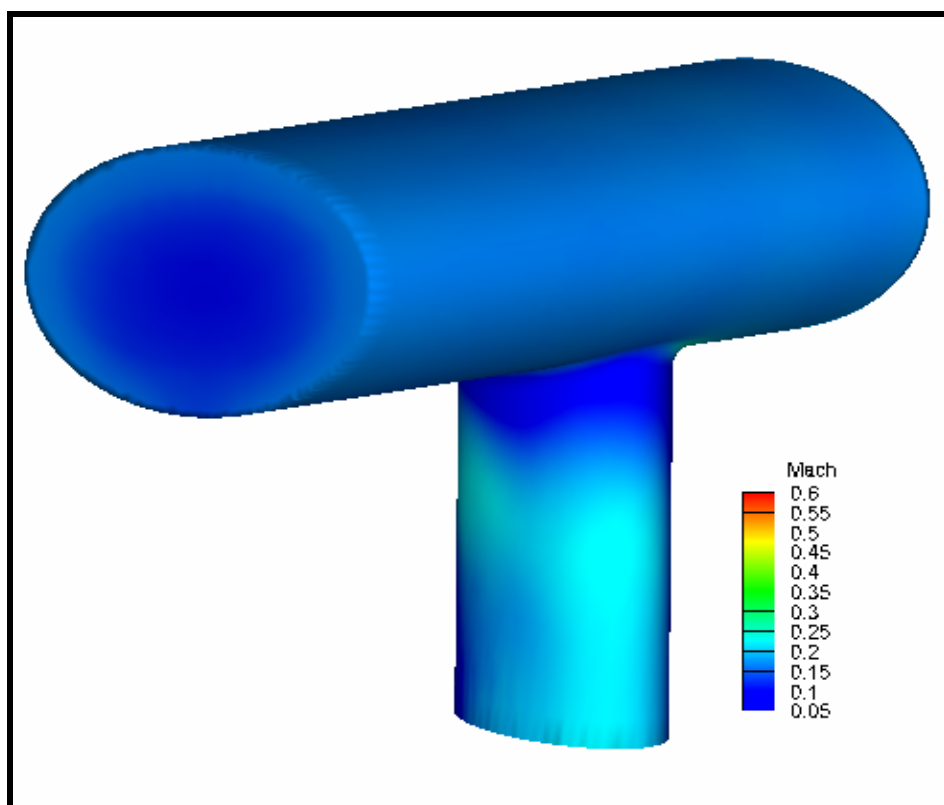
Ενώ τα αποτελέσματα φαίνονται στα σχήματ 6.16, 6.17 και 6.18...



Σχήμα 6.16: Απεικόνιση αριθμού Mach



Σχήμα 6.17: Διανύσματα ταχυτήτων της ροής



Σχήμα 6.18: Απεικόνιση αριθμού Mach σε κλίμακα όμοια με της 1^{ης} δοκιμής

7. Συμπεράσματα και Προτάσεις Μελλοντικής Εργασίας

7.1 Ανακεφαλαίωση / Συμπεράσματα

Λαμβάνοντας υπόψη όλα όσα προηγήθηκαν και συγκρίνοντας μία ολοκληρωμένη διαδικασία βελτιστοποίησης χωρίς την ένταξη του CATIA με μία αντίστοιχη, εμπλουτισμένη με τη χρήση του λογισμικού αυτού και των όσων υλοποιήθηκαν, συμπεραίνουμε τα εξής:

- ✓ Το λογισμικό CATIA, αν και είναι ένα interactive λογισμικό πακέτο σχεδιασμού και έχει κατασκευαστεί για συγκεκριμένες εφαρμογές όπου είναι απαραίτητη η παρέμβαση χρήστη, μπορεί να χρησιμοποιηθεί και ως script based λογισμικό, πλήρως αυτοματοποιημένων λειτουργιών. Το γεγονός αυτό καθιστά εφικτή την συνεργασία του με τους εξελικτικούς αλγορίθμους και του δίνει τη δυνατότητα να λειτουργήσει “κάτω” από αυτούς σε μία διαδικασία βελτιστοποίησης μορφής, όπου απουσιάζει πλήρως η παρέμβαση χρήστη. Με τον τρόπο αυτό πραγματοποιείται και άμεση συνεργασία με μεθόδους Υπολογιστικής Ρευστοδυναμικής ή με άλλες μεθόδους που απαιτούν την επίλυση μερικών διαφορικών εξισώσεων σε πλέγματα.
- ✓ Η ένταξή του σε μεθόδους βελτιστοποίησης μορφής εμπλουτίζει την διαδικασία με την κατασκευή γεωμετρικών μοντέλων από πλήθος τιμών και πιθανώς την δημιουργία του αντίστοιχου μη-δομημένου πλέγματος. Εάν ο γενέτης πλέγματος του CATIA δεν ικανοποιεί τις απαιτήσεις τις εφαρμογής, μπορεί να αντικατασταθεί από κάποιον άλλο περιορίζοντας έτσι το CATIA στην δημιουργία του παραμετροποιημένου γεωμετρικού μοντέλου της εφαρμογής. Στην παρούσα διπλωματική εργασία περιγράφεται και η χρήση του γενέτη πλέγματος. Σε αντίθετη περίπτωση επιβάλλεται η κατάλληλη τροποποίηση του κώδικα TaskCode που παρουσιάστηκε προηγουμένως.
- ✓ Η ενεργοποίηση του λογισμικού είναι χρονοβόρα λόγω του μεγέθους και της πολυπλοκότητάς του. Έτσι, η οποιαδήποτε χρήση του σε τέτοιου είδους διαδικασίες επιφέρει ένα πάγιο κόστος σε χρόνο, το οποίο, ανάλογα με την εφαρμογή, ίσως να επιβαρύνει αισθητά την διαδικασία.

Το κόστος αυτό αυξάνει εάν χρησιμοποιηθεί και ο γενέτης πλέγματος του CATIA για την δημιουργία μη-δομημένων πλεγμάτων μεγάλου πλήθους κόμβων. Φυσικά, σε εφαρμογές όπου απαιτείται η επίκλυση λογισμικού επίλυσης εξισώσεων ροής ή κάποιου αντίστοιχου, σε κάθε επανάληψη, το κόστος αυτό είναι συγκριτικά αναισιώσιμο.

- ✓ Η χρήση της έκδοσης CATIA V5R18 δεν είναι δεσμευτική για την λειτουργικότητα του συνόλου. Παρ' όλ' αυτά, σημειώνεται ότι, δεν είναι εφικτή η χρήση εκδόσεων CATIA V4. Επίσης, όσον αφορά εκδόσεις CATIA V5R16 και προηγούμενες releases από αυτή απαιτείται μία ελάχιστη τροποποίηση του κώδικα TaskCode η οποία σχετίζεται με την δήλωση μεταβλητών. Η έκδοση CATIA V5R17 είναι πλήρως συμβατή με τα όσα υλοποιήθηκαν.
- ✓ Η χρήση του συνόλου και η προσαρμογή του σε κάθε εφαρμογή απαιτεί, εκτός από γνώσεις πάνω στο ίδιο το λογισμικό CATIA, και γνώσεις τις γλώσσας προγραμματισμού Visual Basic. Η έκταση των γνώσεων αυτών είναι μικρή και σχετίζεται με την διαμεταγωγή δεδομένων μεταξύ εξελικτικού αλγορίθμου και CATIA και την αναγνώριση των αρχείων που περιέχουν την εκάστοτε γεωμετρία και το αντίστοιχο πλέγμα.
- ✓ Λόγω της εντολής Sendkeys που εκτελείται για την εξαγωγή του πλέγματος σε αρχείο κειμένου, δεν επιτρέπεται η οποιαδήποτε άλλη χρήση του υπολογιστικού συστήματος κατά την εκτέλεση της διαδικασίας διότι υπάρχει κίνδυνος παύσης αυτής, κάτι που δεν συνέβαινε χωρίς την προσθήκη του CATIA. Σε μία τέτοια περίπτωση απαιτείται χειροκίνητη πληκτρολόγηση των στοιχείων για την τρέχουσα επανάληψη και ενεργοποίηση (activation) εκ νέου του παραθύρου της διαδικασίας έτσι ώστε να συνεχίσει η ομαλή λειτουργία του. Σε περιπτώσεις όπου δεν χρησιμοποιείται ο γενέτης πλέγματος δεν υπάρχει κανένα τέτοιο πρόβλημα.

7.2 Προτάσεις Περαιτέρω Ανάπτυξης του Αντικειμένου

Σε αυτό το σημείο, θεωρείται απαραίτητη η καταγραφή ορισμένων παρατηρήσεων για τα όσα προηγήθηκαν, υπό τη μορφή προτάσεων για περαιτέρω ανάπτυξη του αντικειμένου της παρούσας διπλωματικής εργασίας:

Πρόταση 1: Δημιουργία γραφικού περιβάλλοντος

Στόχος: Απόδοση φιλικού χαρακτήρα στο σύνολο.

Περιγραφή: Η δημιουργία ενός γραφικού περιβάλλοντος φιλικού προς τον χρήστη με δυνατότητα επεξεργασίας όλων των παραμέτρων της διαδικασίας που κατασκευάστηκε προηγουμένως θα μπορούσε να αποδώσει στο σύνολο μία διαφορετική χροιά από αυτήν που του έχει δοθεί έως τώρα, η οποία σίγουρα θα διευκολύνει την εφαρμογή του και πιθανώς να καθιστά εφικτή τη χρήση του από άτομα μη εξοικειωμένα με το αντικείμενο – αρχάριους χρήστες.

Πρόταση 2: Εφαρμογή κατάλληλα προσαρμοσμένου συνόλου σε προβλήματα στατικής ανάλυσης

Στόχος: Επέκταση πεδίου εφαρμογής.

Περιγραφή: Μέχρι στιγμής, το βασικό μέρος του CATIA στη διαδικασία βελτιστοποίησης είναι η δημιουργία ενός μη-δομημένου πλέγματος κατάλληλα σχεδιασμένης γεωμετρίας και η αυτόματη προσαρμογή αυτών σε κάθε γενιά του εξελικτικού αλγορίθμου. Οι δυνατότητες όμως αυτού του λογισμικού δεν περιορίζονται σε αυτή τη λειτουργία μόνο. Εάν δοκιμαστεί η παράλληλη χρήση του τμήματος προσομοίωσης του CATIA στο σύνολο, ίσως να μπορούσε να επεκταθεί το πεδίο εφαρμογής του αντικειμένου και σε προβλήματα στατικής ανάλυσης, και μάλιστα χωρίς την χρήση επιπλέον λογισμικού επίλυσης.

Πρόταση 3: Προσαρμοσμένη εγκατάσταση λογισμικού CATIA

Στόχος: Μείωση απαιτήσεων και πολυπλοκότητας συνόλου.

Περιγραφή: Λόγω του μεγέθους του λογισμικού CATIA αλλά και των υψηλών απαιτήσεών του σε πόρους κατά την δημιουργία μη-δομημένου πλέγματος θα μπορούσε να αποβεί πολύ χρήσιμη η δημιουργία ενός πακέτου εγκατάστασης του λογισμικού υπό τη μορφή αρχείου *.msi το οποίο θα εγκαθιστά, στο υπολογιστικό σύστημα, τα τμήματα εκείνα του CATIA που απαιτούνται για την ολοκλήρωση της διαδικασίας. Το πλεονέκτημα ενός τέτοιου εγχειρήματος είναι ο αποκλεισμός περιττών στοιχείων του

CATIA με συνέπεια τον περιορισμό των απαιτήσεων του συνόλου, τη μείωση του πάγιου κόστους χρόνου που προκύπτει από τη χρήση του και τη μείωση της πολυπλοκότητας που συνοδεύει το συγκεκριμένο CAD – CAM – CAE πρόγραμμα.

Πρόταση 4: Δημιουργία ενιαίου λογισμικού

Στόχος: Πλήρης ένταξη του CATIA στη βελτιστοποίηση με τη δημιουργία ενιαίου υπόβαθρου.

Περιγραφή: Εάν όλα όσα ακολούθησαν συγκεντρωθούν και αποδειχθεί ελπιδοφόρα η εφαρμογή τους στους τομείς της βιομηχανίας αλλά και των ερευνών γενικότερα, η δημιουργία ενός ενιαίου λογισμικού το οποίο θα αποτελεί τον συνδετικό κρίκο μεταξύ εξελικτικών αλγορίθμων και CATIA, συνδυάζοντας τις δυνατότητές τους μέσω ενός κατάλληλα διαμορφωμένου περιβάλλοντος, θα ολοκλήρωνε το σύνολο αποδίδοντας στις μεθόδους βελτιστοποίησης διαφορετική διάσταση, επεκτείνοντας τις δυνατότητες χρήσης τους και εξαπλώνοντάς τες ταυτόχρονα σε πολλούς νέους κλάδους εφαρμογής.

8. Βιβλιογραφία

8.1 Έντυπη Βιβλιογραφία

- [Air00] Airbus UK, “CATIA V5 Foundation Course: Sketcher”, Απρίλιος 2003
- [Air01] Airbus UK, “CATIA V5 Foundation Course: Part Design”, Απρίλιος 2003
- [Bro00] Brown, S., “Visual Basic 6 Complete”, Μάρτιος 1999
- [Coz00] Cozzens, R., “CATIA V5 Workbook Release 17”, Αύγουστος 2007
- [Coz01] Cozzens, R., “Advanced CATIA V5 Workbook: Knowledgeware and Workbenches Release 16”, Σεπτέμβριος 2006
- [For00] Ford, Jerry L., “Microsoft Visual Basic 2005 Express Edition Programming for the Absolute Beginner”, Δεκέμβριος 2005
- [Gian00] Γιαννάκογλου, Κ. Χ., “Μέθοδοι Βελτιστοποίησης στην Αεροδυναμική”, Αθήνα, Νοέμβριος 2003, σελ.7.1 – 7.6
- [Hal00] Halvorson, M., “Microsoft Visual Basic 2005 Step by Step”, Νοέμβριος 2005
- [Kar00] Karam, F. – Kleismit, Charles D., “Using CATIA V5”, Δεκέμβριος 2003
- [Kos00] Κωστόπουλος, Θ. Ν., “Υδραυλικά και Πνευματικά Συστήματα”, Εκδόσεις Συμεών, Αθήνα, 1999
- [Mak00] Μακρής, Π. Α., “Στοιχεία Μηχανών Ι”, Αθήνα, 1998, σελ. 4.13 – 4.22
- [Mat00] Μαθιουλάκης, Δ.Σ. – Αναγνωστόπουλος, Ι.Σ., “Σημειώσεις Βιομηχανικής Ρευστομηχανικής”, Αθήνα, 2003
- [Mic00] Microsoft Corporation, “Microsoft Visual Basic 6.0: Deluxe Learning”, Αύγουστος 1998
- [Pet00] Πετρούτσος, Ε., “Mastering Microsoft Visual Basic 2008”, Μάρτιος 2008
- [Pet01] Πετρούτσος, Ε. – Hough, K., “Visual Basic 6 Developer's Handbook”, Δεκέμβριος 1998

- [Roe00] Roe, P.L., “*Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes*”, *Journal of Computational Physics*, Vol. 43, 1981, σελ. 357-372
- [Tic00] Sham Tickoo, “CATIA V5R18 for Designers”, Αύγουστος 2008
- [Tsa00] Τσαγκάρης, Σ., “Μηχανική των Ρευστών”, Εκδόσεις Συμεών, Αθήνα, 1995
- [Zam00] Zamani, Nader G., “CATIA V5 FEA Tutorials, Releases 12 & 13”, Εκδόσεις SDC
- [Zam01] Zamani, Nader G., “CATIA V5 FEA Tutorials, Release 15”, Εκδόσεις SDC, Μάιος 2006
- [Zam02] Zamani, Nader G., “CATIA V5 FEA Tutorials, Release 17”, Εκδόσεις SDC, Φεβρουάριος 2008

8.2 Ψηφιακή Βιβλιογραφία

- [Cat00] CATIA Documentation
- [Dgt00] http://www.engineeringtoolbox.com/flow-velocity-air-ducts-d_388.html
- [Dgt01] <http://en.wikipedia.org/wiki/CATIA>
- [Eas00] <http://velos0.ltt.mech.ntua.gr/EASY>
- [Ibm00] <http://www-01.ibm.com/software/applications/plm/catia/v5/>
- [Mic01] <http://support.microsoft.com>
- [Mic02] <http://msdn.microsoft.com/en-us/vbasic>

Παράρτημα I

Στη συνέχεια παρουσιάζονται οι γραμμές κώδικα που έχουν καταγραφεί στο βασικό τμήμα του κώδικα TaskCode και των οποίων ο ρόλος είναι η ενεργοποίηση των αρχείων της γεωμετρίας και του πλέγματος - Geometry.CATPart και Geometry.CATAnalysis αντίστοιχα - , ο συγχρονισμός τους με τις τιμές που είναι καταγεγραμμένες στο design.txt και η εξαγωγή του πλέγματος στο αρχείο ExportedMesh_Temp.dat.

Μεταξύ των γραμμών αυτών έχουν καταγραφεί επεξηγήσεις για τις επιμέρους λειτουργίες του κώδικα, οι οποίες συμπληρώνουν την περιγραφή που έχει προηγηθεί σε προηγούμενα κεφάλαια. Επειδή όμως η γλώσσα γραφής του κώδικα είναι η αγγλική, θεωρήθηκε σκόπιμο και οι επεξηγήσεις αυτές να καταγραφούν στην ίδια γλώσσα. Παράλληλα, η αγγλική γλώσσα είναι η μόνη που υποστηρίζεται από όλα τα λειτουργικά συστήματα υπολογιστών ανά τον κόσμο. (Τα όσα αναφέρονται σε αυτή την παράγραφο ισχύουν και για τα παραρτήματα II και III)

Για την καταγραφή αυτών των εντολών κώδικα χρησιμοποιήθηκε η εξής βιβλιογραφία, όπως αυτή αναφέρεται στο κεφάλαιο 8: [Bro00], [For00], [Hal00], [Mic00], [Pet00], [Pet01]. Ομοίως και για τα επόμενα παραρτήματα.

```
' #####
' CODE PART: TaskCode Basic Part
' Purpose:  Open and update the part and analysis files, export the desired mesh
' Description: This code opens the already created part file and synchronizes it with the design.txt
               design table. Then it opens the appropriate analysis file and updates the mesh according
               with the synchronized part. The mesh data are exported in a *.dat file. The path of the
               file is described in the sendkeys.vbs file
' Author:   Tzonis Giannis
' Languages: VBScript
' Locales:  English
' CATIA Level: V5R18
' #####

' #####
' NOTE: In order for this code to work, there must be a sendkeys.vbs file with the appropriate code. The
       path of that file has to be predefined and stable
' #####
```

Dim documents1 As Documents

Set documents1 = CATIA.Documents

```
'Open the part file
Dim partDocument1 As PartDocument
Set partDocument1 = documents1.Open("path\Geometry.CATPart")

Dim part1 As Part
Set part1 = partDocument1.Part

'Update the file
part1.Update

Set partDocument1 = CATIA.ActiveDocument

'Save the updated file
partDocument1.Save

Set documents1 = CATIA.Documents

'Open the analysis file
Dim analysisDocument1 As AnalysisDocument
Set analysisDocument1 = documents1.Open("path\Geometry.CATAnalysis")

'#####
'Make the appropriate zoom and positioning of the geometry
'#####

Dim specsAndGeomWindow1 As SpecsAndGeomWindow
Set specsAndGeomWindow1 = CATIA.ActiveWindow

Dim viewer3D1 As Viewer3D
Set viewer3D1 = specsAndGeomWindow1.ActiveViewer

viewer3D1.Reframe

Dim viewpoint3D1 As Viewpoint3D
Set viewpoint3D1 = viewer3D1.Viewpoint3D

'#####
'Begin resolution script for object : Nodes and Elements
'#####
```

```
Set analysisDocument1 = CATIA.ActiveDocument

Dim analysisManager1 As AnalysisManager
Set analysisManager1 = analysisDocument1.Analysis

Dim analysisModels1 As AnalysisModels
Set analysisModels1 = analysisManager1.AnalysisModels

Dim analysisModel1 As AnalysisModel
Set analysisModel1 = analysisModels1.Item(1)

Dim analysisSets1 As AnalysisSets
Set analysisSets1 = analysisModel1.AnalysisSets

Dim analysisMeshManager1 As AnalysisMeshManager
Set analysisMeshManager1 = analysisSets1.ItemByType("MSHMeshSet")

'update the analysis file
analysisMeshManager1.Update
CATIA.StartCommand "update mesh"

'save the analysis file
analysisDocument1.Save

'#####
'export the updated mesh
'#####

CATIA.StartCommand "export mesh"
Dim cmd As String
cmd = "wscript path\sendkeys_ExportMesh.vbs"
Call Shell(cmd)

'#####
'Close documents and CATIA
'#####

analysisDocument1.Close
partDocument1.Close
```

Παράρτημα ΙΙ

Στο παρόν παράρτημα αναφέρονται οι γραμμές κώδικα που έχουν καταγραφεί στο τμήμα VarUpdate του συνολικού κώδικα TaskCode. Όπως φαίνεται και από τις επιμέρους επεξηγήσεις εντός του κώδικα, ρόλος αυτών είναι η ανανέωση των τιμών του design.txt σε κάθε επανάληψη της διαδικασίας, βάσει των τιμών του αρχείου task.dat. Επίσης, μεταξύ αυτών υπολογίζονται και οι υπόλοιπες μεταβλητές, δηλαδή αυτές που δεν αφορούν τον εξελικτικό αλγόριθμο.

```
' #####
' CODE PART: VarUpdate
' Purpose: The update of the design.txt file according to the values in the task.dat file.
' Description: The code reads the values of the variables in the task.dat file and updates the design.txt
               file, that is linked to the Geometry.CATPart document, accordingly.
' Author: Tzonis Giannis
' Languages: VBScript
' Locales: English
' CATIA Level: V5R18
' #####

'#####
'Values Input
'#####

'Open the input file and read the values
Open "path\task.dat" For Input As #1

Input #1, NumVar
Input #1, Radius11
Input #1, Radius12
Input #1, Angle1
Input #1, Radius21
Input #1, Radius22
Input #1, Angle2
Input #1, Fillet1

'Close the file
Close #1
```

```

#####
'Depended Variables Calculation
#####

'Definiton of constants
pi = 3.14159265

'Calculate the depended variables
InnerRadius11 = Radius11 - 10
InnerRadius12 = Radius12 - 10
InnerRadius21 = Radius21 - 10
InnerRadius22 = Radius22 - 10
Fillet2 = Fillet1 + 5

'Calculation of horizontal tube's length
If Angle1 = 0 Then
    Translate5 = Radius11 + Fillet1 + 20
    Translate6 = Translate5
ElseIf Angle1 > 0 Then
    Rad1A = Angle1 * pi / 180
    Rad1B = (90 - Angle1) * pi / 180
    A1 = Radius11 * Cos(Rad1A)
    B1 = Radius12 * Cos(Rad1B)
    If A1 >= B1 Then
        Translate5 = A1 + Fillet1 + 20
        Translate6 = Translate5
    ElseIf A1 < B1 Then
        Translate5 = B1 + Fillet1 + 20
        Translate6 = Translate5
    End If
ElseIf Angle1 < 0 Then
    Rad1A = Angle1 * pi / 180
    Rad1B = (90 + Angle1) * pi / 180
    A1 = Radius11 * Cos(Rad1A)
    B1 = Radius12 * Cos(Rad1B)
    If A1 >= B1 Then
        Translate5 = A1 + Fillet1 + 20
        Translate6 = Translate5
    ElseIf A1 < B1 Then

```

Translate5 = B1 + Fillet1 + 20

Translate6 = Translate5

End If

End If

'Calculation of vertical tube's length and horizontal tube's offset

If Angle2 < 90 Then

Rad2A = Angle2 * pi / 180

Rad2B = (90 - Angle2) * pi / 180

A2 = Radius21 * Sin(Rad2A)

B2 = Radius22 * Sin(Rad2B)

If A2 >= B2 Then

Offset = A2 + Fillet1 + 30

Translate3 = Offset + A2 + Fillet1 + 20

Translate4 = Translate3

ElseIf A2 < B2 Then

Offset = B2 + Fillet1 + 30

Translate3 = Offset + B2 + Fillet1 + 20

Translate4 = Translate3

End If

ElseIf Angle2 > 90 Then

Rad2A = Angle2 * pi / 180

Rad2B = (Angle2 - 90) * pi / 180

A2 = Radius21 * Sin(Rad2A)

B2 = Radius22 * Sin(Rad2B)

If A2 >= B2 Then

Offset = A2 + Fillet1 + 30

Translate3 = Offset + A2 + Fillet1 + 20

Translate4 = Translate3

ElseIf A2 < B2 Then

Offset = B2 + Fillet1 + 30

Translate3 = Offset + B2 + Fillet1 + 20

Translate4 = Translate3

End If

ElseIf Angle2 = 90 Then

Offset = Radius21 + Fillet1 + 30

Translate3 = Offset + Radius21 + Fillet1 + 20

Translate4 = Translate3

End If


```
'#####  
'Values Output in design.txt  
'#####
```

```
'Open the output file and store the updated values
```

```
Open "path\Design.txt" For Output As #2
```

```
Print #2, "PartBody\Sketch.1\Radius.5\Radius (mm)" & vbTab; Radius11
```

```
Print #2, "PartBody\Sketch.1\Radius.7\Radius (mm)" & vbTab; Radius12
```

```
Print #2, "PartBody\Sketch.1\Angle.12\Angle (deg)" & vbTab; Angle1
```

```
Print #2, "PartBody\Sketch.2\Radius.17\Radius (mm)" & vbTab; InnerRadius11
```

```
Print #2, "PartBody\Sketch.2\Radius.19\Radius (mm)" & vbTab; InnerRadius12
```

```
Print #2, "PartBody\Translate.1\Length (mm)" & vbTab; 30
```

```
Print #2, "PartBody\Translate.2\Length (mm)" & vbTab; 30
```

```
Print #2, "PartBody\Translate.3\Length (mm)" & vbTab; Translate3
```

```
Print #2, "PartBody\Translate.4\Length (mm)" & vbTab; Translate4
```

```
Print #2, "PartBody\Sketch.3\Radius.24\Radius (mm)" & vbTab; Radius21
```

```
Print #2, "PartBody\Sketch.3\Radius.26\Radius (mm)" & vbTab; Radius22
```

```
Print #2, "PartBody\Sketch.3\Angle.30\Angle (deg)" & vbTab; Angle2
```

```
Print #2, "PartBody\Sketch.3\Offset.31\Offset (mm)" & vbTab; Offset
```

```
Print #2, "PartBody\Sketch.4\Radius.35\Radius (mm)" & vbTab; InnerRadius21
```

```
Print #2, "PartBody\Sketch.4\Radius.37\Radius (mm)" & vbTab; InnerRadius22
```

```
Print #2, "PartBody\Translate.5\Length (mm)" & vbTab; Translate5
```

```
Print #2, "PartBody\Translate.6\Length (mm)" & vbTab; Translate6
```

```
Print #2, "PartBody\Translate.7\Length (mm)" & vbTab; 60
```

```
Print #2, "PartBody\Translate.8\Length (mm)" & vbTab; 60
```

```
Print #2, "PartBody\Translate.9\Length (mm)" & vbTab; 60
```

```
Print #2, "PartBody\Translate.10\Length (mm)" & vbTab; 60
```

```
Print #2, "PartBody\Fillet.1\Radius (mm)" & vbTab; Fillet1
```

```
Print #2, "PartBody\Fillet.2\Radius (mm)" & vbTab; Fillet2
```

```
'close the file
```

```
Close #2
```

Παράρτημα ΙΙΙ

Το τελευταίο τμήμα του κώδικα TaskCode, όπως έχει αναφερθεί, έχει ονομαστεί Processing_of_Data. Οι γραμμές κώδικα που είναι καταγεγραμμένες σε αυτό παρουσιάζονται στη συνέχεια, εμπλουτισμένες με τις απαραίτητες επεξηγήσεις, όπως φαίνεται και από τις τελευταίες, το τμήμα αυτό του κώδικα επεξεργάζεται τα δεδομένα του πλέγματος, υπό τη μορφή που τα εξάγει το λογισμικό CATIA και τα μετατρέπει σε άλλη μορφή έτσι ώστε να μπορούν να γίνουν αποδεκτά από τον επιλύτη των εξισώσεων ροής (βλέπε κεφάλαιο 5). Επίσης, συμπληρώνει τα δεδομένα υπολογίζοντας και καταγράφοντας τα logfr των κόμβων του πλέγματος.

```
' #####
' CODE PART: Processing_of_Data
' Purpose:   Creation of K.nod and K.ele that are needed for the Thermal Engines Lab
' Description: The code takes the data from the ExportedMesh_Temp.dat file and the files
               BoundariesMesh.txt, OutputMesh.txt and InputMesh.txt and creates 2 files that have the
               essential data in a particular format so that they can be used by the Thermal Engines
               Lab.
' Author:    Tzonis Giannis
' Languages: VBScript
' Locales:   English
' CATIA Level: V5R18
' #####

'#####
'Number of Nodes Caclulation
'#####

j = 0
Open "path/ExportedMesh_Temp.dat" For Input As #1
For i = 1 To 11
    Line Input #1, A
Next
Do Until B = "$.."
    Line Input #1, B
    j = j + 1
Loop
j = j - 1
NumNodes = j / 2
```

```

#####
'Number of Tetrahedrons Calculation
#####

k = 0
Do Until EOF(1)
  Line Input #1, text1
  If text1 = "$..      MESH PART: MSHPartGHS3D.1" Then
    Input #1, Text2
    C = Input(6, #1)
    Do Until C <> "CTETRA"
      Input #1, D, E, F, G, H, i
      If C = "CTETRA" Then
        k = k + 1
      End If
      C = Input(6, #1)
    Loop
  End If

  If text1 = "$..      MESH PART: MSHPartGHS3D.2" Then
    Input #1, Text2
    C = Input(6, #1)
    Do Until C <> "CTETRA"
      Input #1, D, E, F, G, H, i
      If C = "CTETRA" Then
        k = k + 1
      End If
      C = Input(6, #1)
    Loop
  End If

  If text1 = "$..      MESH PART: MSHPartGHS3D.3" Then
    Input #1, Text2
    C = Input(6, #1)
    Do Until C <> "CTETRA"
      Input #1, D, E, F, G, H, i
      If C = "CTETRA" Then
        k = k + 1
      End If
      C = Input(6, #1)
    Loop
  End If

```

```

    Loop
End If

If text1 = "$..      MESH PART: MSHPartGHS3D.4" Then
    Input #1, Text2
    C = Input(6, #1)
    Do Until C <> "CTETRA"
        Input #1, D, E, F, G, H, i
        If C = "CTETRA" Then
            k = k + 1
        End If
        C = Input(6, #1)
    Loop
End If

If text1 = "$..      MESH PART: MSHPartGHS3D.5" Then
    Input #1, Text2
    C = Input(6, #1)
    Do Until C <> "CTETRA"
        Input #1, D, E, F, G, H, i
        If C = "CTETRA" Then
            k = k + 1
        End If
        C = Input(6, #1)
    Loop
End If

If text1 = "$..      MESH PART: MSHPartGHS3D.6" Then
    Input #1, Text2
    C = Input(6, #1)
    Do Until C <> "CTETRA"
        Input #1, D, E, F, G, H, i
        If C = "CTETRA" Then
            k = k + 1
        End If
        C = Input(6, #1)
    Loop
End If

Loop
```

NumTet = k

Close #1

'#####

'Logfr assignment

'#####

'Create array named Nodes() with the values of logfr for each node

Dim Nodes() As Integer

ReDim Nodes(1 To NumNodes)

'Assign logfr to the nodes of the Input(4), Boundaries(3) and Output(5) mesh parts

Open "path/ExportedMesh_Temp.dat" For Input As #1

Do Until EOF(1)

Line Input #1, text1

If text1 = "\$.. MESH PART: MSHPartSmartSurf.1" Then

Line Input #1, Text2

A = Input(6, #1)

Do Until A <> "CTRIA3"

Input #1, B, C, D, E, F

Nodes(D) = 3

Nodes(E) = 3

Nodes(F) = 3

A = Input(6, #1)

Loop

End If

If text1 = "\$.. MESH PART: MSHPartSmartSurf.3" Then

Line Input #1, Text2

A = Input(6, #1)

Do Until A <> "CTRIA3"

Input #1, B, C, D, E, F

Nodes(D) = 4

Nodes(E) = 4

Nodes(F) = 4

A = Input(6, #1)

Loop

End If

If text1 = "\$.. MESH PART: MSHPartSmartSurf.6" Then

Line Input #1, Text2

```

A = Input(6, #1)
Do Until A <> "CTRIA3"
    Input #1, B, C, D, E, F
    Nodes(D) = 4
    Nodes(E) = 4
    Nodes(F) = 4
    A = Input(6, #1)
Loop
End If

If text1 = "$..      MESH PART: MSHPartExtrTranslation.1" Then
    Line Input #1, Text2
    A = Input(6, #1)
    Do Until A <> "CTRIA3"
        Input #1, B, C, D, E, F
        Nodes(D) = 3
        Nodes(E) = 3
        Nodes(F) = 3
        A = Input(6, #1)
    Loop
End If

If text1 = "$..      MESH PART: MSHPartSmartSurf.9" Then
    Line Input #1, Text2
    A = Input(6, #1)
    Do Until A <> "CTRIA3"
        Input #1, B, C, D, E, F
        Nodes(D) = 5
        Nodes(E) = 5
        Nodes(F) = 5
        A = Input(6, #1)
    Loop
End If

If text1 = "$..      MESH PART: MSHPartSmartSurf.10" Then
    Line Input #1, Text2
    A = Input(6, #1)
    Do Until A <> "CTRIA3"
        Input #1, B, C, D, E, F
        Nodes(D) = 5

```

```
Nodes(E) = 5
Nodes(F) = 5
A = Input(6, #1)
Loop
End If

If text1 = "$..      MESH PART: MSHPartExtrTranslation.3" Then
Line Input #1, Text2
A = Input(6, #1)
Do Until A <> "CTRIA3"
Input #1, B, C, D, E, F
Nodes(D) = 3
Nodes(E) = 3
Nodes(F) = 3
A = Input(6, #1)
Loop
End If

If text1 = "$..      MESH PART: MSHPartSmartSurf.11" Then
Line Input #1, Text2
A = Input(6, #1)
Do Until A <> "CTRIA3"
Input #1, B, C, D, E, F
Nodes(D) = 5
Nodes(E) = 5
Nodes(F) = 5
A = Input(6, #1)
Loop
End If

If text1 = "$..      MESH PART: MSHPartSmartSurf.12" Then
Line Input #1, Text2
A = Input(6, #1)
Do Until A <> "CTRIA3"
Input #1, B, C, D, E, F
Nodes(D) = 5
Nodes(E) = 5
Nodes(F) = 5
A = Input(6, #1)
Loop
```

```

End If
Loop

Close #1

'Assign logfr to the nodes of the Input - Boundaries(34) mesh part
Open " path /ExportedMesh_Temp.dat" For Input As #1

Do Until EOF(1)
    Line Input #1, text1
    If text1 = "$..      MESH PART: MSHPart1D.1" Then
        Line Input #1, Text2
        A = Input(4, #1)
        Do Until A <> "CBAR"
            Input #1, B, C, D, E
            Nodes(D) = 34
            Nodes(E) = 34
            A = Input(4, #1)
        Loop
    End If
Loop
Close #1

'Create the files Translation1Mesh.txt, Translation2Mesh.txt, VOutputRingMesh.txt,
HOutputRingMesh.txt which contain the description of some mesh parts that are needed for the
upcoming assignment of some multi logfrs(34-35).
Open " path /ExportedMesh_Temp.dat" For Input As #1
Do Until EOF(1)
    Line Input #1, text1
    If text1 = "$..      MESH PART: MSHPartExtrTranslation.1" Then
        Open "path/Translation1Mesh.txt" For Output As #2
        Input #1, Text2
        A = Input(6, #1)
        Do Until A <> "CTRIA3"
            Input #1, B, C, D, E, F
            Print #2, A & vbTab; B & vbTab; C & vbTab; D & vbTab; E & vbTab; F
            A = Input(6, #1)
        Loop
    End If
Loop
Close #2

```



```

If text1 = "$..      MESH PART: MSHPartExtrTranslation.3" Then
  Open " path /Translation3Mesh.txt" For Output As #2
  Input #1, Text2
  A = Input(6, #1)
  Do Until A <> "CTRIA3"
    Input #1, B, C, D, E, F
    Print #2, A & vbTab; B & vbTab; C & vbTab; D & vbTab; E & vbTab; F
    A = Input(6, #1)
  Loop
End If
Close #2

If text1 = "$..      MESH PART: MSHPartSmartSurf.9" Then
  Open " path /VOutputRingMesh.txt" For Output As #2
  Input #1, Text2
  A = Input(6, #1)
  Do Until A <> "CTRIA3"
    Input #1, B, C, D, E, F
    Print #2, A & vbTab; B & vbTab; C & vbTab; D & vbTab; E & vbTab; F
    A = Input(6, #1)
  Loop
End If
Close #2

If text1 = "$..      MESH PART: MSHPartSmartSurf.11" Then
  Open " path /HOutputRingMesh.txt" For Output As #2
  Input #1, Text2
  A = Input(6, #1)
  Do Until A <> "CTRIA3"
    Input #1, B, C, D, E, F
    Print #2, A & vbTab; B & vbTab; C & vbTab; D & vbTab; E & vbTab; F
    A = Input(6, #1)
  Loop
End If
Close #2
Loop

Close #1

```

'Count the lines of each *.txt file in order to use that number later on for the upcoming comparisons

```
Open " path /Translation1Mesh.txt" For Input As #1
    Translation1_Counter = 0
    Do Until EOF(1)
        Line Input #1, text1
        Translation1_Counter = Translation1_Counter + 1
    Loop
Close #1
```

```
Open " path /Translation3Mesh.txt" For Input As #1
    Translation3_Counter = 0
    Do Until EOF(1)
        Line Input #1, text1
        Translation3_Counter = Translation3_Counter + 1
    Loop
Close #1
```

```
Open " path /VOutputRingMesh.txt" For Input As #1
    VOutputRing_Counter = 0
    Do Until EOF(1)
        Line Input #1, text1
        VOutputRing_Counter = VOutputRing_Counter + 1
    Loop
Close #1
```

```
Open " path /HOutputRingMesh.txt" For Input As #1
    HOutputRing_Counter = 0
    Do Until EOF(1)
        Line Input #1, text1
        HOutputRing_Counter = HOutputRing_Counter + 1
    Loop
Close #1
```

'Compare the nodes in the appropriate *.txt file and assign the appropriate logfr

```
Open " path /Translation1Mesh.txt" For Input As #1
i = 0
j = 0
For i = 1 To Translation1_Counter
    A1 = Input(6, #1)
```

Input #1, B1, C1, D1, E1, F1

For j = 1 To VOutputRing_Counter

 If j = 1 Then

 Open " path /VOutputRingMesh.txt" For Input As #2

 End If

 A2 = Input(6, #2)

 Input #2, B2, C2, D2, E2, F2

 If D1 = D2 Then

 Nodes(D1) = 35

 ElseIf D1 = E2 Then

 Nodes(D1) = 35

 ElseIf D1 = F2 Then

 Nodes(D1) = 35

 End If

 If E1 = D2 Then

 Nodes(E1) = 35

 ElseIf E1 = E2 Then

 Nodes(E1) = 35

 ElseIf E1 = F2 Then

 Nodes(E1) = 35

 End If

 If F1 = D2 Then

 Nodes(F1) = 35

 ElseIf F1 = E2 Then

 Nodes(F1) = 35

 ElseIf F1 = F2 Then

 Nodes(F1) = 35

 End If

 If j = VOutputRing_Counter Then

 Close #2

 End If

Next

Next

Close #1

Open " path /Translation3Mesh.txt" For Input As #1

For i = 1 To Translation3_Counter

A1 = Input(6, #1)

Input #1, B1, C1, D1, E1, F1

For j = 1 To HOutputRing_Counter

If j = 1 Then

Open " path /HOutputRingMesh.txt" For Input As #2

End If

A2 = Input(6, #2)

Input #2, B2, C2, D2, E2, F2

If D1 = D2 Then

Nodes(D1) = 35

ElseIf D1 = E2 Then

Nodes(D1) = 35

ElseIf D1 = F2 Then

Nodes(D1) = 35

End If

If E1 = D2 Then

Nodes(E1) = 35

ElseIf E1 = E2 Then

Nodes(E1) = 35

ElseIf E1 = F2 Then

Nodes(E1) = 35

End If

If F1 = D2 Then

Nodes(F1) = 35

ElseIf F1 = E2 Then

Nodes(F1) = 35

ElseIf F1 = F2 Then

Nodes(F1) = 35

End If

If j = HOutputRing_Counter Then

```

        Close #2
    End If
Next
Next

Close #1

'#####
'Coordinates Recording
'#####

'Create array named Coor(,) with the coordinates of each node
Dim Coor()
ReDim Coor(1 To NumNodes, 1 To 3)

'Read Coordinates from ExportedMesh_Temp.dat and store them in the array
i = 0
j = 0
Open " path /ExportedMesh_Temp.dat" For Input As #1

For i = 1 To 11
    Line Input #1, A
Next
text1 = Input(5, #1)

Do Until text1 <> "GRID*"
    Input #1, B, XCoor, YCoor, C
    Text2 = Input(8, #1)
    Input #1, ZCoor
    Coor(B, 1) = XCoor
    Coor(B, 2) = YCoor
    Coor(B, 3) = ZCoor
    text1 = Input(5, #1)
Loop

Close #1

'#####
'Creation of K.nod file
'#####

```

```
Open " path /K.nod" For Output As #1

'Print Number of Nodes
Write #1, NumNodes

'Print logfr of each node
For l = 1 To NumNodes
    Print #1, Nodes(l) & Space(1);
Next

'Print on a new line X cordimates for each node
Print #1, NextLine
For j = 1 To NumNodes
    Print #1, Coor(j, 1) & Space(1);
Next

'Print on a new line Y cordimates for each node
Print #1, NextLine
For j = 1 To NumNodes
    Print #1, Coor(j, 2) & Space(1);
Next

'Print on a new line Z cordimates for each node
Print #1, NextLine
For j = 1 To NumNodes
    Print #1, Coor(j, 3) & Space(1);
Next

Close #1

#####
'Creation of K.ele file
#####

Open " path /ExportedMesh_Temp.dat" For Input As #1
Open " path /K.ele" For Output As #2

'Print the number of tetrahedrons that have been created
Write #2, NumTet
```

'Read and print the nodes of each tetrahedron

Do Until EOF(1)

Line Input #1, text1

If text1 = "\$.. MESH PART: MSHPartGHS3D.1" Then

Line Input #1, text1

A = Input(6, #1)

Do Until A <> "CTETRA"

Input #1, B, C, Node1, Node2, Node3, Node4

Print #2, Node1 & Space(1) & Node2 & Space(1) & Node3 & Space(1) & Node4 & Space(1);

A = Input(6, #1)

Loop

End If

If text1 = "\$.. MESH PART: MSHPartGHS3D.2" Then

Line Input #1, text1

A = Input(6, #1)

Do Until A <> "CTETRA"

Input #1, B, C, Node1, Node2, Node3, Node4

Print #2, Node1 & Space(1) & Node2 & Space(1) & Node3 & Space(1) & Node4 & Space(1);

A = Input(6, #1)

Loop

End If

If text1 = "\$.. MESH PART: MSHPartGHS3D.3" Then

Line Input #1, text1

A = Input(6, #1)

Do Until A <> "CTETRA"

Input #1, B, C, Node1, Node2, Node3, Node4

Print #2, Node1 & Space(1) & Node2 & Space(1) & Node3 & Space(1) & Node4 & Space(1);

A = Input(6, #1)

Loop

End If

If text1 = "\$.. MESH PART: MSHPartGHS3D.4" Then

Line Input #1, text1

A = Input(6, #1)

Do Until A <> "CTETRA"

Input #1, B, C, Node1, Node2, Node3, Node4

Print #2, Node1 & Space(1) & Node2 & Space(1) & Node3 & Space(1) & Node4 & Space(1);

```
A = Input(6, #1)
Loop
End If

If text1 = "$.. MESH PART: MSHPartGHS3D.5" Then
    Line Input #1, text1
    A = Input(6, #1)
    Do Until A <> "CTETRA"
        Input #1, B, C, Node1, Node2, Node3, Node4
        Print #2, Node1 & Space(1) & Node2 & Space(1) & Node3 & Space(1) & Node4 & Space(1);
        A = Input(6, #1)
    Loop
End If

If text1 = "$.. MESH PART: MSHPartGHS3D.6" Then
    Line Input #1, text1
    A = Input(6, #1)
    Do Until A <> "CTETRA"
        Input #1, B, C, Node1, Node2, Node3, Node4
        Print #2, Node1 & Space(1) & Node2 & Space(1) & Node3 & Space(1) & Node4 & Space(1);
        A = Input(6, #1)
    Loop
End If
Loop

Close #1
Close #2

'Close CATIA
CATIA.Quit
```