



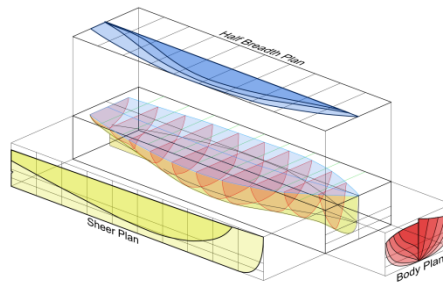
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ ΜΕΛΕΤΗΣ ΠΛΟΙΟΥ & ΘΑΛΑΣΣΙΩΝ
ΜΕΤΑΦΟΡΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ:

«ΠΑΡΑΜΕΤΡΙΚΗ ΣΧΕΔΙΑΣΗ ΓΑΣΤΡΑΣ ΜΕ ΧΡΗΣΗ ΤΗΣ ΣΥΣΤΗΜΑΤΙΚΗΣ ΣΕΙΡΑΣ FORMDATA»



ΦΕΛΕΣΑΚΗΣ ΠΑΝΑΓΙΩΤΗΣ

ΕΠΙΒΛΕΠΩΝ: Α.ΓΚΙΝΗΣ, ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

ΑΘΗΝΑ, ΦΕΒΡΟΥΑΡΙΟΣ 2013

ΠΕΡΙΕΧΟΜΕΝΑ

	<u>Σελίδα</u>
1.ΕΙΣΑΓΩΓΗ	9
1.1.Γενικά	9
1.2.Σκοπός της εργασίας	9
1.3.Δομή της εργασίας.....	10
2.B-SPLINES.....	11
2.1.Ορισμός	11
2.2.Ο αλγόριθμος De Boor	11
2.3.Ιδιότητες καμπύλων B-Spline	14
3.Η ΣΥΣΤΗΜΑΤΙΚΗ ΣΕΙΡΑ FORMDATA	15
3.1.Χαρακτηριστικά της σειράς:	15
3.2.Παραδείγματα ονοματολογίας.....	17
3.3.Πίνακας διαθέσιμων οικογενειών	18
3.4.Διαδικασία εφαρμογής της σειράς:	19
4.ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΑ ΠΑΚΕΤΑ ΛΟΓΙΣΜΙΚΟΥ	22
4.1.Rhinoceros 3D.....	22
4.2.Grasshopper.....	22
4.3.CATIA.....	23
4.4.Python.....	23
5.ΚΑΤΑΣΚΕΥΗ ΤΡΙΣΔΙΑΣΤΑΤΩΝ ΚΑΜΠΥΛΩΝ.....	24
5.1.Δεδομένα Εισόδου (Αρχεία)	24
5.2.Κατασκευή και αποθήκευση των καμπύλων.....	27
5.2.1.Περιγραφή κατασκευής καμπύλων στο Grasshopper	27
5.2.1.A.Επεξεργασία των σημείων ελέγχου.....	28
5.2.1.B.Επεξεργασία των κομβοδιανυσμάτων.....	29
5.2.1.C.Κατασκευή των καμπύλων.....	30
5.2.1.D.Κατασκευή των ονομάτων των καμπύλων	31
5.2.1.E.Αποθήκευση των καμπύλων.....	32
5.2.2.Grasshopper Components index.....	33
6.ΚΑΤΑΣΚΕΥΗ ΤΟΥ ΤΡΙΣΔΙΑΣΤΑΤΟΥ ΜΟΝΤΕΛΟΥ.....	37
6.1.Δημιουργία βιβλιοθήκης καμπύλων στο CATIA.....	37
6.2.Σχεδίαση των προφίλ πλήρης και πρύμνης.....	40
6.2.1.Πρύμνη Καταδρομικού.....	41

6.2.2.Πρύμνη Άβακας.....	42
6.2.3.Πλώρη Χωρίς Βολβό.....	43
6.2.4.Βολβοειδής Πλώρη B0.....	44
6.2.5.Βολβοειδής Πλώρη B4-5.....	45
6.2.6.Βολβοειδής Πλώρη B8-10.....	46
6.3.Επιλογή οικογένειας πλώρης και πρύμνης.....	47
6.4.Κατασκευή βοηθητικών γεωμετρικών στοιχείων.....	48
6.4.1.Αντιγραφή των νομέων και των προφίλ.....	52
6.4.2.Δημιουργία Top & Lower points επί των νομέων.....	53
6.4.3.Δημιουργία του Deck Plane.....	55
6.4.4.Δημιουργία των σημείων των flat of side και flat of bottom.....	56
6.4.4.1.Κατασκευή αρχείων με τις συντεταγμένες των σημείων των flat of side και flat of bottom στο Grasshopper.....	56
6.4.4.2.Κατασκευή των σημείων των flat of side και flat of bottom στο CATIA.....	58
6.4.5.Δημιουργία της καμπύλης του Flat of Bottom.....	62
6.4.6.Δημιουργία πρύμνης άβακα τύπου D.....	67
6.4.7.Δημιουργία της Deckline.....	71
6.4.8.Δημιουργία πρύμνης άβακα τύπου ABC.....	73
6.4.9.Δημιουργία της Center Line.....	76
6.4.10.Δημιουργία της καμπύλης του Flat of Side.....	78
6.4.11.Δημιουργία των Waterlines του Flat of Side.....	85
6.4.Διαστατοποίηση αδιάστατου μοντέλου.....	89
6.5.Κατασκευή επιφανειών.....	90
6.5.1.Κατασκευή ενδιάμεσου τμήματος της επιφάνειας της γάστρας.....	90
6.5.2.Δημιουργία επιφάνειας πρύμνης.....	104
6.5.2.1.Δημιουργία Πρύμνης Καταδρομικού.....	104
6.5.2.2.Δημιουργία Πρύμνης Άβακα τύπου A,B,C.....	110
6.5.2.3.Δημιουργία Πρύμνης Άβακα τύπου D.....	111
6.5.3.Κατασκευή επιφάνειας βολβοειδούς πλώρης.....	112
7.ΠΑΡΑΔΕΙΓΜΑΤΑ.....	117
7.1.Κατηγορία 1: (πλώρη χωρίς βολβό , πρύμνη καταδρομικού).....	118
7.2.Κατηγορία 2: (πλώρη χωρίς βολβό , πρύμνη άβακας A-B-C).....	120
7.3.Κατηγορία 3: (πλώρη χωρίς βολβό , πρύμνη άβακας D).....	122
7.4.Κατηγορία 4: (βολβοειδής πλώρη , πρύμνη καταδρομικού).....	124

7.5.Κατηγορία 5: (βολβοειδής πλώρη , πρύμνη άβακας A-B-C).....	126
7.6.Κατηγορία 6: (βολβοειδής πλώρη , πρύμνη άβακας D).....	128
8.ΠΑΡΑΡΤΗΜΑΤΑ	130
8.1.Παράρτημα 1: «Η Συστηματική Σειρά Formdata»	130
8.2.Παράρτημα 2: «Catia Objects & Functions»	161
8.3.Παράρτημα 3: «Περιεχόμενα προγράμματος ανά γραμμή κώδικα».....	169
9. ΒΙΒΛΙΟΓΡΑΦΙΑ	174

Ευρετήριο Εικόνων

	<u>Σελίδα</u>
Εικόνα 1: Το αρχείο DEBOOR.DB της οικογένειας B01F90.....	25
Εικόνα 2: Το αρχείο KNOT.KV της οικογένειας B01F90.....	26
Εικόνα 3: Συνολική επισκόπηση του καμβά στο Grasshopper.	27
Εικόνα 4: Επεξεργασία των σημείων ελέγχου	28
Εικόνα 5: Επεξεργασία των Κομβοδιανυσμάτων	29
Εικόνα 6: Κατασκευή των καμπύλων	30
Εικόνα 7: Κατασκευή των ονομάτων των καμπύλων	31
Εικόνα 8: Αποθήκευση των καμπύλων	32
Εικόνα 9: Ιεραρχία τμήματος βιβλιοθήκης του CATIA που περιέχει τους νομείς της FORMDATA.....	38
Εικόνα 10: Παράδειγμα ιεραρχημένης οργάνωσης των νομέων εντός της βιβλιοθήκης του CATIA.....	39
Εικόνα 11: Ιεραρχία τμήματος βιβλιοθήκης του CATIA που περιέχει τα προφίλ της FORMDATA.....	40
Εικόνα 12: Κατασκευή προφίλ πρύμνης καταδρομικού	41
Εικόνα 13: Κατασκευή προφίλ πρύμνης άβακα.....	42
Εικόνα 14: Κατασκευή προφίλ πλήρης χωρίς βολβό	43
Εικόνα 15: Κατασκευή προφίλ βολβοειδούς πλήρης τύπου B ₀	44
Εικόνα 16: Κατασκευή προφίλ βολβοειδούς πλήρης τύπου B ₄₋₅	45
Εικόνα 17: Κατασκευή προφίλ βολβοειδούς πλήρης τύπου B ₈₋₁₀	46
Εικόνα 18: παράδειγμα οργάνωσης με geometric set	50
Εικόνα 19: παράδειγμα οργάνωσης με geometric set	51
Εικόνα 20: Αντιγραφή των νομέων και των προφίλ	52
Εικόνα 21: Δημιουργία ανώτερων ακραίων σημείων νομέων και προφίλ.....	53
Εικόνα 22: Δημιουργία κατώτερων ακραίων σημείων νομέων και προφίλ.....	54
Εικόνα 23: Κατασκευή επιπέδου κυρίου καταστρώματος	55
Εικόνα 24 : Δημιουργία σημείων Flat of Side.....	58
Εικόνα 25 : Δημιουργία σημείων Flat of Bottom.....	59
Εικόνα 26α : foscateory=both	60
Εικόνα 26b : foscateory=fore	60
Εικόνα 26c : foscateory=aft.....	60
Εικόνα 26d : foscateory=none.....	60
Εικόνα 27a: Οι ευθείες που ορίζουν την κλίση.....	61
Εικόνα 27b: Οι κατασκευαζόμενες ευθείες.....	61
Εικόνα 27c: Δημιουργία των ακραίων κορυφαίων σημείων του FoS.....	62

Εικόνα 28a: Δημιουργία ευθειογενούς τμήματος της καμπύλης του Flat of Bottom	63
Εικόνα 28b: Δημιουργία προωραίου τμήματος της καμπύλης του Flat of Bottom.....	63
Εικόνα 28c: Δημιουργία πρυμναίου τμήματος της καμπύλης του Flat of Bottom.....	64
Εικόνα 28d: Δημιουργία προωραίου σημείου κατάληξης της καμπύλης του Flat of Bottom..	65
Εικόνα 28e: Δημιουργία πρυμναίου σημείου κατάληξης της καμπύλης του Flat of Bottom .	65
Εικόνα 28f: Δημιουργία ακραίου προωραίου τμήματος της καμπύλης του Flat of Bottom ...	66
Εικόνα 28g: Δημιουργία ακραίου πρυμναίου τμήματος της καμπύλης του Flat of Bottom .	66
Εικόνα 29a: Κατασκευή του επιπέδου του Flat of Side	68
Εικόνα 29b:Κατασκευή της Transom Deck Line.....	68
Εικόνα 29c: Κατασκευή του DeckLine Transom End Point.....	68
Εικόνα 29e: Κατασκευή του 0_extention Top Point.....	69
Εικόνα 29f: Κατασκευή του Move Point	69
Εικόνα 29g: Κατασκευή της Transom Vertical Part	70
Εικόνα 29i: Κατασκευή του Transom Point.....	70
Εικόνα 29j: Κατασκευή του Transom Spline.....	71
Εικόνα 30a: Δημιουργία παράλληλου τμήματος.....	72
Εικόνα 30b: Δημιουργία προωραίου τμήματος της γραμμής καταστρώματος	72
Εικόνα 30b: Δημιουργία πρυμναίου τμήματος της γραμμής καταστρώματος.....	73
Εικόνα 31a:Δημιουργία του TransomPlane	74
Εικόνα 31b: Δημιουργία της προέκτασης της DeckLineAft.....	74
Εικόνα 31c: Δημιουργία ακραίου σημείου της νέας DeckLine	75
Εικόνα 31d: Δημιουργία του TransomPoint.....	75
Εικόνα 31e: Δημιουργία του TransomSpline	75
Εικόνα 31f: Δημιουργία της TransomDeckLine	76
Εικόνα 32a: Δημιουργία προωραίου τμήματος της Center Line.....	77
Εικόνα 32b: Δημιουργία μεσαίου τμήματος της Center Line	77
Εικόνα 32c: Δημιουργία πρυμναίου τμήματος της Center Line	78
Εικόνα 33a: Κατασκευή ευθειογενούς τμήματος της καμπύλης του Flat of Side (πρύμνη καταδρομικού).....	79
Εικόνα 33b: Κατασκευή προωραίου spline της καμπύλης του Flat of Side (πρύμνη καταδρομικού).....	80
Εικόνα 33c: Κατασκευή πρυμναίου spline της καμπύλης του Flat of Side (πρύμνη πρύμνη καταδρομικού).....	80
Εικόνα 33d: Κατασκευή προωραίου ακραίου spline της καμπύλης του Flat of Side (πρύμνη πρύμνη καταδρομικού).....	81
Εικόνα 33e: Κατασκευή πρυμναίου ακραίου spline της καμπύλης του Flat of Side (πρύμνη καταδρομικού).....	81
Εικόνα 33f: Κατασκευή ευθειογενών τμημάτων της καμπύλης του Flat of Side (πρύμνη άβακας τύπου D)	82

Εικόνα 33g: Κατασκευή προραίου spline της καμπύλης του Flat of Side (πρύμνη άβακας τύπου D).....	83
Εικόνα 33h: Κατασκευή μεσαίου spline της καμπύλης του Flat of Side (πρύμνη άβακας τύπου D).....	83
Εικόνα 33i: Κατασκευή πρυμναίου spline της καμπύλης του Flat of Side (πρύμνη άβακας τύπου D).....	84
Εικόνα 33j: Κατασκευή ακραίου προραίου spline της καμπύλης του Flat of Side (πρύμνη άβακας τύπου D).....	84
Εικόνα 34a: Δημιουργία επιπέδου fore fos waterline plane.....	86
Εικόνα 34b: Δημιουργία τομών του επιπέδου fore fos waterline plane και των προραίων νομέων.....	86
Εικόνα 34c: Δημιουργία προραίας FoS ισάλου (fore fos waterline).....	87
Εικόνα 34d: Δημιουργία επιπέδου aft fos waterline plane.....	87
Εικόνα 34e: Δημιουργία τομών του επιπέδου aft fos waterline plane και των πρυμναίων νομέων.....	88
Εικόνα 34f: Δημιουργία πρυμναίας FoS ισάλου (aft fos waterline).....	88
Εικόνα 35: Διαστατοποίηση τρισδιάστατου μοντέλου.....	89
Εικόνα 36: Κατασκευή επιφάνειας του Flat of Side.....	91
Εικόνα 37: Κατασκευή επιφάνειας του Flat of Bottom.....	92
Εικόνα 38: Δημιουργία καμπύλης GeneralCL	92
Εικόνα 39: Δημιουργία καμπύλης GeneralDeckLine	93
Εικόνα 40: Διαίρεση των νομέων.....	93
Εικόνα 41: Δημιουργία κορυφαίων ακραίων σημείων και ευθειών των νομέων και των προφίλ.....	94
Εικόνα 42: Κατασκευή της DeckLine Support Surface.....	95
Εικόνα 43: Δημιουργία κατώτατων ακραίων σημείων και ευθειών των νομέων και του προφίλ της πρύμνης.....	95
Εικόνα 44: Δημιουργία ευθειών στα άκρα του επίπεδου πυθμένα.....	96
Εικόνα 45: Δημιουργία επιφάνειας CL Support Surface.....	96
Εικόνα 46: Δημιουργία επιφανειών General DeckLine support surface και General CL support surface.....	97
Εικόνα 47: Κατασκευή επιφάνειας ανάμεσα στο Flat of Side και το Flat of Bottom.....	98
Εικόνα 48: Δημιουργία της επιφάνειας κάτω από την προραία FoS ίσαλο.....	99
Εικόνα 49: Δημιουργία της επιφάνειας πάνω από την προραία FoS ίσαλο.....	99
Εικόνα 50: Δημιουργία επιφάνειας “GAP”.....	100
Εικόνα 51: Συνένωση προραίων επιφανειών μέχρι το νομέα 9.000.....	101
Εικόνα 52: Δημιουργία επιφάνειας μη βολβοειδούς πλήρης (νομείς 9.000-10.00).....	101
Εικόνα 53: Δημιουργία ακραίου τμήματος επιφάνειας μη βολβοειδούς πλήρης.....	102
Εικόνα 54: Δημιουργία επιφάνειας κάτω από την πρυμναία FoS ίσαλο.....	102
Εικόνα 55: Δημιουργία επιφάνειας πάνω από την πρυμναία FoS ίσαλο.....	103

Εικόνα 56: Δημιουργία επιφάνειας “AFT-GAP”	103
Εικόνα 57: Κατασκευή AftCruiserSurface	104
Εικόνα 58: Βοηθητικές γεωμετρίες επί του νομέα 0.000	105
Εικόνα 59: Βοηθητικές γεωμετρίες επί του νομέα 0.500	106
Εικόνα 60: Δημιουργία πρυμναίου βοηθητικού spline	106
Εικόνα 61: Κατασκευή της LOWER_0.5-0_SURFACE	107
Εικόνα 62: Κατασκευή της MIDDLE_0.5-0_SURFACE	108
Εικόνα 63: Κατασκευή TOP_0.5-0_SURFACE	109
Εικόνα 64: Κατασκευή πρύμνης άβακα τύπου ABC	110
Εικόνα 65: Κατασκευή πρύμνης άβακα τύπου D	111
Εικόνα 66: Κατασκευή βοηθητικών γεωμετρικών στοιχείων για τη σχεδίαση της επιφάνειας βολβοειδούς πλώρης	112
Εικόνα 67: Κατασκευή του κάτω τμήματος της επιφάνειας της βολβοειδούς πλώρης	113
Εικόνα 68: Κατασκευή της κορυφής του βολβού	114
Εικόνα 69: Κατασκευή ενδιάμεσης επιφάνειας βολβοειδούς πλώρης	115
Εικόνα 70: Κατασκευή του επάνω τμήματος της επιφάνειας της βολβοειδούς πλώρης	116
Εικόνα 71Α: Τρισδιάστατο μοντέλο κατηγορίας 1 (πλάγια όψη)	118
Εικόνα 71Β: Τρισδιάστατο μοντέλο κατηγορίας 1 (τρισδιάστατη όψη 1)	118
Εικόνα 71Γ: Τρισδιάστατο μοντέλο κατηγορίας 1 (τρισδιάστατη όψη 2)	119
Εικόνα 72Α: Τρισδιάστατο μοντέλο κατηγορίας 2 (πλάγια όψη)	120
Εικόνα 72Β: Τρισδιάστατο μοντέλο κατηγορίας 2 (τρισδιάστατη όψη 1)	120
Εικόνα 72Γ: Τρισδιάστατο μοντέλο κατηγορίας 2 (τρισδιάστατη όψη 2)	121
Εικόνα 73Α: Τρισδιάστατο μοντέλο κατηγορίας 3 (πλάγια όψη)	122
Εικόνα 73Β: Τρισδιάστατο μοντέλο κατηγορίας 3 (τρισδιάστατη όψη 1)	122
Εικόνα 73Γ: Τρισδιάστατο μοντέλο κατηγορίας 3 (τρισδιάστατη όψη 2)	123
Εικόνα 74Α: Τρισδιάστατο μοντέλο κατηγορίας 4 (πλάγια όψη)	124
Εικόνα 74Β: Τρισδιάστατο μοντέλο κατηγορίας 4 (τρισδιάστατη όψη 1)	124
Εικόνα 74Γ: Τρισδιάστατο μοντέλο κατηγορίας 4 (τρισδιάστατη όψη 2)	125
Εικόνα 75Α: Τρισδιάστατο μοντέλο κατηγορίας 5 (πλάγια όψη)	126
Εικόνα 75Β: Τρισδιάστατο μοντέλο κατηγορίας 5 (τρισδιάστατη όψη 1)	126
Εικόνα 75Γ: Τρισδιάστατο μοντέλο κατηγορίας 5 (τρισδιάστατη όψη 2)	127
Εικόνα 76Α: Τρισδιάστατο μοντέλο κατηγορίας 6 (πλάγια όψη)	128
Εικόνα 76Β: Τρισδιάστατο μοντέλο κατηγορίας 6 (τρισδιάστατη όψη 1)	128
Εικόνα 76Γ: Τρισδιάστατο μοντέλο κατηγορίας 6 (τρισδιάστατη όψη 2)	129

1.ΕΙΣΑΓΩΓΗ

1.1.Γενικά

Η συστηματική σειρά FORMDATA δίνει τη δυνατότητα στο σχεδιαστή να παράξει γάστρες πλοίων με συγκεκριμένα χαρακτηριστικά (συντελεστή γάστρας, τύπο πλώρης και πρύμνης, διαστάσεις). Εύκολα γίνεται αντιληπτό ότι οι πιθανοί συνδυασμοί πλώρης πρύμνης αλλά και ο συντελεστής γάστρας δημιουργούν έναν πολύ μεγάλο αριθμό γαστρών που μπορούν να σχεδιαστούν εντός των ορίων της συστηματικής σειράς FORMDATA.

Η FORMDATA παρέχει στο σχεδιαστή τους νομείς και τα προφίλ πλώρης και πρύμνης, για να μπορέσει στη συνέχεια ο ίδιος, αφού διαλέξει με μία συγκεκριμένη διαδικασία ποιά θα χρησιμοποιήσει, να σχεδιάσει τη γάστρα. Στην παρούσα διπλωματική εργασία, έγινε προσπάθεια αυτοματοποίησης όλης αυτής της διαδικασίας. Μέσω ενός λογισμικού που δημιουργήθηκε, δίδεται η δυνατότητα στον σχεδιαστή, εισάγοντας ως δεδομένα εισόδου στο πρόγραμμα, συγκεκριμένα χαρακτηριστικά του πλοίου το οποίο θέλει να σχεδιάσει, να παίρνει ως αποτέλεσμα αυτόματα την γάστρα του πλοίου.

1.2.Σκοπός της εργασίας

Σκοπός της παρούσας διπλωματικής εργασίας είναι η δημιουργία ενός ολοκληρωμένου λογισμικού το οποίο θα δίνει τη δυνατότητα στο χρήστη να παράγει αυτόματα οποιαδήποτε γάστρα μπορεί να παραχθεί μέσα από τη χρήση της συστηματικής σειράς FORMDATA.

Βασικός στόχος κατά την εκπόνηση της παρούσας διπλωματικής εργασίας ήταν δημιουργία ενός λογισμικού το οποίο θα παράγει αποτέλεσμα (τριδιάστατο μοντέλο γάστρας πλοίου) για οποιονδήποτε πιθανό συνδυασμό πλώρης και πρύμνης, μέσα στα όρια εφαρμογής της FORMDATA. Το εγχείρημα αυτό δεν ήταν εύκολο διότι ο κάθε πιθανός συνδυασμός πλώρης-πρύμνης είχε τις δικές του ιδιαιτερότητες, τόσο στη σχεδίαση κάποιων βοηθητικών γεωμετριών (πχ της Deck Line, των καμπύλων του flat of bottom και flat of side κλπ) ,όσο και στη σχεδίαση της ίδιας της επιφάνειας της γάστρας. Πέραν των ιδιαιτεροτήτων στη σχεδίαση που παρουσιάζει η κάθε οικογένεια πλώρης και πρύμνης ξεχωριστά , αυτές προσαυξάνονται ανάλογα με τον εκάστοτε συνδυασμό, αλλά και τον συντελεστή γάστρας του υπό σχεδίαση μοντέλου. Συνεπώς όλες οι κατασκευές που πραγματοποιήθηκαν , έπρεπε να γίνουν με γνώμονα το να είναι εφικτή η υλοποίηση τους για οποιοδήποτε συνδυασμό οικογένειας πλώρης και πρύμνης αλλά και για όλους τους πιθανούς συντελεστές γάστρας.

Τελικά δημιουργήθηκε ένα πρόγραμμα σε γλώσσα python ,το οποίο εκπληρώνει σε αρκετά μεγάλο βαθμό τις αρχικές προσδοκίες και στόχους. Παράγεται γάστρα σε όλες τις περιπτώσεις (πιθανοί συνδυασμοί πλώρης και πρύμνης), με την καλύτερη δυνατή ποιότητα των επιφανειών που σχεδιάζονται, δεδομένων των διαθέσιμων εργαλείων για την κατασκευή τους, αλλά και το πλήθος πιθανών συνδυασμών και συνεπώς διαφορετικών κατασκευών.

1.3.Δομή της εργασίας

Η παρούσα διπλωματική εργασία αποτελείται από τα ακόλουθα κεφάλαια:

- Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στο θέμα της διπλωματικής, αναφέρεται ο σκοπός της εργασίας και περιγράφονται συνοπτικά τα κεφάλαια που την απαρτίζουν.
- Στο δεύτερο κεφάλαιο δίνεται μία σύντομη περιγραφή των καμπύλων B-Splines καθώς και ορισμένων ιδιοτήτων τους.
- Στο τρίτο κεφάλαιο περιγράφεται η συστηματική σειρά FORMDATA, καθώς και η διαδικασία εφαρμογής της για την παραγωγή μίας γάστρας πλοίου.
- Στο τέταρτο κεφάλαιο ακολουθεί μία σύντομη περιγραφή όλων των λογισμικών και γλωσσών προγραμματισμού που χρησιμοποιήθηκαν για την υλοποίηση της διπλωματικής εργασίας.
- Στο πέμπτο κεφάλαιο περιγράφεται ο τρόπος με τον οποίο κατασκευάστηκαν οι τρισδιάστατες καμπύλες (νομείς).
- Στο έκτο κεφάλαιο γίνεται εκτενής περιγραφή της κατασκευής του τρισδιάστατου μοντέλου, από την τοποθέτηση των τρισδιάστατων καμπύλων στις σωστές θέσεις στο χώρο, μέχρι την διαστατοποίησή τους και την κατασκευή της επιφάνειας της γάστρας.
- Στο έβδομο κεφάλαιο παρουσιάζονται ορισμένα παραδείγματα εφαρμογής του δημιουργηθέντος προγράμματος, τα οποία απεικονίζουν όλους τους δυνατούς τύπους γαστρών που μπορούν να δημιουργηθούν με τη χρήση της FORMDATA (όσον αφορά τους πιθανούς συνδυασμούς ανάλογα με τον τύπο της πλώρης ή της πρύμνης).

2.B-SPLINES

2.1.Ορισμός

Δεδομένων $n+1$ σημείων $d_i, i=0,1,\dots,n$, και ενός κομβοδιανύσματος $T=\{t_0,t_1,\dots,t_{n+k}\}$, ονομάζουμε τη διανυσματική συνάρτηση:

με

καμπύλη B-spline τάξεως k με κομβοδιάνυσμα T και σημεία ελέγχου ή σημεία De Boor τα $d_i, i=0,1,\dots,n$. Η πολυγωνική γραμμή, που παρεμβάλλει τα σημεία De Boor d_i κατά τη διαδοχή που επάγει ο δείκτης i , θα ονομάζεται πολύγωνο ελέγχου. Προφανώς κάθε σημείο De Boor d_j , επηρεάζει την $Q(t)$ στο παραμετρικό υποδιάστημα, παρατήρηση η οποία αναδεικνύει την τοπικότητα των καμπύλων B-Splines. Η προηγούμενη ιδιότητα εξασφαλίζει στον σχεδιαστή πως, καθώς μετακινεί το σημείο De Boor d_j , η καμπύλη επηρεάζεται μόνο στο τμήμα της που αποτελεί εικόνα του υποδιαστήματος $[t_j, t_{j+k}]$.

2.2.Ο αλγόριθμος De Boor

Ο αλγόριθμος De boor επιτρέπει τον υπολογισμό των σημείων μιας καμπύλης B-spline $Q(u)$ μέσω επαναληπτικής γραμμικής παρεμβολής. Αν αντικαταστήσουμε στην σχέση 1α την αναδρομική σχέση ορισμού των συναρτήσεων B-splines (1b), η πρώτη λαμβάνει την ακόλουθη μορφή:

Με κατάλληλη αλλαγή μεταβλητής ($j=i+1$, $j=1$ για $i=0$ και $j=n+1$ για $i=n$) στο δεύτερο μέλος της σχέσης 2, με κριτήριο την εμφάνιση κοινής συνάρτησης B-spline και στα δύο αθροίσματα της σχέσης 2, και λαμβάνοντας υπ' όψιν ότι όταν εμφανίζεται σημείο de boor με δείκτη εκτός του $\{0,1,\dots,n\}$ αυτό λαμβάνεται ίσο με 0, έχουμε:

(3)

Άρα η σχέση (2), λόγω της (3), αφού εκτείνουμε προς τα άνω το πρώτο άθροισμα στο δεξί μέλος της (2), γίνεται:

$$\text{-----}$$

Άρα:

$$\text{-----}$$

Συγκρίνοντας τη σχέση (2) με την σχέση (5) παρατηρείται ότι έχει λάβει χώρα αυτό το οποίο αναφέρεται στη βιβλιογραφία ως μεταφορά βαθμού (degree transfer) από τις συναρτήσεις B-spline στα σημεία de boor. Επιπλέον αξίζει να σημειωθεί ότι το σημείο στο διάστημα $[t_i, t_{i+k-1}]$ στο οποίο επηρεάζει την καμπύλη $Q(t)$ εκφράζεται ως κυρτός συνδυασμός των σημείων De boor d_{i-1} και d_i .

Μετά από $j(\leq k-1)$ επαναλήψεις της παραπάνω διαδικασίας, η καμπύλη B-spline αποκτά την ακόλουθη αναπαράσταση:

με

$$\text{-----}$$

Στο τελικό επίπεδο $j=k-1$ η (6) λαμβάνει την ακόλουθη μορφή:

Εάν τότε η διαμόρφωση της καμπύλης $Q(t)$ θα συνεισφέρει μόνο η συνάρτηση βάσης για την οποία γνωρίζουμε ότι. Κατά συνέπεια η (8) απλοποιείται στο υποδιάστημα ως ακολούθως:

Για δεδομένη λοιπόν τιμή της παραμέτρου t , η επαναληπτική γραμμική παρεμβολή που περιγράφεται από την σχέση (7), αποδίδει στο τελευταίο επίπεδο d_{r-k+1} το σημείο της καμπύλης που αντιστοιχεί στην επιλεγείσα παραμετρική τιμή. Ο αλγόριθμος αυτός είναι γνωστός στη βιβλιογραφία ως αλγόριθμος De boor.

Εάν λάβουμε υπόψη ότι ο αλγόριθμος αυτός εφαρμόζεται πάντα για συγκεκριμένη τιμή της παραμέτρου t , άρα και για συγκεκριμένο υποδιάστημα $[t_{k-1}, t_k]$, προκύπτει άμεσα ότι στο δεξί μέλος της (6) θα συνεισφέρουν μόνο εκείνα από τα ενδιαμέσα σημεία de boor d_{r-k+1}^l για το οποία το διάστημα $[t_{k-1}, t_k]$ εμπεριέχεται στον φορέα της βάσης B_{k-1} . Η παρατήρηση αυτή οδηγεί στην παρακάτω τελική διατύπωση του αλγόριθμου De boor:

Γραφικά ο αλγόριθμος De boor αναπαρίσταται με το λεγόμενο αναδρομικό τρίγωνο De boor το οποίο φαίνεται στο σχήμα που ακολουθεί.



Σχήμα1: Αναδρομικό τρίγωνο De boor

2.3.Ιδιότητες καμπύλων B-Spline

Έστω $Q(t)$ μία καμπύλη B-spline τάξεως k (βαθμού $k-1$) η οποία έχει ορισθεί ως προς κομβοδιάνυσμα $T=\{t_0, t_1, \dots, t_{n+k}\}$. Τότε:

- I. Εάν ℓ κόμβοι του T συμπίπτουν, τότε η διαφορισμότητα της $Q(t)$ στον αντίστοιχο κόμβο μειώνεται σε $C^{k-1-\ell}$. Προφανώς, όταν το κομβοδιάνυσμα αποτελείται από απλούς κόμβους ($\ell=1$) τότε η λειότητα της $Q(t)$ είναι C^{k-2} . Εν γένει, είναι επιθυμητό οι ακραίοι κόμβοι να έχουν πολλαπλότητα k , το οποίο εξαναγκάζει τα σημεία De Boor d_0 και d_n να βρίσκονται στα άκρα της καμπύλης B-spline. Με τον τρόπο αυτό, ο σχεδιαστής διαθέτει καλύτερο έλεγχο ως προς την συμπεριφορά της καμπύλης B-spline στα άκρα. Εάν οι ακραίοι κόμβοι έχουν χαμηλότερη ή και απλή πολλαπλότητα τότε τα ακραία σημεία De Boor δεν ανήκουν στην καμπύλη και αποκαλούνται “phantom vertices”.
- II. Εάν $(k-1)$ κορυφές του πολυγώνου De Boor είναι συγγραμμικές, τότε η καμπύλη $Q(t)$ εφάπτεται στο πολύγωνο ελέγχου αυτής.
- III. Εάν (k) κορυφές του πολυγώνου De Boor είναι συγγραμμικές, τότε η καμπύλη $Q(t)$ και το πολύγωνο De Boor ταυτίζονται κατά μήκος ενός παραμετρικού διαστήματος.
- IV. Εάν $(k-1)$ κορυφές του πολυγώνου De Boor συμπίπτουν τότε η $Q(t)$ παρεμβάλλει την κοινή κορυφή και οι γειτονικές πλευρές του πολυγώνου εφάπτονται στην $Q(t)$. Η κοινή αυτή κορυφή μπορεί να είναι και ακμή.
- V. Η ιδιότητα της κυρτής θήκης (convex hull property) ισχύει για κάθε ομάδα k γειτονικών σημείων De Boor, δηλαδή κάθε πολυονυμικό τμήμα μιας καμπύλης B-spline ανήκει στην κυρτή θήκη των k γειτονικών σημείων De Boor που επηρεάζουν το εν λόγω τμήμα. Συνολικά μια καμπύλη B-spline εμπεριέχεται στην ένωση των κυρτών θηκών κάθε τμήματος.
- VI. Εάν η $Q(t)$ είναι επίπεδη καμπύλη, τότε ικανοποιεί την ιδιότητα της μειωμένης μεταβολής για κάθε ομάδα k -γειτονικών σημείων De Boor.

3.Η ΣΥΣΤΗΜΑΤΙΚΗ ΣΕΙΡΑ FORMDATA

Στο κεφάλαιο που ακολουθεί παρουσιάζεται η συστηματική σειρά FORMDATA, Πιο συγκεκριμένα παρουσιάζονται τα χαρακτηριστικά της, οι οικογένειες νομέων οι οποίες την απαρτίζουν, η ονοματολογία αυτών των οικογενειών, καθώς και η διαδικασία εφαρμογής της για την παραγωγή μιας γάστρας πλοίου

3.1.Χαρακτηριστικά της σειράς:

Η συστηματική σειρά FORMDATA αναπτύχθηκε από το Δανικό Πολυτεχνείο της Κοπεγχάγης και δίνει επαρκή στοιχεία για την χάραξη των ναυπηγικών γραμμών πλοίου. Η σειρά αυτή βασίζεται στην στατιστική ανάλυση των στοιχείων υπάρχοντων πλοίων διαφόρων τύπων και έχει χρησιμοποιηθεί ευρύτατα στην σχεδίαση πολλών σύγχρονων πλοίων.

Η συστηματική σειρά FORMDATA καλύπτει συνηθισμένες μορφές πλοίων με μέση τομή, η οποία έχει κατακόρυφες πλευρές. Η σειρά περιλαμβάνει τρεις βασικές μορφές γραμμών πλοίου: **U** (πλήρεις γραμμές σχήματος U), **N** (μη έντονες γραμμές νομέων) και **V** (λεπτές γραμμές σχήματος V) που σχηματίζονται με συνδυασμό δύο σειρών **A** και **F** πρυμναίου και πρωραίου τμήματος πλοίου αντίστοιχα. Οι μορφές των γραμμών μεταβάλλονται συστηματικά με το συντελεστή μέσης τομής C_M και με το συντελεστή γάστρας του πρυμναίου και πρωραίου τμήματος του πλοίου C_{BA} και C_{BF} αντίστοιχα.

Τα διαγράμματα της συστηματικής σειράς FORMDATA δίνουν τις καμπύλες εγκάρσιων τομών σε αδιάστατη μορφή και κατά κύριο λόγο περιλαμβάνουν καμπύλες που αντιπροσωπεύουν παραμετρικές οικογένειες νομέων (εγκάρσιες τομές) για το πρωραίο και το πρυμναίο ήμισυ του πλοίου χωριστά. Σε κάθε πλοίο αντιστοιχούν 11 θεωρητικοί νομείς που χωρίζουν το μήκος του σε 10 ισαπέχοντα διαστήματα με τον **σταθμό 0** να αντιστοιχεί στην πρυμναία κάθετο και τον **σταθμό 10** να αντιστοιχεί στην πρωραία κάθετο του πλοίου. Σε αυτούς προστίθενται άλλοι δύο νομείς στο μέσο του πρώτου και τελευταίου διαστήματος, χαρακτηριζόμενοι με τους αριθμούς $\frac{1}{2}$ και $9\frac{1}{2}$ αντίστοιχα, προκειμένου να περιγράψουν πιο έντονες μεταβολές στη μορφή των εγκάρσιων τομών για τα δύο ανωτέρω άκρα του πλοίου.

Οι παραπάνω σταθμοί συνδυάζονται με τα αντίστοιχα περιγράμματα πρύμνης και πλώρης για τη δημιουργία της εκάστοτε γάστρας. Η διαμόρφωση πρύμνης και πλώρης είναι δυνατή κατά συμβατικό τρόπο (μορφές **U, N, V**, σχήματα 1-2). Επίσης προβλέπεται ο τύπος βολβοειδούς πλώρας (σύμβολο **B**, σχήμα 3), πρύμνης άβακος (transom stern, σύμβολο **C**, σχήμα 4), καθώς και πρύμνης καταδρομικού (cruiser stern, σύμβολο **T**)

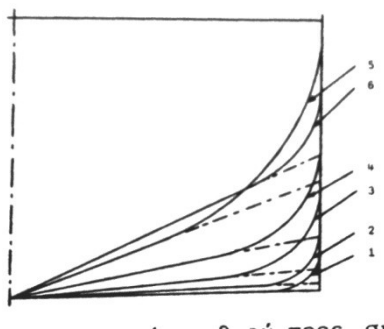
Κάθε οικογένεια από τις προαναφερθέντες καμπύλες χαρακτηρίζεται από τον συνδυασμό συμβόλων και αριθμών, αποτελούμενο από τρεις χαρακτήρες και ενίοτε από ένα δείκτη πχ. **U2F80**, **B₀1F90**, **C_A1A70**. Η ονοματολογία κάθε παραμετρικής οικογένειας νομέων της σειράς FORMDATA στηρίζεται στους ακόλουθους κανόνες:

Ο πρώτος χαρακτήρας είναι ένα γράμμα που προσδιορίζει την οικογένεια γραμμών στην οποία αναφερόμαστε:

- N:** πλοίο με «κανονικές» γραμμές
- U:** πλοίο με «πλήρεις» γραμμές
- V:** πλοίο με «λεπτές» γραμμές
- B:** πλοίο με βολβοειδή πλώρη
- T:** πλοίο με πρύμνη δεξαμενοπλοίου
- C:** πλοίο με πρύμνη άβακας

Ο δεύτερος χαρακτήρας είναι ένας αριθμός που προσδιορίζει το συντελεστή μέσης τομής C_M της δεδομένης οικογένειας νομέων. Στη FORMDATA υπάρχει η ακόλουθη αντιστοιχία μεταξύ αριθμών και συντελεστών μέσης τομής:

- 1** $\Rightarrow C_M=0.995$
- 2** $\Rightarrow C_M=0.980$
- 3** $\Rightarrow C_M=0.940$
- 4** $\Rightarrow C_M=0.880$
- 5 και 6** $\Rightarrow C_M=0.740$



Ο τρίτος χαρακτήρας είναι ένα γράμμα **A** ή **F** που προσδιορίζει αν μια συγκεκριμένη οικογένεια νομέων χαρακτηρίζει το πρωραίο ή το πρυμναίο τμήμα του πλοίου.

Οι τελευταίοι δύο χαρακτήρες αντιπροσωπεύουν ένα διψήφιο αριθμό που προσδιορίζει το συντελεστή γάστρας C_{BF} ή C_{BA} , του πρωραίου ή του πρυμναίου τμήματος του πλοίου αντίστοιχα.

Σε ορισμένες οικογένειες νομέων εμφανίζεται ένας δείκτης μετά τον πρώτο χαρακτήρα του ονόματος της οικογένειας στην οποία αναφερόμαστε:

Για την οικογένεια **B** εμφανίζονται οι αριθμοί **0,4,5,8,10** οι οποίοι συμβολίζουν το λόγο επί της εκατό της επιφάνειας του βολβού στην πρωραία κάθετο ως προς την επιφάνεια της μέσης τομής του πλοίου.

Για την οικογένεια **C** εμφανίζονται οι χαρακτήρες **A,B,C,D** οι οποίοι υποδηλώνουν τη σχετική κλίση της πρύμνης άβακας έναντι της κατακόρυφου θέσης (δείκτης **D**).

3.2. Παραδείγματα ονοματολογίας

Ακολουθούν ορισμένα χαρακτηριστικά παραδείγματα καθώς και πίνακας στον οποίο φαίνονται οι διαθέσιμες οικογένειες νομέων (πίνακας Α):

U3A70 : Μορφή εγκάρσιων τομών τύπου U

Συντελεστής μέσης τομής $C_M=0.940$

Πρυμναίοι νομείς

Συντελεστής γάστρας $C_{BA}=0.70$

N2F65 : Μορφή εγκάρσιων τομών τύπου N

Συντελεστής μέσης τομής $C_M=0.980$

Πρωραίοι νομείς

Συντελεστής γάστρας $C_{BF}=0.65$

V3A55 : Μορφή εγκάρσιων τομών τύπου V

Συντελεστής μέσης τομής $C_M=0.940$

Πρυμναίοι νομείς

Συντελεστής γάστρας $C_{BA}=0.55$

T1A85 : Μορφή εγκάρσιων τομών τύπου T

Συντελεστής μέσης τομής $C_M=0.995$

Πρυμναίοι νομείς

Συντελεστής γάστρας $C_{BA}=0.85$

B₁₀1F90 : Μορφή εγκάρσιων τομών τύπου B

Επί τοις εκατό λόγος της επιφάνειας του βολβού στην πρωραία κάθετο ως προς την επιφάνεια της μέσης τομής του πλοίου ίσος με 10

Συντελεστής μέσης τομής $C_M=0.995$

Πρωραίοι νομείς

Συντελεστής γάστρας $C_{BF}=0.90$

3.3.Πίνακας διαθέσιμων οικογενειών

	ΠΡΥΜΝΗ										ΠΛΩΡΗ													
	όνομα οικογένειας	C _{BA}										όνομα οικογένειας	C _{BF}											
NOMEIS ΤΥΠΟΥ U	U1A						70	75	80			U1F							70	75	80			
	U2A			55	60	65	70	75				U2F			55	60	65							
	U3A		50	55	60	65	70					U3F		50	55	60	65	70						
NOMEIS ΤΥΠΟΥ V	V2A				60	65	70					V2F			55	60	65							
	V3A		50	55	60	65	70					V3F		50	55	60	65	70						
NOMEIS ΤΥΠΟΥ N	N2A			55	60	65	70	75				N2F			55	60	65	70	75					
	N3A		50	55	60	65	70					N3F		50	55	60	65	70						
	N4A	45	50	55	60	65						N4F	45	50	55	60	65							
NOMEIS ΤΥΠΟΥ T	T1A						70	75	80	85														
NOMEIS ΤΥΠΟΥ B												B ₀ 1F							70	75	80	85	90	
												B ₅ 1F							70	75	80	85	90	
												B ₁₀ 1F							70	75	80	85	90	
												B ₀ 2F			55	60	65	70	75					
												B ₄ 2F			55	60	65	70	75					
												B ₈ 2F			55	60	65	70	75					
												B ₀ 3F		50	55	60	65	70						
												B ₄ 3F		50	55	60	65	70						
NOMEIS ΤΥΠΟΥ C	C _A 1A						70	75	80	85	90													
	C _B 1A						70	75	80	85	90													
	C _C 1A						70	75	80	85	90													
	C _D 1A						70	75	80	85	90													
	C _A 2A		50	55	60	65	70																	
	C _B 2A		50	55	60	65	70																	
	C _C 2A		50	55	60	65	70																	
	C _D 2A		50	55	60	65	70																	

Πίνακας Α : Διαθέσιμες οικογένειες νομέων πρύμνης και πλήρης της συστηματικής σειράς FORMDATA

3.4. Διαδικασία εφαρμογής της σειράς:

Από το πρώτο στάδιο της προμελέτης έχουν καθοριστεί τα παρακάτω χαρακτηριστικά του υπό σχεδίαση πλοίου:

- Το μήκος μεταξύ καθέτων του πλοίου L_{BP}
- Το πλάτος B
- Το κοίλο D
- Ο συντελεστής γάστρας C_B
- Ο συντελεστής μέσης τομής C_M
- Η διαμήκης απόσταση του κέντρου άντωσης από τη μέση τομή LCB , με θετική φορά πρύμνηθεν.

Επίσης έχουν καθοριστεί η μορφή της πλώρης, της πρύμνης και των νομέων με βάση τα υδροστατικά, υδροδυναμικά και κατασκευαστικά χαρακτηριστικά των διαφόρων οικογενειών νομέων. Αν πρόκειται να γίνει επιλογή μεταξύ των μορφών U, N και V ενδεικτικά αναφέρεται, ότι ανάλογα με το μέγεθός τους, για δεξαμενόπλοια και πλοία μεταφοράς φορτίων χύδην προτιμούνται οι μορφές U και N , ενώ για πλοία γενικού φορτίου καθώς και για επιβατηγά οι μορφές N και V . Επίσης συνίσταται για το προωαίο τμήμα του πλοίου να μην χρησιμοποιούνται οι εξαιρετικά έντονες μορφές U ή V .

Για την χάραξη των ναυπηγικών γραμμών του πλοίου με βάση τη συστηματική σειρά FORMDATA απαιτούνται, εκτός από τα παραπάνω στοιχεία και οι συντελεστές γάστρας C_{BA} και C_{BF} για το πρυμναίο και προωαίο τμήμα του πλοίου αντίστοιχα. Οι συντελεστές αυτοί υπολογίζονται σαν συνάρτηση του συντελεστή γάστρας του πλοίου C_B , του διαμήκους κέντρου άντωσης LCB και του μήκους μεταξύ καθέτων του πλοίου L_{BP} .

—)

—)

Με βάση τους συντελεστές C_{BA} και C_{BF} είναι δυνατή η επιλογή συνδυασμού της μορφής του τμήματος πλώρης και πρύμνης. Στους πίνακες Β, Γ και Δ που ακολουθούν εμφανίζονται ενδεικτικά στην πρώτη σειρά οι δυνατές μορφές πλώρης, στη δεύτερη οι αντίστοιχες τιμές C_{BF} ενώ στην πρώτη στήλη αναφέρονται οι δυνατές μορφές πρύμνης με τους αντίστοιχους συντελεστές C_{BA} . Οι αναφερόμενες τιμές του πίνακα, που διασταυρώνουν δυνατούς συνδυασμούς πλώρης-πρύμνης, αφορούν τα όρια μεταβολής του C_B (πρώτη σειρά) και του LCB ($\%L_{BP}$ -δεύτερη σειρά).

ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΟΙΚΟΓΕΝΕΙΑΣ ΝΟΜΕΡΝ		B ₀ 1F	B ₅ 1F	B ₁₀ 1F	B ₀ 2F	B ₄ 2F	B ₈ 2F	B ₀ 3F	B ₄ 3F	B ₈ 3F
C BA	C BF	0,70±0,90	0,70±0,90	0,70±0,90	0,50±0,75	0,50±0,75	0,50±0,75	0,50±0,70	0,50±0,70	0,50±0,70
		$\frac{0,70 \pm 0,85}{-4,50 \pm 3,52}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,52}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,52}$						
T1A	0,70±0,85	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,52}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,52}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,52}$						
U1A	0,70±0,80	$\frac{0,70 \pm 0,85}{-4,50 \pm 2,38}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 2,38}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 2,38}$					$\frac{C_B}{x_B \%L_{BP} }$	
U2A	0,55±0,75				$\frac{0,53 \pm 0,75}{-4,50 \pm 5,10}$	$\frac{0,53 \pm 0,75}{-4,50 \pm 5,10}$	$\frac{0,53 \pm 0,75}{-4,50 \pm 5,10}$			
N2A	0,55±0,75				$\frac{0,53 \pm 0,75}{-4,50 \pm 5,10}$	$\frac{0,53 \pm 0,75}{-4,50 \pm 5,10}$	$\frac{0,53 \pm 0,75}{-4,50 \pm 5,10}$			
V2A	0,60±0,70				$\frac{0,55 \pm 0,73}{-3,38 \pm 4,92}$	$\frac{0,55 \pm 0,73}{-3,38 \pm 4,92}$	$\frac{0,55 \pm 0,73}{-3,38 \pm 4,92}$			
U3A	0,50±0,70							$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$	$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$	$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$
N3A	0,50±0,70							$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$	$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$	$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$
V3A	0,50±0,70							$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$	$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$	$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$

Πίνακας Β: Συνδυασμοί πρύμνης καταδρομικού και βολβοειδούς πλήρης.

ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΟΙΚΟΓΕΝΕΙΑΣ ΝΟΜΕΡΝ		U1F	U2F	N2F	V2F	U3F	N3F	V3F	N4F
C BA	C BF	0,70±0,80	0,55±0,65	0,55±0,75	0,55±0,65	0,50±0,70	0,50±0,70	0,50±0,70	0,45±0,65
		$\frac{0,70 \pm 0,83}{-2,20 \pm 3,52}$							
T1A	0,70±0,85	$\frac{0,70 \pm 0,83}{-2,20 \pm 3,52}$							
U1A	0,70±0,80	$\frac{0,70 \pm 0,80}{-2,20 \pm 2,36}$						$\frac{C_B}{x_B \%L_{BP} }$	
U2A	0,55±0,75	$\frac{0,55 \pm 0,70}{-2,25 \pm 4,80}$							
N2A	0,55±0,75			$\frac{0,55 \pm 0,75}{-4,50 \pm 4,80}$					
V2A	0,60±0,70				$\frac{0,58 \pm 0,68}{-1,08 \pm 3,68}$				
U3A	0,50±0,70					$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$			
N3A	0,50±0,70					$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$			
V3A	0,50±0,70						$\frac{0,50 \pm 0,70}{-4,64 \pm 4,92}$		
N4A	0,45±0,65								$\frac{0,50 \pm 0,65}{-4,82 \pm 5,10}$

Πίνακας Γ: Συνδυασμοί πρύμνης καταδρομικού και μη βολβοειδούς πλήρης.

ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΟΙΚΟΓΕΝΕΙΑΣ NOMERN		B ₀ 1F	B ₅ 1F	B ₁₀ 1F	B ₀ 2F	B ₄ 2F	B ₈ 2F	U2F	N2F	V2F
$\begin{matrix} C \\ \swarrow \\ C_{BF} \\ \nwarrow \\ C_{BA} \end{matrix}$		0,70±0,90	0,70±0,90	0,70±0,90	0,50±0,75	0,50±0,75	0,50±0,73	0,55±0,65	0,55±0,75	0,55±0,65
C _A 1A	0,70±0,90	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$						
C _B 1A	0,70±0,90	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$				$\frac{C_B}{x_B \% L_{BP} }$		
C _C 1A	0,70±0,90	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$						
C _D 1A	0,70±0,90	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$	$\frac{0,70 \pm 0,85}{-4,50 \pm 3,50}$						
C _A 2A	0,50±0,70				$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,53 \pm 0,68}{-3,54 \pm 3,68}$	$\frac{0,53 \pm 0,68}{-5,10 \pm 3,68}$	$\frac{0,53 \pm 0,68}{-3,54 \pm 3,68}$
C _B 2A	0,50±0,70				$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,53 \pm 0,68}{-3,54 \pm 3,68}$	$\frac{0,53 \pm 0,68}{-5,10 \pm 3,68}$	$\frac{0,53 \pm 0,68}{-3,54 \pm 3,68}$
C _C 2A	0,50±0,70				$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,53 \pm 0,68}{-3,54 \pm 3,68}$	$\frac{0,53 \pm 0,73}{-5,10 \pm 3,68}$	$\frac{0,53 \pm 0,68}{-3,54 \pm 3,68}$
C _D 2A	0,50±0,70				$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,50 \pm 0,73}{-5,10 \pm 4,92}$	$\frac{0,53 \pm 0,68}{-3,54 \pm 3,68}$	$\frac{0,53 \pm 0,73}{-5,10 \pm 3,68}$	$\frac{0,53 \pm 0,68}{-3,54 \pm 3,68}$

Πίνακας Δ: Συνδυασμοί πρύμνης άβακας και βολβοειδούς και μη πλήρης.

4.ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΑ ΠΑΚΕΤΑ ΛΟΓΙΣΜΙΚΟΥ

Στο κεφάλαιο που ακολουθεί δίδεται μία σύντομη περιγραφή όλων των πακέτων λογισμικού που χρησιμοποιήθηκαν για την εκπόνηση της παρούσας διπλωματικής εργασίας. Ο συνδυασμός του σχεδιαστικού πακέτου Rhinoceros 3D με την οπτική γλώσσα προγραμματισμού Grasshopper, χρησιμοποιήθηκε για την κατασκευή και αποθήκευση των καμπύλων της συστηματικής σειράς FORMDATA. Η γλώσσα προγραμματισμού Python χρησιμοποιήθηκε για την σύνταξη του προγράμματος το οποίο επικοινωνεί με το CATIA, ένα άλλο σχεδιαστικό πακέτο, για την κατασκευή του τρισδιάστατου μοντέλου πλοίου.

4.1.Rhinoceros 3D

Το Rhinoceros είναι ένα αυτόνομο, εμπορικό εργαλείο τρισδιάστατης μοντελοποίησης της εταιρίας Robert McNeel and Associates, το οποίο χρησιμοποιεί καμπύλες NURBS (Non Uniform Rational B-Splines) και χρησιμοποιείται κυρίως για εφαρμογές βιομηχανικού σχεδιασμού, για ναυπηγικές, αρχιτεκτονικές, μηχανολογικές εφαρμογές, καθώς επίσης και για CAD/CAM εφαρμογές. Όπως και άλλα εργαλεία μοντελοποίησης έτσι και το Rhino διαθέτει μία δικιά του γλώσσα προγραμματισμού βασισμένη στην Visual Basic η οποία δίνει τη δυνατότητα στο σχεδιαστή να δημιουργήσει δικά του scripts. Το Rhinoceros έγινε ευρέως γνωστό στο χώρο του αρχιτεκτονικού σχεδιασμού λόγω ενός plug-in του (Grasshopper) το οποίο δίνει τη δυνατότητα της παραμετρικής σχεδίασης στο χρήστη όπου και θα αναφερθούμε εκτενέστερα παρακάτω. Η αυξανόμενη «δημοτικότητα» του συγκεκριμένου πακέτου εφαρμογών οφείλεται κυρίως στο σχετικά χαμηλό του κόστος αλλά και στο γεγονός ότι δίνει τη δυνατότητα στο χρήστη να εισάγει ή να εξάγει πάνω από 30 διαφορετικών τύπων αρχεία. Λόγω αυτού χρησιμοποιείται πολύ συχνά ως εργαλείο μετατροπής αρχείων από έναν τύπο σε άλλο.

4.2.Grasshopper

Το Grasshopper είναι μια οπτική γλώσσα προγραμματισμού η οποία αναπτύχθηκε από τον David Rutten της εταιρίας Robert McNeel and Associates και τρέχει μέσα στο περιβάλλον του πακέτου Rhinoceros 3D της ίδιας εταιρίας. Τα προγράμματα του grasshopper δημιουργούνται σύροντας στοιχεία (components) πάνω σε έναν καμβά. Κάθε στοιχείο δέχεται συγκεκριμένες εισόδους (inputs) και παράγει συγκεκριμένα αποτελέσματα (outputs) τα οποία αποτελούν inputs για άλλα components και ούτω καθεξής. Πολλά από αυτά τα components παράγουν 3D γεωμετρία εντός του Rhinoceros. Τα προγράμματα μπορούν επίσης να περιέχουν και άλλα ειδή αλγορίθμων, όπως αριθμητικές εφαρμογές ή εφαρμογές κειμένου ή ήχου –εικόνας. Το grasshopper έγινε γνωστό κυρίως γιατί προσφέρει τη δυνατότητα στον χρήστη να επεξεργαστεί με έναν «έξυπνο» τρόπο τα σχέδιά του (designs) χωρίς να είναι απαραίτητη η γνώση κάποιας γλώσσας προγραμματισμού (scripting language).

4.3.CATIA

Συνήθως αναφέρεται ως μία σουίτα λογισμικού για την διαχείριση του κύκλου ζωής τρισδιάστατων προϊόντων. Υποστηρίζει πολλαπλά στάδια της ανάπτυξης ενός προϊόντος, από την σύλληψη και τη σχεδίαση μέχρι την μοντελοποίηση και την κατασκευή του. Αποτελούμενο από πληθώρα διαφορετικών «υποπρογραμμάτων» (workbenches) ουσιαστικά παρέχει στο σχεδιαστή διαφορετικά περιβάλλοντα και εργαλεία ανάλογα με το τι θέλει να σχεδιάσει ή να πραγματοποιήσει. Μεταξύ αυτών παρέχει εργαλεία σχεδίασης, αντίστροφης μηχανικής και οπτικοποίησης για τη δημιουργία και την τροποποίηση σύνθετων, καθώς και καινοτόμων σχημάτων. Βασικό χαρακτηριστικό του είναι το ότι παρέχει στο χρήστη τη δυνατότητα εισαγωγής και συσχέτισης παραμέτρων κατά τη διαδικασία της σχεδίασης. Το workbench που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία είναι το Generative Shape Design (GSD). Το συγκεκριμένο workbench παρέχει στο σχεδιαστή εργαλεία τα οποία αφορούν κυρίως τη δημιουργία, την επεξεργασία και την ανάλυση πολύπλοκων σχημάτων και επιφανειών.

4.4.Python

Η Python είναι μία γενικής χρήσης, υψηλού επιπέδου, αντικειμενοστραφής γλώσσα προγραμματισμού η οποία έχει σχεδιαστεί με έμφαση στην αναγνωσιμότητα του κώδικα. Οι δημιουργοί της ισχυρίζονται ότι προσφέρει πάρα πολλές δυνατότητες υψηλού επιπέδου προγραμματισμού με πάρα πολύ απλό και σαφές συντακτικό. Επίσης η πρότυπη βιβλιοθήκη της είναι πολύ μεγάλη και ολοκληρωμένη. Επιπλέον το γεγονός ότι επιτρέπει στο χρήστη να οριοθετεί τα block εντολών με πάρα πολύ απλό τρόπο την κάνει να ξεχωρίζει από τις υπόλοιπες δημοφιλείς γλώσσες προγραμματισμού. Η Python υποστηρίζει πολλαπλά πρότυπα προγραμματισμού με κύριο τον αντικειμενοστραφή προγραμματισμό χωρίς όμως να περιορίζεται μόνο σε αυτό. Διαθέτει ένα πλήρως δυναμικό σύστημα εισαγωγής κειμένου και αυτόματη διαχείριση μνήμης παρόμοια με εκείνη των Scheme, Ruby, Perl και Tcl. Όπως άλλες δυναμικές γλώσσες προγραμματισμού έτσι και η Python χρησιμοποιείται συχνά ως scripting language, αλλά χρησιμοποιείται επίσης και σε ένα ευρύ φάσμα non scripting περιβαλλόντων. Χρησιμοποιώντας εργαλεία άλλων κατασκευαστών, κώδικας γραμμένος σε python είναι δυνατό να «συσκευαστεί» σε αυτόνομα εκτελέσιμα προγράμματα. Τέλος υπάρχουν διαθέσιμοι interpreters της python για διάφορα λειτουργικά συστήματα.

5.ΚΑΤΑΣΚΕΥΗ ΤΡΙΣΔΙΑΣΤΑΤΩΝ ΚΑΜΠΥΛΩΝ

Στο κεφάλαιο που ακολουθεί παρουσιάζεται η διαδικασία παραγωγής των τρισδιάστατων καμπύλων (νομέων) της συστηματικής σειράς FORMDATA. Πιο συγκεκριμένα παρουσιάζεται η επεξεργασία των δεδομένων εισόδου, η κατασκευή των καμπύλων με χρήση του Rhinoceros 3D και του Grasshopper και τέλος η αποθήκευσή τους.

5.1.Δεδομένα Εισόδου (Αρχεία)

Τα δεδομένα τα οποία χρησιμοποιήθηκαν για την κατασκευή των νομέων της συστηματικής σειράς FORMDATA ήταν αποτέλεσμα προηγούμενης διπλωματικής εργασίας του εργαστηρίου μελέτης πλοίου, εκπονημένη από τον Σαραϊδάρη Ιωάννη με τίτλο «ΕΞΟΜΑΛΥΝΣΗ ΣΥΣΤΗΜΑΤΙΚΗΣ ΣΕΙΡΑΣ FORMDATA».

Συγκεκριμένα για κάθε μία οικογένεια, πλήρης ή πρόμνης, της συστηματικής σειράς είχαμε στη διάθεσή μας έναν φάκελο ο οποίος περιείχε δύο αρχεία. Το πρώτο με όνομα “DEBOOR.DB” περιείχε τις συντεταγμένες των κορυφών, των χαρακτηριστικών πολυγώνων, των νομέων που απαρτίζουν στην εκάστοτε οικογένεια και το δεύτερο με όνομα “KNOT.KV” περιείχε τα αντίστοιχα κομβοδιανύσματα αυτών των νομέων. Στις εικόνες 1 και 2 που ακολουθούν παρουσιάζονται τα περιεχόμενα των παραπάνω αρχείων. Έτσι για παράδειγμα είχαμε έναν φάκελο με όνομα “B01F90” ο οποίος περιείχε τα αρχεία “DEBOOR.DB” και “KNOT.KV”.

Φανερά προέκυψε το πρόβλημα ότι όλοι οι φάκελοι περιείχαν δύο αρχεία τα οποία σε κάθε φάκελο είχαν το ίδιο όνομα. Στη συνέχεια με τη βοήθεια της γλώσσας προγραμματισμού rython 3.0 συντάχθηκε ένα πρόγραμμα το οποίο άλλαζε το όνομα των δύο αυτών αρχείων, προσθέτοντας στην αρχή το όνομα του φακέλου μέσα στον οποίο βρίσκονταν. Έτσι πλέον τα αρχεία ήταν της μορφής “B01F90_DEBOOR.DB”, και πλέον ήταν εύκολο να επεξεργαστούν όλα μαζί εκτός των φακέλων στα οποία βρίσκονταν αρχικά.

Για την κατασκευή κάθε νομέα χρειάζεται το χαρακτηριστικό του πολύγωνα και το κομβοδιάνυσμα. Τα αρχεία τα οποία αναφέρθηκαν παραπάνω περιέχουν πληροφορίες για περισσότερους από έναν νομείς. Συνεπώς ήταν αναγκαίο να διασπαστούν τα αρχικά αρχεία σε περισσότερα τα οποία περιείχαν πληροφορίες για έναν νομέα μόνο το καθένα. Πάλι με τη χρήση της γλώσσας rython 3.0 συντάχθηκε ένα πρόγραμμα το οποίο διάβαζε τα αρχικά αρχεία, όπως πχ “B01F90_DEBOOR.DB” και “B01F90_KNOT.KV” και δημιουργούσε τα παρακάτω αρχεία:

B01F90_DEBOOR_6.000.DB	B01F90_KNOT_6.000.KV
B01F90_DEBOOR_7.000.DB	B01F90_KNOT_7.000.KV
B01F90_DEBOOR_8.000.DB	B01F90_KNOT_8.000.KV
B01F90_DEBOOR_9.000.DB	B01F90_KNOT_9.000.KV
B01F90_DEBOOR_9.500.DB	B01F90_KNOT_9.500.KV
B01F90_DEBOOR_10.00.DB	B01F90_KNOT_10.00.KV

Πλέον δημιουργήθηκαν δύο φάκελοι, ο ένας (Control Points) περιείχε τα αρχεία με τις συντεταγμένες των σημείων ελέγχου του κάθε νομέα ξεχωριστά και ο άλλος (Knot Vectors) τα αντίστοιχα κομβοδιανύσματα. Τα παραπάνω αρχεία θα χρησιμοποιηθούν με τρόπο που περιγράφεται παρακάτω για τη δημιουργία των νομέων της συστηματικής σειράς FORMDATA, με τη βοήθεια του Rhinoceros 3D και του Grasshopper.

6.000	0.023741690	0.000000000	0.	9.000	0.910424310	0.060033450	1.
6.000	0.444751779	0.000000000	0.	9.000	0.931281630	0.096598960	1.
6.000	0.865761867	0.000000000	1.	9.000	0.960588820	0.259638670	1.
6.000	0.880000000	0.000000000	1.	9.000	0.969958000	0.527651270	1.
6.000	0.894238130	0.000000000	1.	9.000	0.997125910	1.305640590	1.
6.000	0.927635880	0.002921920	1.	9.000	1.000000000	1.388230220	1.
6.000	0.976138180	0.035020250	1.	9.000	1.000000000	1.465204000	1.
6.000	0.997466010	0.092077020	1.	9.000	1.000000000	1.542177777	1.
6.000	1.000000000	0.121040620	1.	9.000	1.000000000	1.571088889	0.
6.000	1.000000000	0.136617100	1.	9.000	1.000000000	1.600000000	0.
6.000	1.000000000	0.152193577	1.	9.500	0.023741690	0.000000000	0.
6.000	1.000000000	0.876096789	0.	9.500	0.063939260	0.000000000	0.
6.000	1.000000000	1.600000000	0.	9.500	0.104136830	0.000000000	1.
7.000	0.023741690	0.000000000	0.	9.500	0.146603800	0.000000000	1.
7.000	0.444751779	0.000000000	0.	9.500	0.189070770	0.000000000	1.
7.000	0.865761867	0.000000000	1.	9.500	0.253046510	0.005597150	1.
7.000	0.880000000	0.000000000	1.	9.500	0.328955040	0.032631820	1.
7.000	0.894238130	0.000000000	1.	9.500	0.408315640	0.096254150	1.
7.000	0.927635880	0.002921920	1.	9.500	0.457062440	0.153940090	1.
7.000	0.976138180	0.035020250	1.	9.500	0.496798860	0.224389490	1.
7.000	0.997466010	0.092077020	1.	9.500	0.527711200	0.303219500	1.
7.000	1.000000000	0.121040620	1.	9.500	0.551909810	0.391459240	1.
7.000	1.000000000	0.136617100	1.	9.500	0.569713020	0.487260090	1.
7.000	1.000000000	0.152193577	1.	9.500	0.584543420	0.592451980	1.
7.000	1.000000000	0.876096789	0.	9.500	0.597119670	0.691389830	1.
7.000	1.000000000	1.600000000	0.	9.500	0.610344500	0.795516210	1.
8.000	0.023741690	0.000000000	0.	9.500	0.620217530	0.893764500	-1.
8.000	0.440623332	0.000000000	0.	9.500	0.630595310	0.963859870	-1.
8.000	0.857504974	0.000000000	1.	9.500	0.641294740	1.028568280	-1.
8.000	0.870754800	0.000000000	1.	9.500	0.653878250	1.097838200	-1.
8.000	0.884004630	0.000000000	1.	9.500	0.674077590	1.194039410	-1.
8.000	0.953816680	0.011057140	1.	9.500	0.697647400	1.297496460	-1.
8.000	0.988942860	0.067556280	1.	9.500	0.720754860	1.394117250	-1.
8.000	1.000000000	0.137368340	1.	9.500	0.746226050	1.497837200	-1.
8.000	1.000000000	0.152193300	1.	9.500	0.762687430	1.563900280	-1.
8.000	1.000000000	0.167018259	1.	9.500	0.771698100	1.600000000	0.
8.000	1.000000000	0.883509130	0.	10.000	0.023741690	1.000000000	0.
8.000	1.000000000	1.600000000	0.	10.000	0.026683640	1.026380640	-1.
9.000	0.023741690	0.000000000	0.	10.000	0.037133960	1.116266030	-1.
9.000	0.369877689	0.000000000	0.	10.000	0.050097350	1.204957100	-1.
9.000	0.716013688	0.000000000	1.	10.000	0.077323010	1.299847480	-1.
9.000	0.739811300	0.000000000	1.	10.000	0.113427370	1.394078370	-1.
9.000	0.763608910	0.000000000	1.	10.000	0.168505990	1.495113170	-1.
9.000	0.811665970	0.004204450	1.	10.000	0.209501830	1.564001180	-1.
9.000	0.864416900	0.021561130	1.	10.000	0.231132100	1.600000000	0.

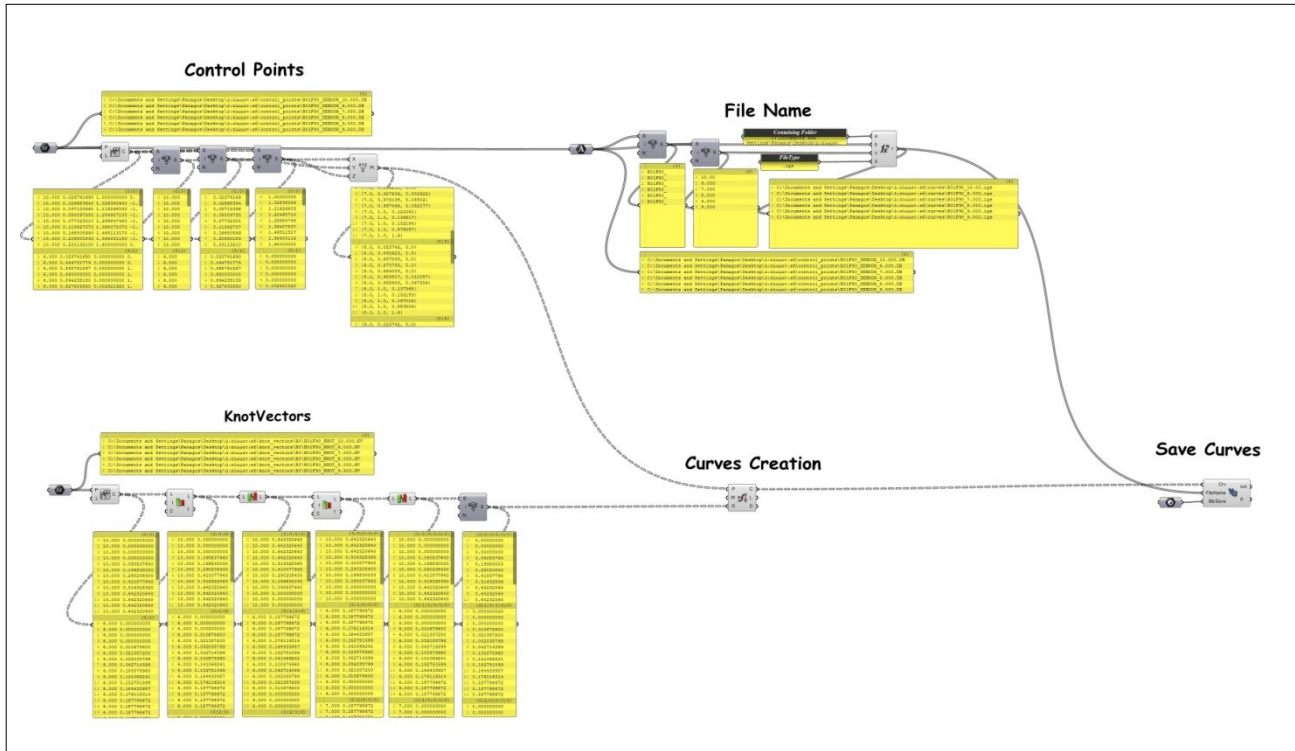
Εικόνα 1: Το αρχείο DEBOOR.DB της οικογένειας B01F90

6.000	0.000000000	8.000	0.013687370	9.500	0.063700455
6.000	0.000000000	8.000	0.027374740	9.500	0.095550683
6.000	0.000000000	8.000	0.041062109	9.500	0.127400910
6.000	0.000000000	8.000	0.054749479	9.500	0.192660340
6.000	0.010678600	8.000	0.116571949	9.500	0.266107340
6.000	0.021357200	8.000	0.140440668	9.500	0.354184510
6.000	0.032035799	8.000	0.164309387	9.500	0.448129000
6.000	0.042714399	8.000	0.188178106	9.500	0.525948660
6.000	0.100575960	8.000	0.212046825	9.500	0.615527460
6.000	0.141069241	8.000	0.212046825	9.500	0.722827580
6.000	0.152751599	8.000	0.212046825	9.500	0.813938660
6.000	0.164433957	8.000	0.212046825	9.500	0.929011430
6.000	0.176116314	9.000	0.000000000	9.500	1.018149150
6.000	0.187798672	9.000	0.000000000	9.500	1.126651090
6.000	0.187798672	9.000	0.000000000	9.500	1.225543910
6.000	0.187798672	9.000	0.000000000	9.500	1.231214730
6.000	0.187798672	9.000	0.017848209	9.500	1.324022490
7.000	0.000000000	9.000	0.035696418	9.500	1.437512850
7.000	0.000000000	9.000	0.053544627	9.500	1.527103950
7.000	0.000000000	9.000	0.071392836	9.500	1.642913930
7.000	0.000000000	9.000	0.144721870	9.500	1.735197610
7.000	0.010678600	9.000	0.209722140	9.500	1.846812200
7.000	0.021357200	9.000	0.307407530	9.500	1.846812200
7.000	0.032035799	9.000	0.402057070	9.500	1.846812200
7.000	0.042714399	9.000	1.325254300	9.500	1.846812200
7.000	0.100575960	9.000	1.374838284	10.000	0.000000000
7.000	0.141069241	9.000	1.424422268	10.000	0.000000000
7.000	0.152751599	9.000	1.474006251	10.000	0.000000000
7.000	0.164433957	9.000	1.523590235	10.000	0.000000000
7.000	0.176116314	9.000	1.573174219	10.000	0.090837640
7.000	0.187798672	9.000	1.573174219	10.000	0.199830030
7.000	0.187798672	9.000	1.573174219	10.000	0.290208400
7.000	0.187798672	9.000	1.573174219	10.000	0.410077940
7.000	0.187798672	9.500	0.000000000	10.000	0.516328360
8.000	0.000000000	9.500	0.000000000	10.000	0.642320640
8.000	0.000000000	9.500	0.000000000	10.000	0.642320640
8.000	0.000000000	9.500	0.000000000	10.000	0.642320640
8.000	0.000000000	9.500	0.031850228	10.000	0.642320640

Εικόνα 2: Το αρχείο KNOT.KV της οικογένειας B01F90

5.2.Κατασκευή και αποθήκευση των καμπύλων

5.2.1.Περιγραφή κατασκευής καμπύλων στο Grasshopper

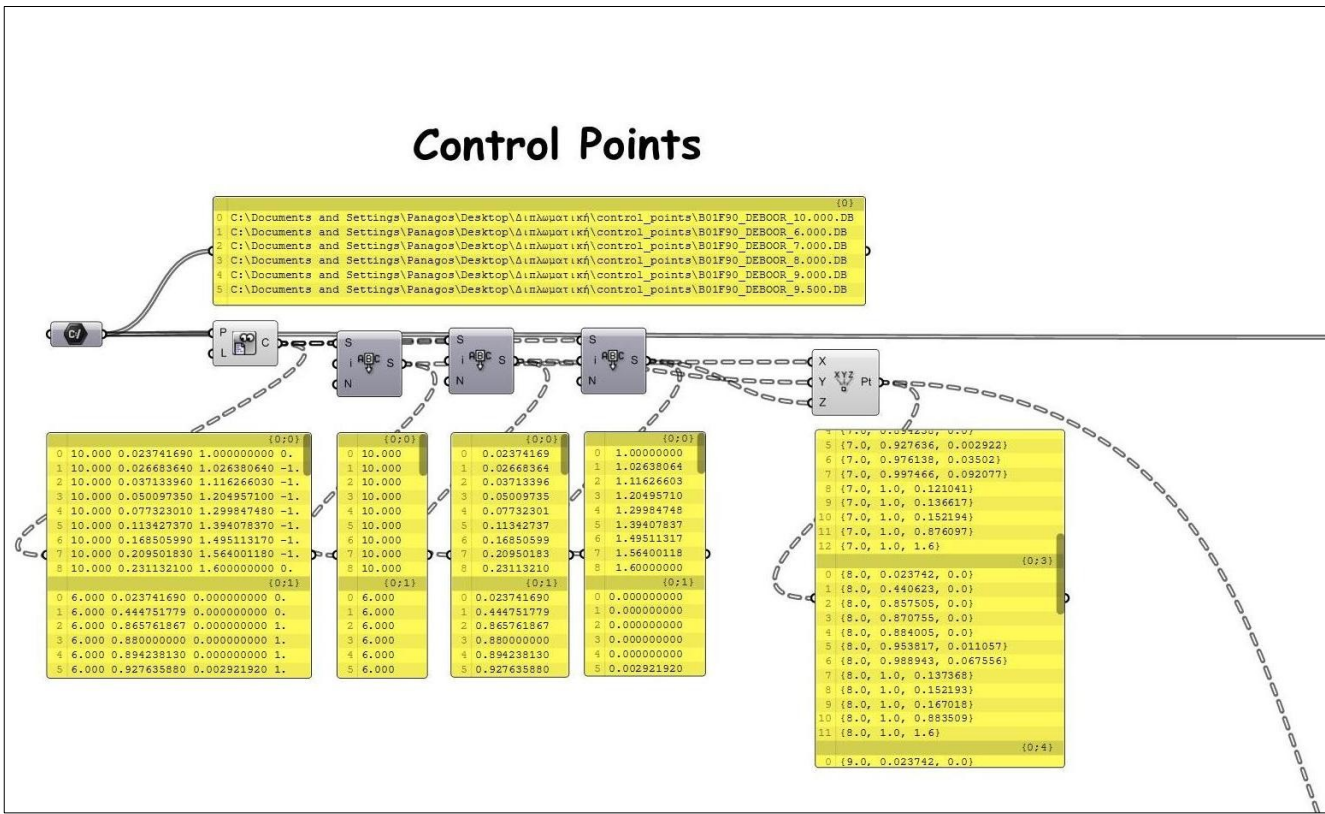


Εικόνα 3: Συνολική επισκόπηση του καμβά στο Grasshopper.

Στην παραπάνω εικόνα παρουσιάζεται στο σύνολό του ο καμβάς του Grasshopper, μαζί με τα components που χρησιμοποιήθηκαν για την κατασκευή και αποθήκευση των καμπύλων των νομέων της συστηματικής σειράς FORMDATA. Στην παράγραφο 5.2.2 που ακολουθεί περιγράφονται αναλυτικά όλα τα components που χρησιμοποιήθηκαν. Στον καμβά παρατηρούμε την ύπαρξη πέντε block από components, των οποίων ακολουθεί περιγραφή παρακάτω.

1. Control Points
2. Knot Vectors
3. Curves Creation
4. File Name Creation
5. Save Curves

5.2.1.A.Επεξεργασία των σημείων ελέγχου



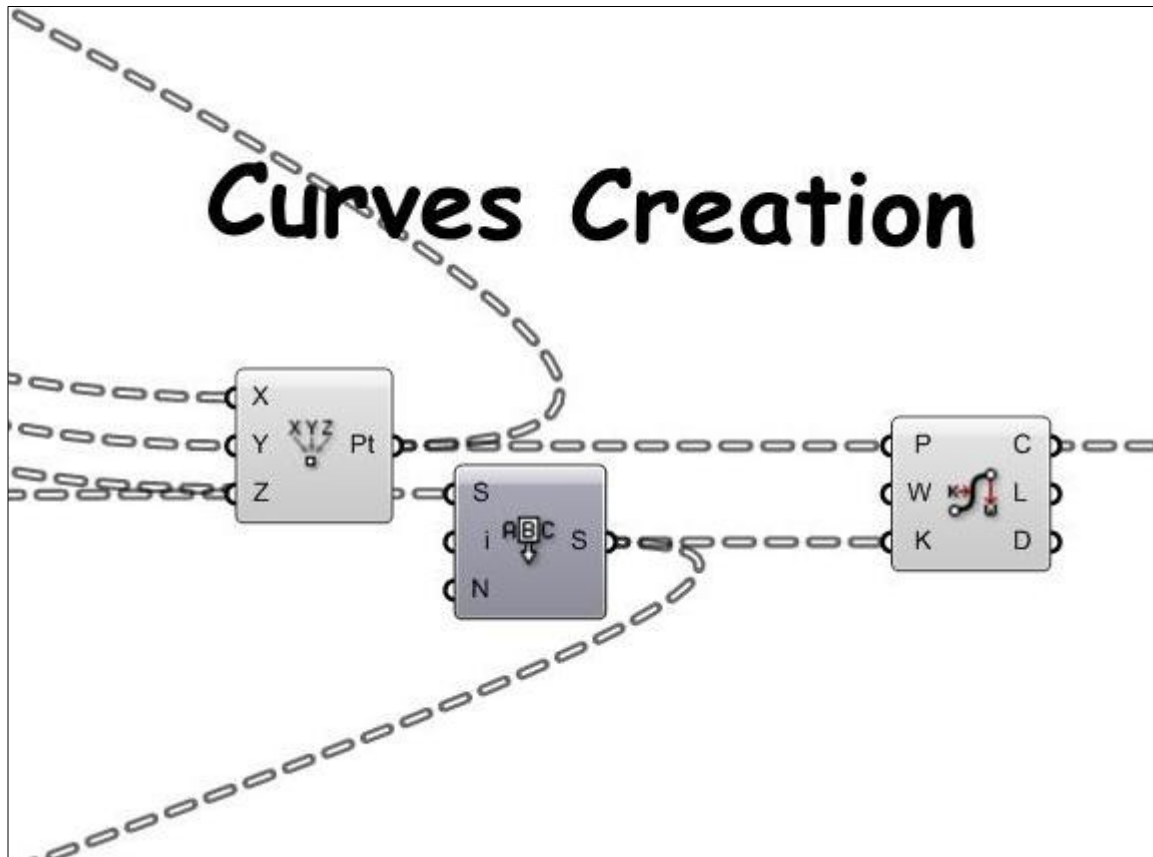
Εικόνα 4: Επεξεργασία των σημείων ελέγχου

Στο block, control points, χρησιμοποιήθηκαν τα εξής components: File Path, Read File, Sub String, Point XYZ, όπως φαίνεται στην εικόνα 4.

Αρχικά διαβάζουμε τα αρχεία που περιέχουν τα σημεία ελέγχου με το component Read File, το οποίο δέχεται ως input ένα file path. Το πρόγραμμα διαβάζει το κάθε αρχείο ως μία λίστα από strings. Κάθε γραμμή αναπαριστά ένα control point και αποτελείται από ένα string που περιέχει τις τρεις συντεταγμένες του. Χρησιμοποιώντας το component SubString χωρίζουμε τις τρεις συντεταγμένες του κάθε σημείου έλεγχου. Έτσι έχουμε πλέον τρεις λίστες, μια για κάθε μία συντεταγμένη των σημείων, οι οποίες θα αποτελέσουν δεδομένα εισόδου για το component PointXYZ, το οποίο δημιουργεί τα σημεία ελέγχου της κάθε εκάστοτε καμπύλης.

Παρέχοντας ως input περισσότερα file paths η διαδικασία εφαρμόζεται αυτόματα σε όλα, αφού το grasshopper διακριτοποιεί μόνο του τα δεδομένα δημιουργώντας λίστες και υπολίστες.

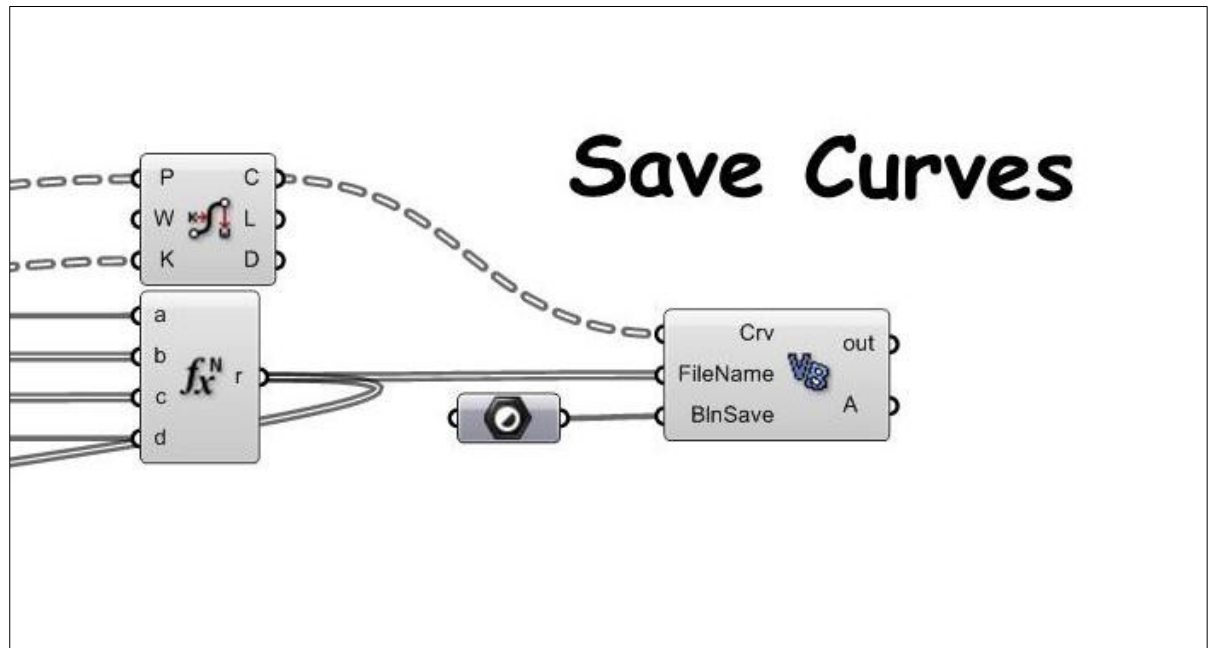
5.2.1.C.Κατασκευή των καμπύλων



Εικόνα 6: Κατασκευή των καμπύλων

Χρησιμοποιώντας το component Nurbs Curve (Εικόνα 6) το οποίο δέχεται ως input τα σημεία ελέγχου και τα αντίστοιχα κομβοδιανύσματα, στον οποίων τη δημιουργία αναφερθήκαμε στις παραγράφους 5.2.2.A και 5.2.2.B αντίστοιχα, κατασκευάζουμε τις καμπύλες.

5.2.1.E.Αποθήκευση των καμπύλων



Εικόνα 8: Αποθήκευση των καμπύλων

Για την αποθήκευση των καμπύλων χρησιμοποιήθηκε το component VB.NET Scripting, στο οποίο ορίσαμε τρία inputs, τις καμπύλες το filename το οποίο δημιουργήσαμε όπως περιγράφηκε νωρίτερα και μια λογική μεταβλητή (BlnSave) η οποία αναλόγως αν είναι αληθής η ψευδής ενεργοποιεί το εν λόγω component.

5.2.2.Grasshopper Components index

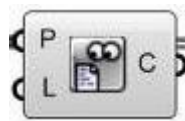
1) File Path:



Περιγραφή:

Represents a collection of file paths.

2) Read File:



Περιγραφή:

Read the contents of a file.

Δεδομένα εισόδου(input parameters):

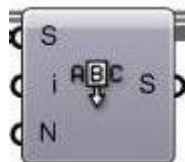
P(String): File paths to monitor.

L(Boolean): Read individual lines.

Αποτελέσματα(output parameters):

C(Generic Data): File content.

3) Sub String:



Περιγραφή:

Extract a portion of a string

Δεδομένα εισόδου(input parameters):

S(String): String to operate on.

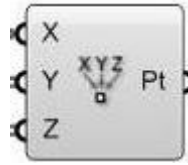
i(Integer): Zero based index of first character to copy.

N(Integer): Optional number of characters to copy. If blank the entire remainder will be copied.

Αποτελέσματα(output parameters):

S(String): The resulting sub string.

4) Point XYZ:



Περιγραφή:

Create a point from X,Y,Z coordinates

Δεδομένα εισόδου(input parameters):

X(number): {x} component.

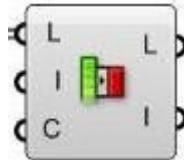
Y(number): {y} component.

Z(number): {z} component.

Αποτελέσματα(output parameters):

Pt(point): Point coordinate.

5) Sub List:



Περιγραφή:

Extract a subset from a list. Elements in a list are identified by their index. The first element is stored at index zero, the second is stored at index one and so on and so forth. The subset allows you to copy a continuous range of elements as a new list.

Δεδομένα εισόδου(input parameters):

L(Generic Data): Base List.

I(Domain): Interval of indices to copy.

C(Boolean): Remap indices that overshoot list domain.

Αποτελέσματα(output parameters):

L(Generic Data): Subset of base list.

I(Integer): Indices of subset items.

6) Reverse List:



Περιγραφή:

Reverse the order of a list. The new index of each element will be $N-i$, where N is the highest index on the list and I is the old index of the element.

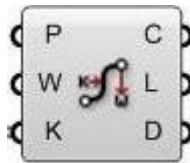
Δεδομένα εισόδου(input parameters):

L(Generic Data): Base list.

Αποτελέσματα(output parameters):

L(Generic Data): Reversed list.

7) Nurbs Curve:



Περιγραφή:

Create o nurbs curve from control points, weights and knots

Δεδομένα εισόδου(input parameters):

P(point): Curve control points.

W(Number): Optional curve control weights.

K(number): Nurbs curve knot vector.

Αποτελέσματα(output parameters):

C(Curve): Resulting nurbs curve.

L(Number): Curve length.

D(Domain): Curve domain.

8) String Parameter:



Περιγραφή:

Represents a collection of strings. String parameters are capable of storing persistent data. You can set the persistent records through the parameter menu.

9) Variable Expression:



Περιγραφή:

Expression component with a variable amount of input parameters

Δεδομένα εισόδου(input parameters):

a(Expression Variant): Expression input variable a.

b(Expression Variant): Expression input variable b.

c(Expression Variant): Expression input variable c.

d(Expression Variant): Expression input variable d.

Αποτελέσματα(output parameters):

r(Generic Data): Expression result.

10) VB.NET Scripting:



Περιγραφή:

This component attempts to compile and run user specified VB.NET code. By default the component contains no code, there are two input parameters (x,y) and one output parameter (A) all of which are of type System. Object. Input and output parameters can be added, deleted and renamed through the component menu.

6.ΚΑΤΑΣΚΕΥΗ ΤΟΥ ΤΡΙΣΔΙΑΣΤΑΤΟΥ ΜΟΝΤΕΛΟΥ

Στο παρόν κεφάλαιο περιγράφεται αναλυτικά όλη η διαδικασία κατασκευής του τρισδιάστατου μοντέλου του πλοίου. Πιο συγκεκριμένα περιγράφεται η δημιουργία μιας βιβλιοθήκης μέσα στο CATIA η οποία περιέχει όλες τις οικογένειες νομέων αλλά και τα προφίλ πλώρης και πρύμνης, η κατασκευή μιας σειράς βοηθητικών γεωμετριών (καμπύλες Flat of Side και Flat of Bottom, το περίγραμμα του κυρίου καταστρώματος, η Center Line) και τέλος η κατασκευή των επί μέρους επιφανειών που απαρτίζουν τη γάστρα του τρισδιάστατου μοντέλου. Αξίζει να σημειωθεί ότι παρουσιάζονται όλες οι κατασκευές που πραγματοποιήθηκαν για όλες τις διαφορετικές οικογένειες πλώρης και πρύμνης (βολβοειδής ή μη πλώρη, πρύμνη καταδρομικού ή πρύμνη άβακας).

6.1.Δημιουργία βιβλιοθήκης καμπύλων στο CATIA

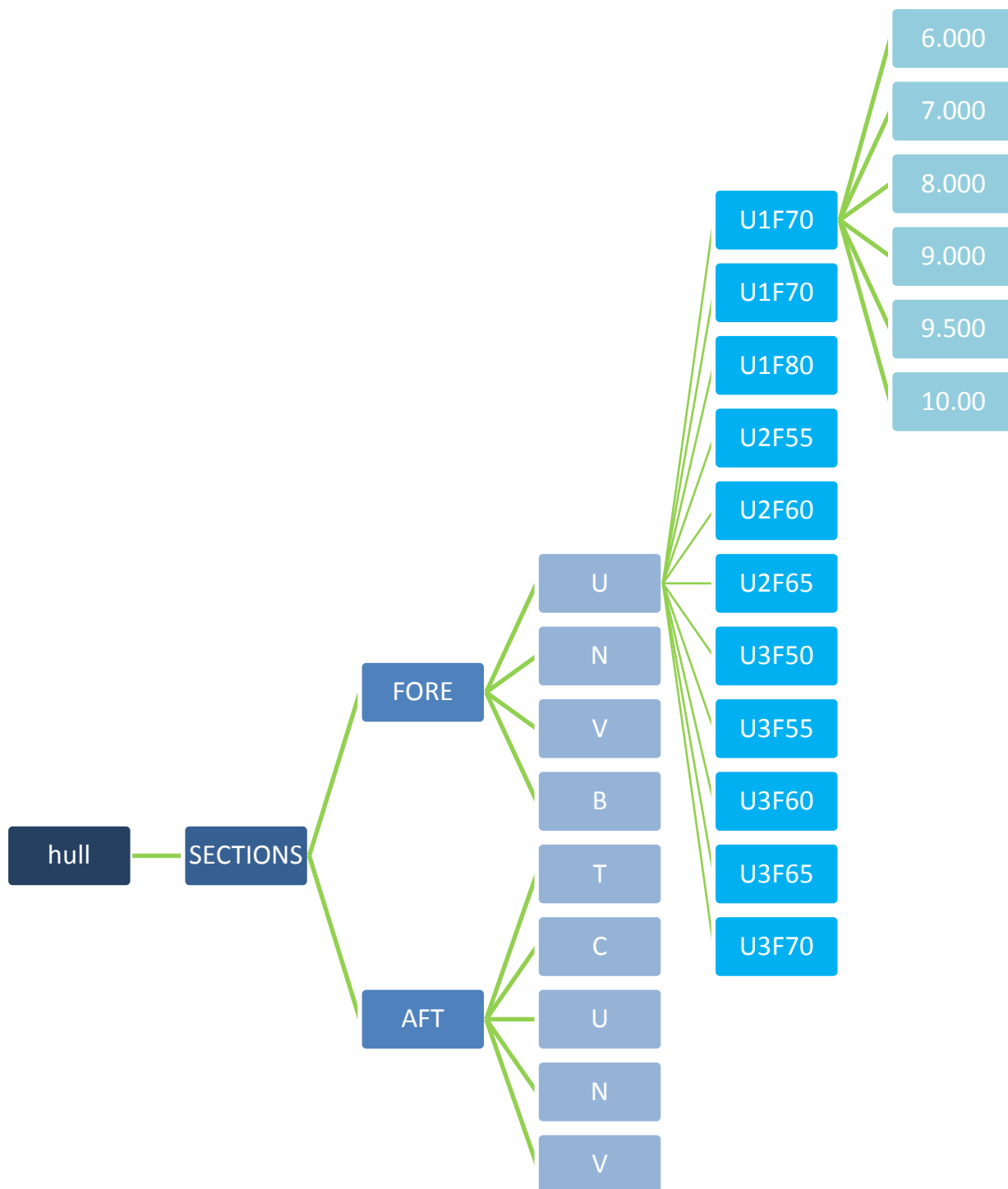
Αφού δημιουργήθηκαν όλες τις καμπύλες (νομείς) της συστηματικής σειράς FORMDATA με τον τρόπο που περιγράφηκε σε προηγούμενο κεφάλαιο, ακολουθεί η κατασκευή μίας «βιβλιοθήκης» εντός του CATIA η οποία θα περιέχει το σύνολο των καμπύλων αρχειοθετημένων ανά οικογένεια πλώρης ή πρύμνης.

Εντός του περιβάλλοντος του CATIA και συγκεκριμένα του, χρησιμοποιούμενου στην περίπτωση μας, workbench Generative Shape Design (GSD) υπάρχουν δύο τρόποι για την οργάνωση των δημιουργηθέντων γεωμετρικών στοιχείων, η χρήση geometric set και η χρήση ordered geometric set. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε η οργάνωση με χρήση απλών και όχι ordered geometric set.

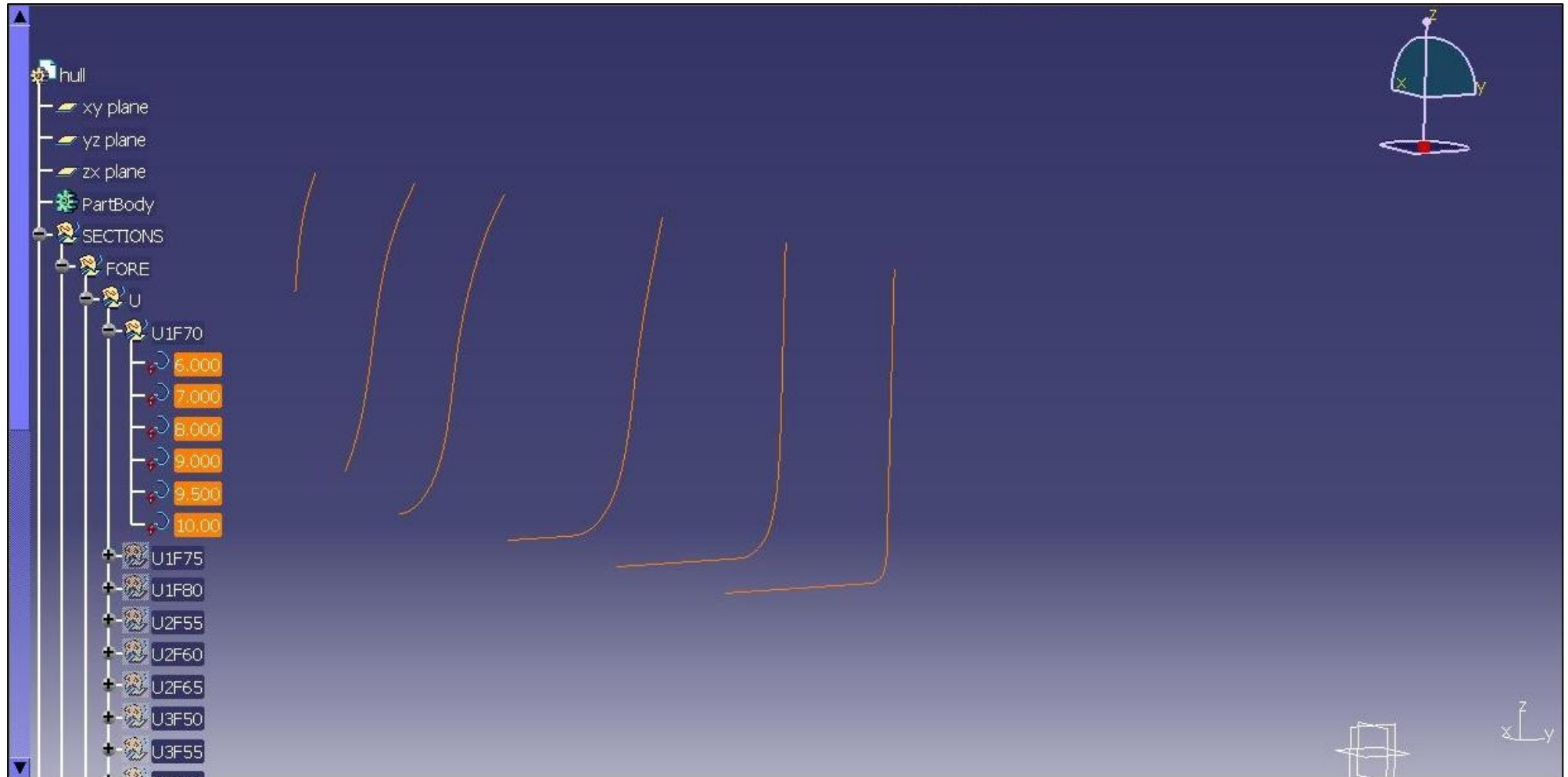
Εντός του object hull δημιουργούμε ένα geometric set το οποίο θα περιέχει τις καμπύλες και το ονομάζουμε SECTIONS. Εντός αυτού του geometric set δημιουργούμε άλλα δύο νέα και τα ονοματίζουμε FORE και AFT. Εντός του geometric set FORE δημιουργούμε τέσσερα νέα geometric sets, ένα για κάθε οικογένεια πλώρης, δηλαδή U,N,V,B. Αντίστοιχα εντός του geometric set AFT δημιουργούμε πέντε νέα, τα T,C,U,N,V. Στη συνέχεια εντός καθενός από τα παραπάνω geometric set δημιουργούμε νέα, ένα για κάθε σετ καμπύλων πλώρης ή πρύμνης, πχ. U1F70, U1F75, U1F80 κλπ. Συνεπώς η «βιβλιοθήκη» που δημιουργήθηκε έχει την μορφή που φαίνεται στην εικόνα 9 (χάρην εξοικονόμησης χώρου φαίνονται μόνο τα σετ νομέων πλώρης της οικογένειας “U”, αντίστοιχα δημιουργούμε geometric set εντός και των άλλων οικογενειών τόσο για την πλώρη όσο και για την πρύμνη:

Στην συνέχεια αφού δημιουργήθηκε η ιεραρχημένη σε geometric sets δομή της «βιβλιοθήκης», έγινε η εισαγωγή των καμπύλων-νομέων στα αντίστοιχα geometric set. Στην παρούσα φάση αντιμετωπίστηκε το παρακάτω πρόβλημα. Το CATIA δεν δίνει τη δυνατότητα στο σχεδιαστή να κάνει import αρχεία. Όλων των τύπων τα αρχεία ανοίγουν με την εντολή open το καθένα σε ξεχωριστό παράθυρο εντός του CATIA. Έτσι έπρεπε να ανοίγουμε τα αρχεία των νομέων που απαρτίζουν μία οικογένεια πλώρης ή πρύμνης το καθένα σε διαφορετικό παράθυρο και στη συνέχεια να αντιγράψουμε την κάθε καμπύλη στο αντίστοιχο geometric set εντός της «βιβλιοθήκης» που κατασκευάστηκε νωρίτερα. Λόγω του μεγάλου αριθμού αρχείων που έπρεπε να επεξεργαστούν, όλη η παραπάνω διαδικασία έγινε μέσω ενός προγράμματος γραμμένου σε γλώσσα Python το οποίο άνοιγε κάθε φορά όλες τις

καμπύλες που απαρτίζουν μία οικογένεια τις αντέγραφε στην αντίστοιχη θέση εντός της βιβλιοθήκης και τέλος τις μετονόμαζε ανάλογα με το ποιόν νομέα αναπαριστούν. Στην εικόνα 10 φαίνεται μέρος της «βιβλιοθήκης» στο CATIA (hull-SECTIONS-FORE-U-U1F70) καθώς και οι περιεχόμενοι νομείς (6.000 , 7.000 , 8.000 , 9.000 , 9.500 , 10.00)



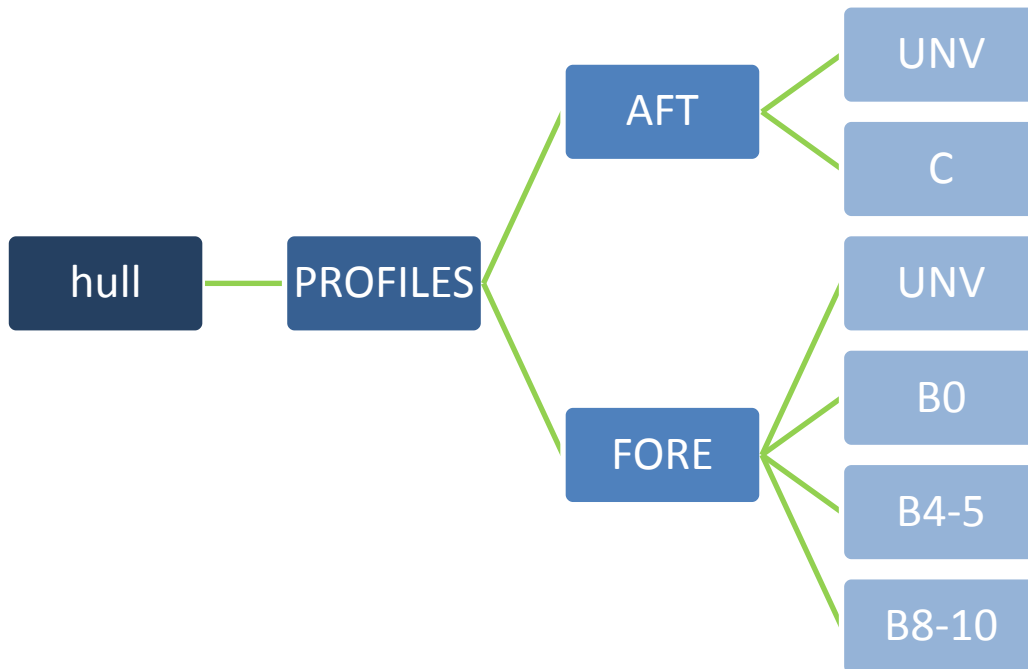
Εικόνα 9: Ιεραρχία τμήματος βιβλιοθήκης του CATIA που περιέχει τους νομείς της FORMDATA



Εικόνα 10: Παράδειγμα ιεραρχημένης οργάνωσης των νομέων εντός της βιβλιοθήκης του CATIA

6.2.Σχεδίαση των προφίλ πλώρης και πρύμνης

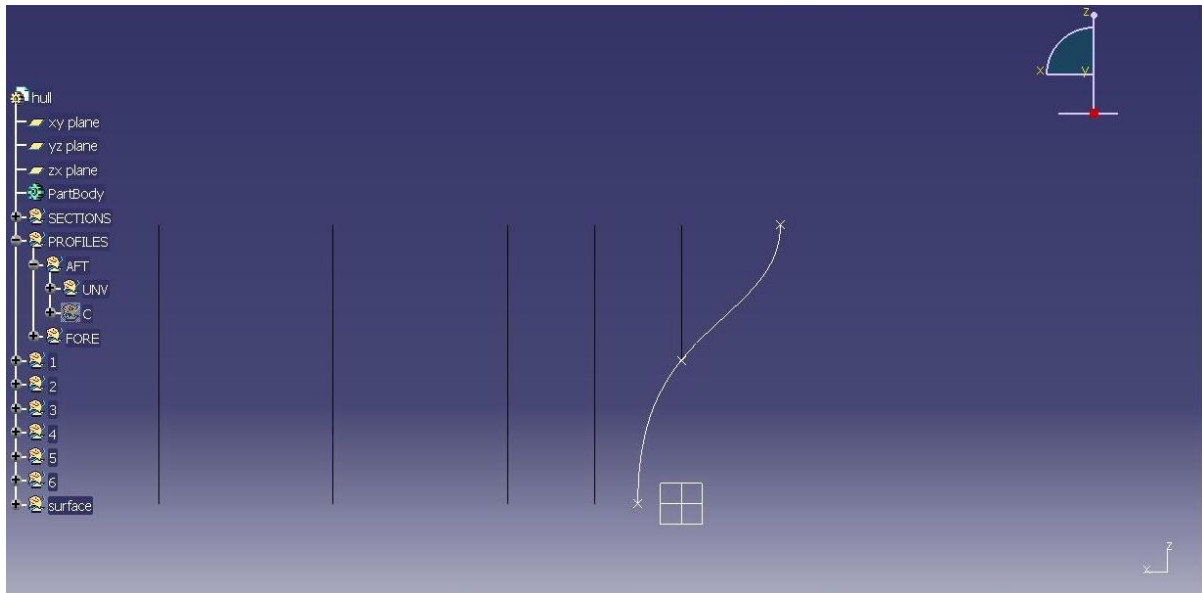
Με αντίστοιχη με την προηγούμενη διαδικασία δημιουργήθηκαν τα geometric sets που φαίνονται στην εικόνα 11 και κατόπιν σχεδιάστηκαν τα αντίστοιχα προφίλ πλώρης και πρύμνης με τρόπο που περιγράφεται παρακάτω.



Εικόνα 11: Ιεραρχία τμήματος βιβλιοθήκης του CATIA που περιέχει τα προφίλ της FORMDATA

Για την σχεδίαση των προφίλ χρησιμοποιήθηκαν σημεία επί των ακραίων νομέων καθώς και σημεία που δημιουργήθηκαν εκ νέου. Πιο συγκεκριμένα:

6.2.1. Πρύμνη Καταδρομικού



Εικόνα 12: Κατασκευή προφίλ πρύμνης καταδρομικού

Για την κατασκευή του προφίλ πρύμνης καταδρομικού (UNV) χρησιμοποιήθηκαν τα εξής σημεία:

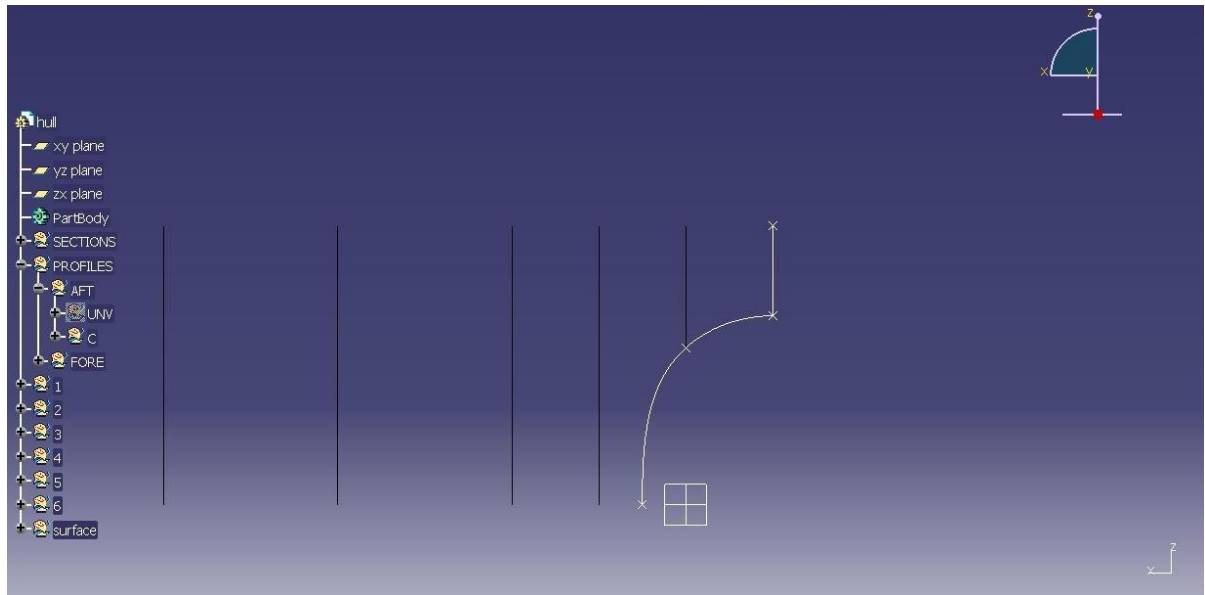
σημείο A: το σημείο τομής του νομέα -0.5 και του επιπέδου του Deck

σημείο B: το κατώτερο σημείο του νομέα 0

σημείο C: το σημείο τομής του νομέα 0.25 και του επιπέδου της Centerline

τα οποία παρεμβλήθηκαν από ένα Spline στο οποίο επιβλήθηκαν συγκεκριμένα εφαπτομενικά διανύσματα τόσο στο σημείο A όσο και στο σημείο C.

6.2.2. Πρύμνη Άβακας



Εικόνα 13: Κατασκευή προφίλ πρύμνης άβακα

Για την κατασκευή του προφίλ πρύμνης άβακα (C) χρησιμοποιήθηκαν τα εξής σημεία:

σημείο A: το σημείο τομής του νομέα -0.5 και του επιπέδου του Deck

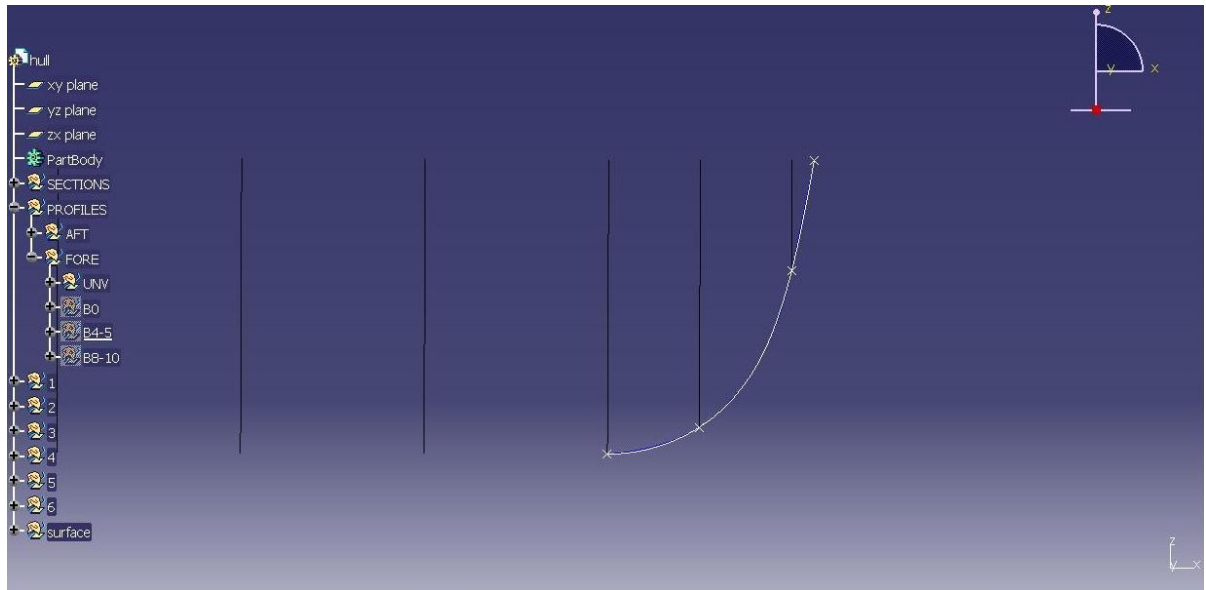
σημείο B: σημείο επί του επιπέδου του άβακα και ελάχιστα πάνω από την ίσαλο σχεδίασης

σημείο C: το κατώτερο σημείο του νομέα 0

σημείο D: το σημείο τομής του νομέα 0.25 και του επιπέδου της Centerline

Τα σημεία A και B παρεμβάλλονται με ευθεία και τα σημεία B,C και D με spline στο οποίο επιβλήθηκαν συγκεκριμένα εφαπτομενικά διανύσματα τόσο στο σημείο B όσο και στο σημείο D.

6.2.3. Πλώρη Χωρίς Βολβό



Εικόνα 14: Κατασκευή προφίλ πλώρης χωρίς βολβό

Για την κατασκευή της πλώρης χωρίς βολβό (UNV) χρησιμοποιήθηκαν τα εξής σημεία:

σημείο A: το κατώτερο σημείο του νομέα 9

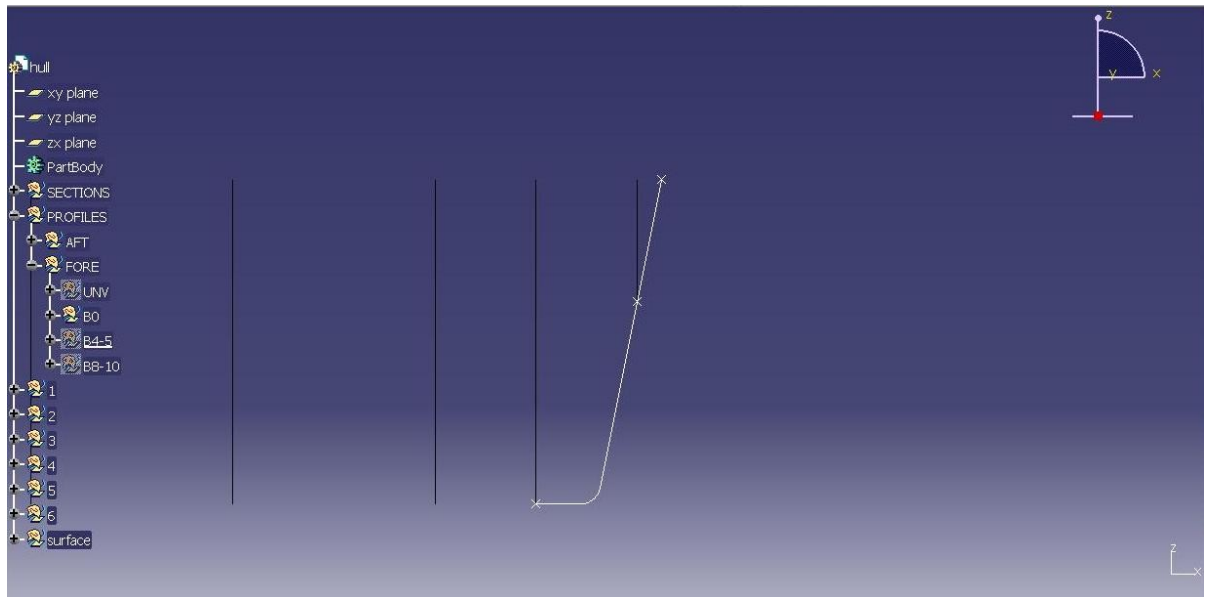
σημείο B: το κατώτερο σημείο του νομέα 9.5

σημείο C: το κατώτερο σημείο του νομέα 10

σημείο D: το ακραίο σημείο της Deckline

οποία παρεμβλήθηκαν από ένα Spline. Στο σημείο D επιβλήθηκε εφαπτομενικό διανύσματα παράλληλο στον άξονα X.

6.2.4.Βολβοειδής Πλώρη Β0



Εικόνα 15: Κατασκευή προφίλ βολβοειδούς πλώρης τύπου Β₀

Για την κατασκευή της βολβοειδούς πλώρης τύπου Β₀ χρησιμοποιήθηκαν τα εξής σημεία:

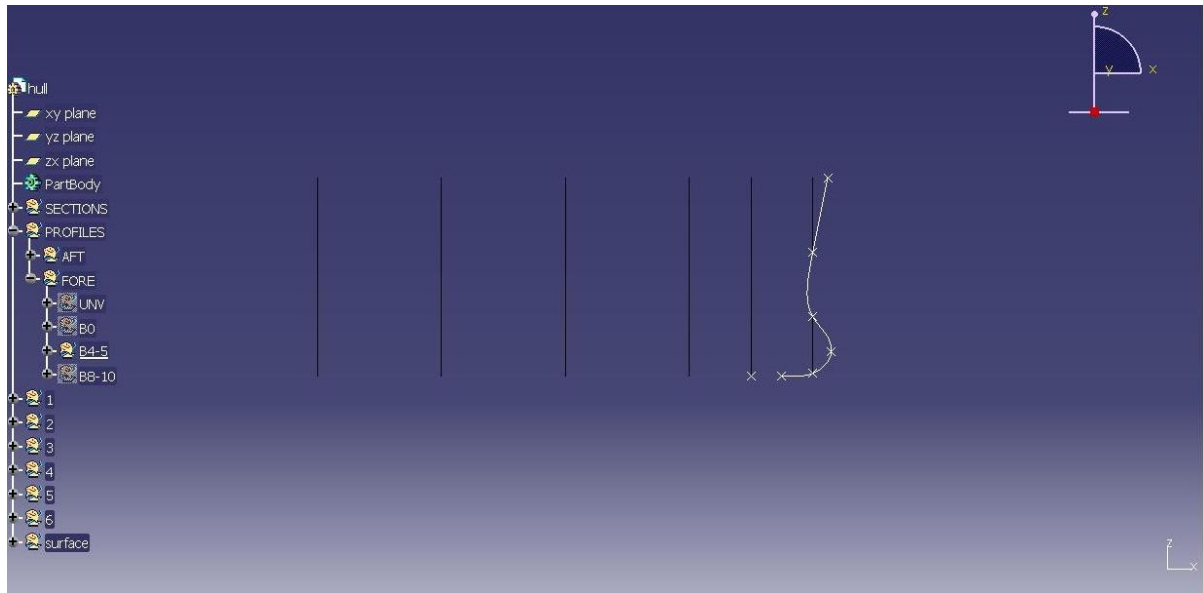
σημείο Α: το κατώτερο σημείο του νομέα 9.5

σημείο Β: το κατώτερο σημείο του νομέα 10

σημείο C: το ακραίο σημείο της Deckline

Από το σημείο Α και με φορά προς την πλώρη σχεδιάσαμε μία ευθεία παράλληλη με τον άξονα X, από το σημείο C στο σημείο Β άλλη μία ευθεία την οποία επεκτείναμε μέχρι να τμήσει την πρώτη ευθεία. Οι δύο ευθείες ενώθηκαν με τη χρήση ενός κυκλικού τομέα.

6.2.5.Βολβοειδής Πλώρη B4-5



Εικόνα 16: Κατασκευή προφίλ βολβοειδούς πλώρης τύπου B₄₋₅

Για την κατασκευή της βολβοειδούς πλώρης τύπου B4-5 χρησιμοποιήθηκαν τα εξής σημεία:

σημείο A: το σημείο τομής του νομέα $9 \frac{3}{4}$ με το επίπεδο της Center Line

σημείο B: το κατώτερο σημείο του νομέα 10 (τμήμα βολβού)

σημείο C: το ακραίο σημείο του Βολβού κατά τον άξονα X

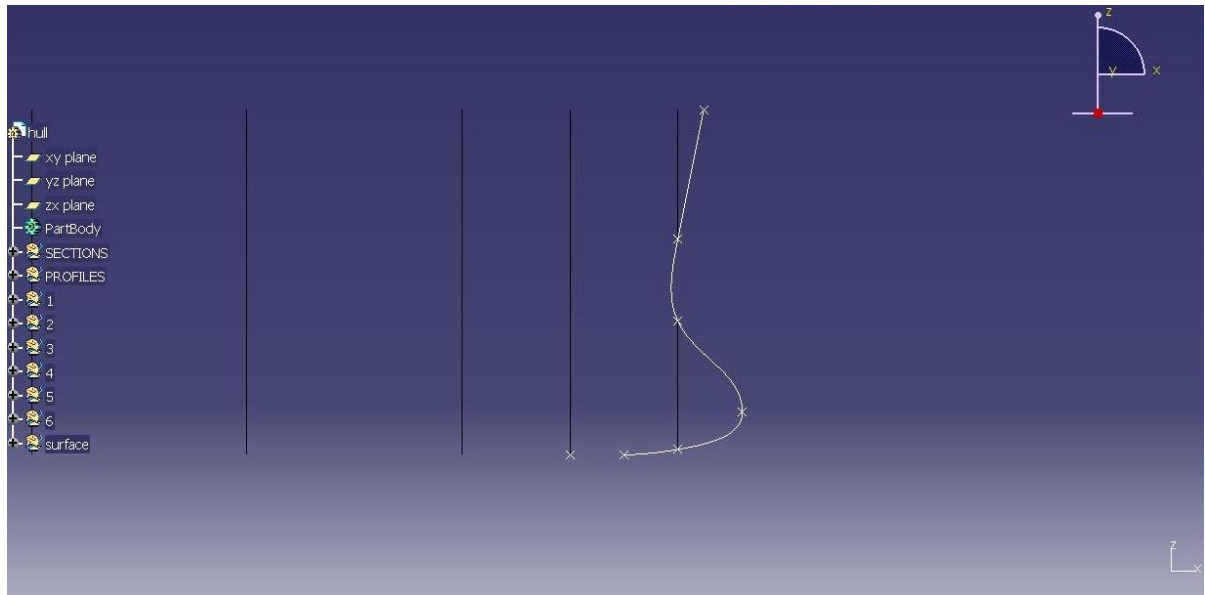
σημείο D: το ανώτερο σημείο του νομέα 10 (τμήμα βολβού)

σημείο E: το κατώτερο σημείο του νομέα 10 (ανώτερο τμήμα)

σημείο F: το ακραίο σημείο της Deckline

Τα σημεία A,B και C παρεμβλήθηκαν με ένα spline όπου στα άκρα A και C επιβλήθηκαν εφαπτομενικά διανύσματα παράλληλα στον άξονα X και Z αντίστοιχα. Τα σημεία E και F παρεμβλήθηκαν με μια ευθεία και τα σημεία C,D και E με ένα spline στο οποίο τα άκρα επιβλήθηκαν τα εφαπτομενικά διανύσματα του προηγούμενου spline στο άκρο C και της προηγούμενης ευθείας στο άκρο E.

6.2.6.Βολβοειδής Πλώρη B8-10



Εικόνα 17: Κατασκευή προφίλ βολβοειδούς πλώρης τύπου B₈₋₁₀

Για την κατασκευή της βολβοειδούς πλώρης τύπου B8-10 χρησιμοποιήθηκαν τα εξής σημεία:

σημείο A: το σημείο τομής του νομέα $9 \frac{3}{4}$ με το επίπεδο της Center Line

σημείο B: το κατώτερο σημείο του νομέα 10 (τμήμα βολβού)

σημείο C: το ακραίο σημείο του Βολβού κατά τον άξονα X

σημείο D: το ανώτερο σημείο του νομέα 10 (τμήμα βολβού)

σημείο E: το κατώτερο σημείο του νομέα 10 (ανώτερο τμήμα)

σημείο F: το ακραίο σημείο της Deckline

Τα σημεία A,B και C παρεμβλήθηκαν με ένα spline όπου στα άκρα A και C επιβλήθηκαν εφαπτομενικά διανύσματα παράλληλα στον άξονα X και Z αντίστοιχα. Τα σημεία E και F παρεμβλήθηκαν με μια ευθεία και τα σημεία C,D και E με ένα spline στο οποίο τα άκρα επιβλήθηκαν τα εφαπτομενικά διανύσματα παράλληλο με αυτό του προηγούμενου spline στο άκρο C και παράλληλο με την προηγούμενη ευθεία στο άκρο E.

6.3.Επιλογή οικογένειας πλώρης και πρύμνης

Στην γραμμή 22 του προγράμματος **final.py** καλείται το module **formdata** και συγκεκριμένα το function **selection** αυτού του module. Αυτό το module είναι υπεύθυνο για την επιλογή οικογένειας πλώρης και πρύμνης μέσω της συστηματικής σειράς FORMDATA και των δεδομένων που εισάγει ο χρήστης του προγράμματος. Συγκεκριμένα ο χρήστης καλείται να εισάγει τα εξής δεδομένα εισόδου:

- **L** : το μήκος μεταξύ καθέτων της υπό δημιουργία γάστρας σε μέτρα (m)
- **B** : το πλάτος της υπό δημιουργία γάστρας σε μέτρα (m)
- **D** : το κοίλο της υπό δημιουργία γάστρας σε μέτρα (m)
- **C_B** : ο συντελεστής γάστρας C_B της υπό δημιουργία γάστρας
- **LCB** : η θέση του διαμήκους κέντρου άντωσης σε μέτρα (m) με θετική φορά πρύμνηθεν
- **Τύπος πλώρης** : επιλογή μεταξύ βολβοειδούς και μη πλώρας

Αφού γίνει η εισαγωγή των δεδομένων, ακολουθεί ο υπολογισμός των C_{BA} και C_{BF} σύμφωνα με τη μέθοδο εφαρμογής της συστηματικής σειράς FORMDATA, και κατόπιν η στρογγυλοποίηση τους στο δεύτερο δεκαδικό ψηφίο και ανά πεντάδα. Για παράδειγμα αν C_{BA}=0.678 τότε μετά τη στρογγυλοποίηση θα έχουμε C_{BA}=0.70, ενώ αν είχαμε C_{BA}=0.671 μετά τη στρογγυλοποίηση θα γινόταν C_{BA}=0.65. Στη συνέχεια ανάλογα με την ύπαρξη ή όχι βολβοειδούς πλώρης καλείται ένα νέο module.

Στην περίπτωση ύπαρξης βολβοειδούς πλώρης καλείται το module Bulbrels και συγκεκριμένα το function του with_bulb. Πιο συγκεκριμένα διαβάζεται το αρχείο aft-families-bulb-relationships.txt και κατασκευάζεται ένα dictionary(rels) το οποίο αντιστοιχίζει σε κάθε διαθέσιμη πρύμνη την οικογένεια βολβοειδούς πλώρης με την οποία μπορεί να συνδυαστεί (πχ U3A50:B3F). Έπειτα διαβάζει όλους τους διαθέσιμους τύπους πρύμνης και τοποθετεί σε μία λίστα (aft_names_for_cba) όσους έχουν το C_{BA} που έχει υπολογιστεί νωρίτερα. Στη συνέχεια κατασκευάζεται ένα νέο dictionary (cb_given_cm{1:[70, 75, 80, 85, 90], 2:[55, 60, 65, 70, 75], 3:[50, 55, 60, 65, 70]}) το οποίο αντιστοιχίζει τιμές του συντελεστή μέσης τομής σε διαθέσιμες τιμές του συντελεστή γάστρας. Για παράδειγμα αν ο συντελεστής μέσης τομής μίας οικογένειας είναι ίσος με ένα τότε αυτή οι οικογένεια θα είναι διαθέσιμη με συντελεστές μέσης τομής 70,75,80,85 και 90. Ακολούθως κατασκευάζεται μία νέα λίστα (available_afts) ως εξής: χρησιμοποιώντας το dictionary rels με input τα στοιχεία της λίστας aft_names_for_cba βρίσκουμε κάθε φορά την οικογένεια πλώρης με την οποία μπορεί να συνδυαστεί η κάθε πρύμνη, καθώς και το Cm της. Μετά ελέγχουμε εάν υπάρχει πλώρη με τα παραπάνω C_{BF},Cm και κατασκευάζουμε τη λίστα available_afts από την οποία επιλέγουμε την πρύμνη που επιθυμούμε. Έπειτα κατασκευάζεται το dictionary bulb_size_given_cm = {1:[0, 5, 10],2:[0, 4, 8],3:[0, 4, 8]}, έτσι ώστε γνωρίζοντας το συντελεστή μέσης τομής να μπορέσουμε να επιλέξουμε μέγεθος βολβού. Τέλος γνωρίζοντας την οικογένεια πλώρης, το μέγεθος βολβού, και το C_BF δημιουργούμε και το όνομα της πλώρης που επιλέχθηκε.

Αντίστοιχα στην περίπτωση της μη βολβοειδούς πλήρης καλείται το module NoBulbrels και συγκεκριμένα το function του no bulb. Πιο συγκεκριμένα διαβάζεται το αρχείο aft-families-relationships.txt και κατασκευάζεται ένα dictionary(rels) το οποίο αντιστοιχίζει σε κάθε διαθέσιμη πρύμνη την οικογένεια μη βολβοειδούς πλήρης με την οποία μπορεί να συνδυαστεί (πχ U3A50:U3F). Έπειτα διαβάζει όλους τους διαθέσιμους τύπους πρύμνης και τοποθετεί σε μία λίστα (aft_names_for_cba) όσους έχουν το C_{BA} που έχει υπολογιστεί νωρίτερα. Στη συνέχεια κατασκευάζεται ένα νέο dictionary (cb_given_cm{1:[70, 75, 80, 85, 90],2:[55, 60, 65, 70, 75],3:[50, 55, 60, 65, 70],4:[45, 50, 55, 60, 65],5:[55, 60, 65],6:[70, 75, 80]}) το οποίο αντιστοιχίζει τιμές του συντελεστή μέσης τομής σε διαθέσιμες τιμές του συντελεστή γάστρας. Οι κατηγορίες 5 και 6 υπάρχουν γιατί εντοπίζονται μερικές εξαιρέσεις. Ουσιαστικά η 5 είναι υποκατηγορία της 2 και η 6 της 1. Ακολούθως κατασκευάζεται μία νέα λίστα (available_afts) ως εξής: χρησιμοποιώντας το dictionary rels με input τα στοιχεία της λίστας aft_names_for_cba βρίσκουμε κάθε φορά την οικογένεια πλήρης με την οποία μπορεί να συνδυαστεί η κάθε πρύμνη, καθώς και το C_M της. Μετά ελέγχουμε εάν υπάρχει πλήρη με τα παραπάνω C_{BF}, C_M και κατασκευάζουμε τη λίστα available_afts από την οποία επιλέγουμε την πρύμνη που επιθυμούμε. Με βάση την πρύμνη που επιλέχθηκε, χρησιμοποιώντας το dictionary rels βρίσκουμε την αντίστοιχη οικογένεια πλήρης. Τέλος γνωρίζοντας την οικογένεια πλήρης το C_M και C_{BF} δημιουργούμε και το όνομα της πλήρης

Και στις δύο παραπάνω περιπτώσεις καθορίζονται τα εξής δεδομένα: bowtype, BowFamilyName, bowprofile, sterntype, SternFamilyName, sternprofile, C_B , L, B, D, τα οποία εξάγονται στο κυρίως πρόγραμμα διότι χρησιμοποιούνται παρακάτω.

6.4.Κατασκευή βοηθητικών γεωμετρικών στοιχείων

Για την καλύτερη οργάνωση των γεωμετριών που θα κατασκευαστούν στη συνέχεια στο CATIA, δημιουργήθηκαν οι παρακάτω έξι κατηγορίες (σειρές 53-71 του προγράμματος final.py), ανάλογα με τον συνδυασμό πλήρης και πρύμνης.

ΚΑΤΗΓΟΡΙΑ 1: Πλήρη χωρίς βολβό και πρύμνη καταδρομικού.

ΚΑΤΗΓΟΡΙΑ 2: Πλήρη χωρίς βολβό και πρύμνη άβακας τύπου A,B ή C.

ΚΑΤΗΓΟΡΙΑ 3: Πλήρη χωρίς βολβό και πρύμνη άβακας τύπου D.

ΚΑΤΗΓΟΡΙΑ 4: Βολβοειδής πλήρη και πρύμνη καταδρομικού.

ΚΑΤΗΓΟΡΙΑ 5: Βολβοειδής πλήρη και πρύμνη άβακας τύπου A,B ή C.

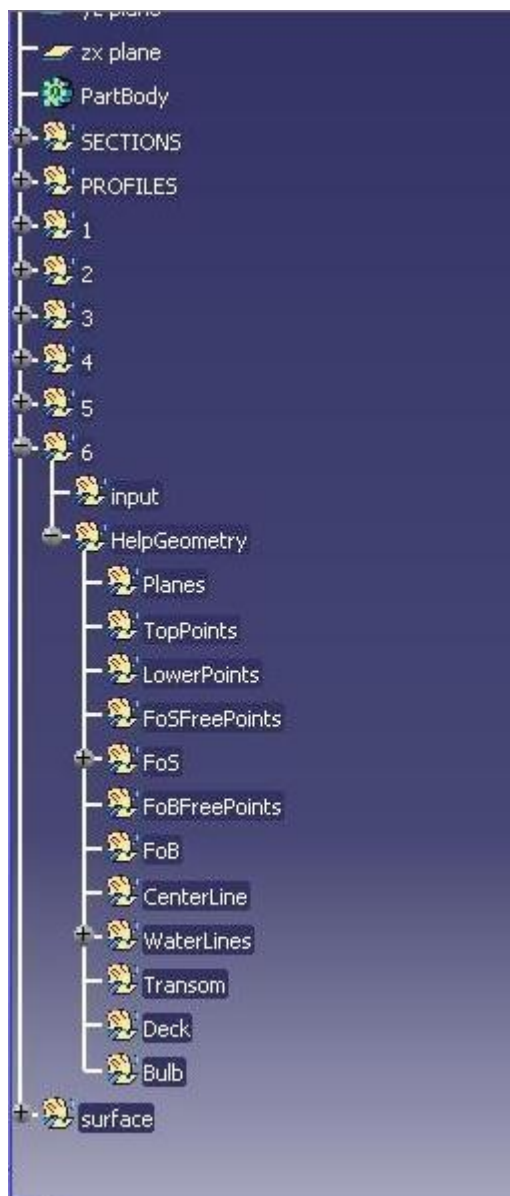
ΚΑΤΗΓΟΡΙΑ 6: Βολβοειδής πλήρη και πρύμνη άβακας τύπου D.

Σε αντιστοιχία με τον τρόπο που περιγράφηκε νωρίτερα, για την ιεραρχημένη οργάνωση της βιβλιοθήκης που περιέχει τους νομείς της συστηματικής σειράς FORMDATA, δημιουργήθηκαν έξι geometric sets, ένα για κάθε μια από της παραπάνω κατηγορίες. Εντός καθενός από αυτά δημιουργήθηκαν δύο νέα geometric set, ένα (input)

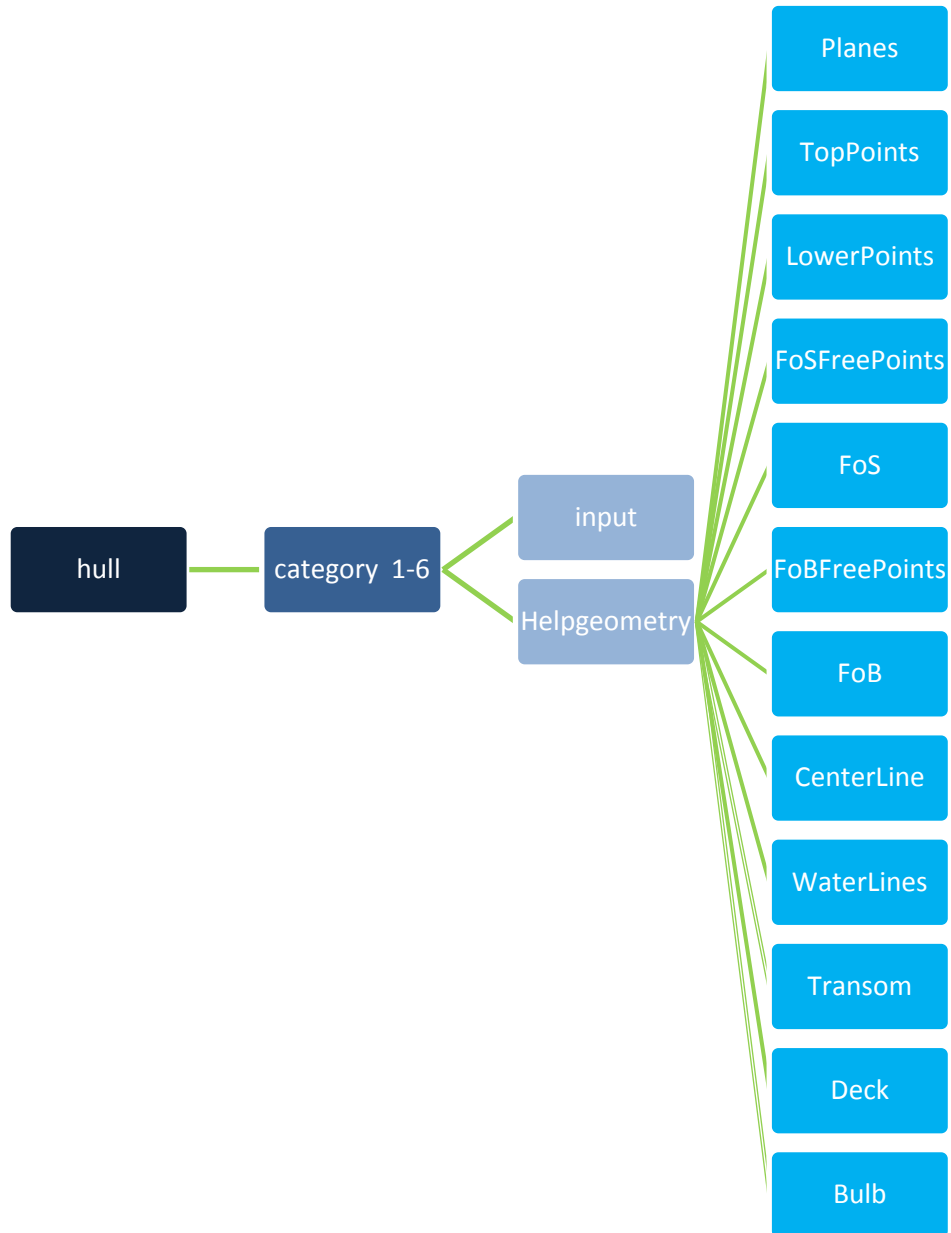
στο οποίο θα αντιγράφονται οι εκάστοτε νομείς που χρησιμοποιούνται για την κατασκευή του τρισδιάστατου μοντέλου, και άλλο ένα (**HelpGeometry**) στο οποίο θα τοποθετούνται οι όποιες βοηθητικές γεωμετρικές κατασκευάζονται κατά τη διάρκεια της εκτέλεσης του προγράμματος **final.py**.

Εντός του geometric set **HelpGeometry** της καθεμίας κατηγορίας (1-6) δημιουργήθηκαν τα εξής geometric sets για την οργάνωση των δημιουργούμενων γεωμετριών (Εικόνες 18-19):

<u>Planes:</u>	Εδώ τοποθετούνται όποιου είδους επίπεδα δημιουργούνται κατά τη διαδικασία της σχεδίασης.
<u>TopPoints:</u>	Εδώ τοποθετούνται τα κορυφαία σημεία των νομέων και των προφίλ.
<u>LowerPoints:</u>	Εδώ τοποθετούνται τα κατώτερα σημεία των νομέων και των προφίλ.
<u>FoSFreePoints:</u>	Εδώ τοποθετούνται σημεία, επί των νομέων που διαθέτουν επίπεδο κομμάτι, από τα οποία θα σχεδιαστεί η καμπύλη του flat of side.
<u>FoS:</u>	Εδώ τοποθετούνται τα κομμάτια που απαρτίζουν την καμπύλη του flat of side.
<u>FoBFreePoints:</u>	Εδώ τοποθετούνται σημεία, επί των νομέων που διαθέτουν επίπεδο κομμάτι, από τα οποία θα σχεδιαστεί η καμπύλη του flat of bottom.
<u>FoB:</u>	Εδώ τοποθετούνται τα κομμάτια που απαρτίζουν την καμπύλη του flat of bottom.
<u>CenterLine:</u>	Εδώ τοποθετούνται τα κομμάτια που απαρτίζουν την center line.
<u>WaterLines:</u>	Εδώ τοποθετούνται οι τομές των νομέων με το επίπεδο της εκάστοτε water line (ξεχωριστό geometric set: Intersections), καθώς και οι water lines που σχεδιάζονται.
<u>Transom:</u>	Εδώ τοποθετούνται όλες οι γεωμετρικές που δημιουργούνται για την κατασκευή της πρύμνης άβακας. Το συγκεκριμένο geometric set εμπεριέχεται μόνο στις κατηγορίες 2,3,5 και 6.
<u>Deck:</u>	Εδώ τοποθετούνται τα κομμάτια που απαρτίζουν την deck line.
<u>Bulb:</u>	Εδώ τοποθετούνται όλες οι γεωμετρικές που δημιουργούνται για την κατασκευή της βολβοειδούς πλώρης. Το συγκεκριμένο geometric set εμπεριέχεται μόνο στις κατηγορίες 4,5 και 6.



Εικόνα 18: παράδειγμα οργάνωσης με geometric set

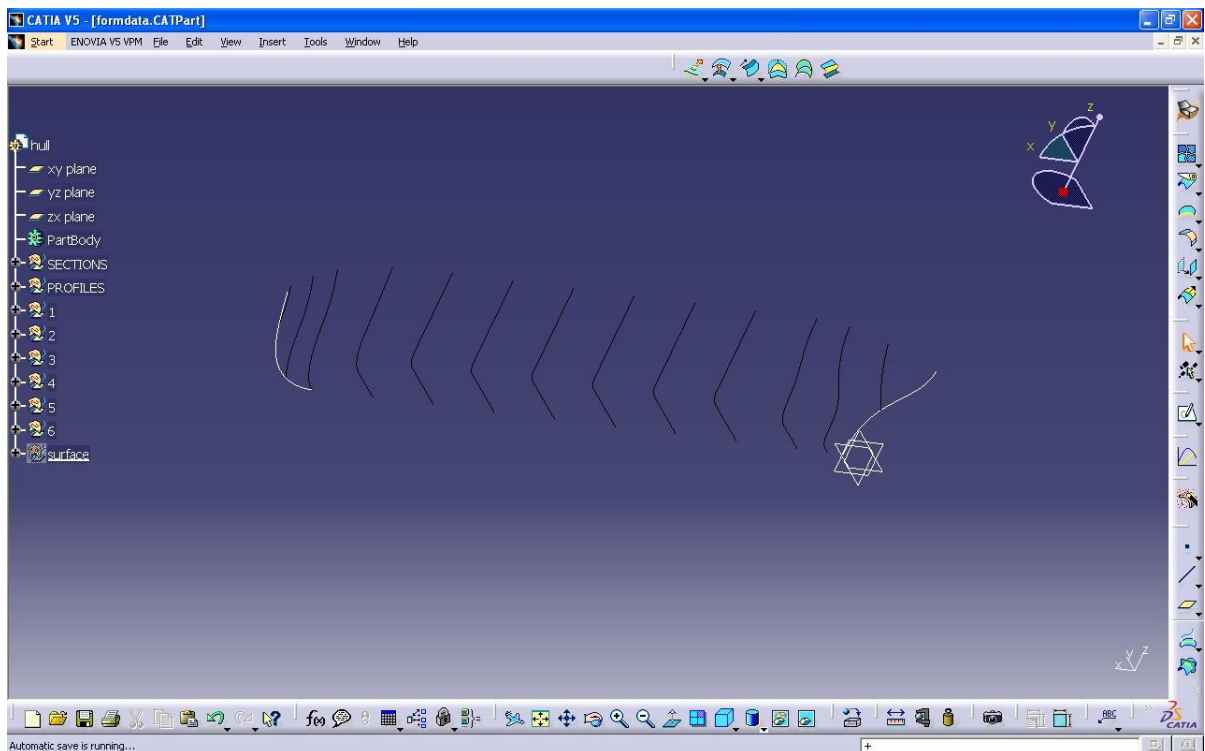


Εικόνα 19: παράδειγμα οργάνωσης με `geometric set`

Στη συνέχεια (κεφ.6.3.1-6.3.10) ακολουθεί περιγραφή του τρόπου που δημιουργείται καθεμία από τις παραπάνω βοηθητικές γεωμετρίες, καθώς και περιγραφή του αντίστοιχου μέρους του προγράμματος `final.py`, με τη σειρά που αυτό γίνεται κατά την εκτέλεση του.

6.4.1. Αντιγραφή των νομέων και των προφίλ

Αφού γίνει η επιλογή οικογένειας πλώρης και πρύμνης (*block: INPUT SELECTION, γραμμές: 21-34 του προγράμματος final.py*) και οριστεί η κατηγορία 1 έως 6 ανάλογα με τον τύπο πλώρης και πρύμνης, ακολουθεί η αντιγραφή των αντίστοιχων νομέων και προφίλ (Εικόνα 20) από την «βιβλιοθήκη» στο geometric set **input** εντός του geometric set της αντίστοιχης κατηγορίας. Πχ Έστω ότι έχει επιλεγεί πλοίο με πλώρη χωρίς βολβό και πρύμνη καταδρομικού: η κατηγορία θα είναι η “1” και συνεπώς οι νομείς και τα προφίλ θα αντιγραφούν στη θέση **hull**⇒**1**⇒**input**.



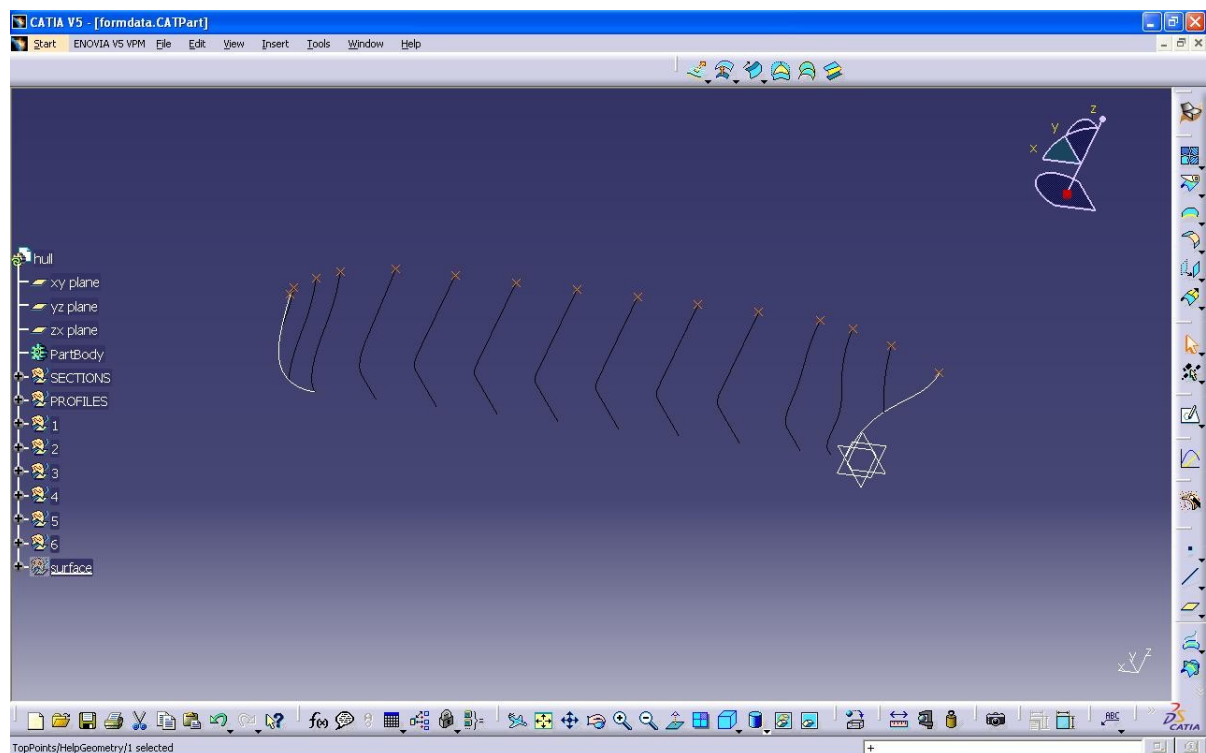
Εικόνα 20: Αντιγραφή των νομέων και των προφίλ

6.4.2.Δημιουργία Top & Lower points επί των νομέων

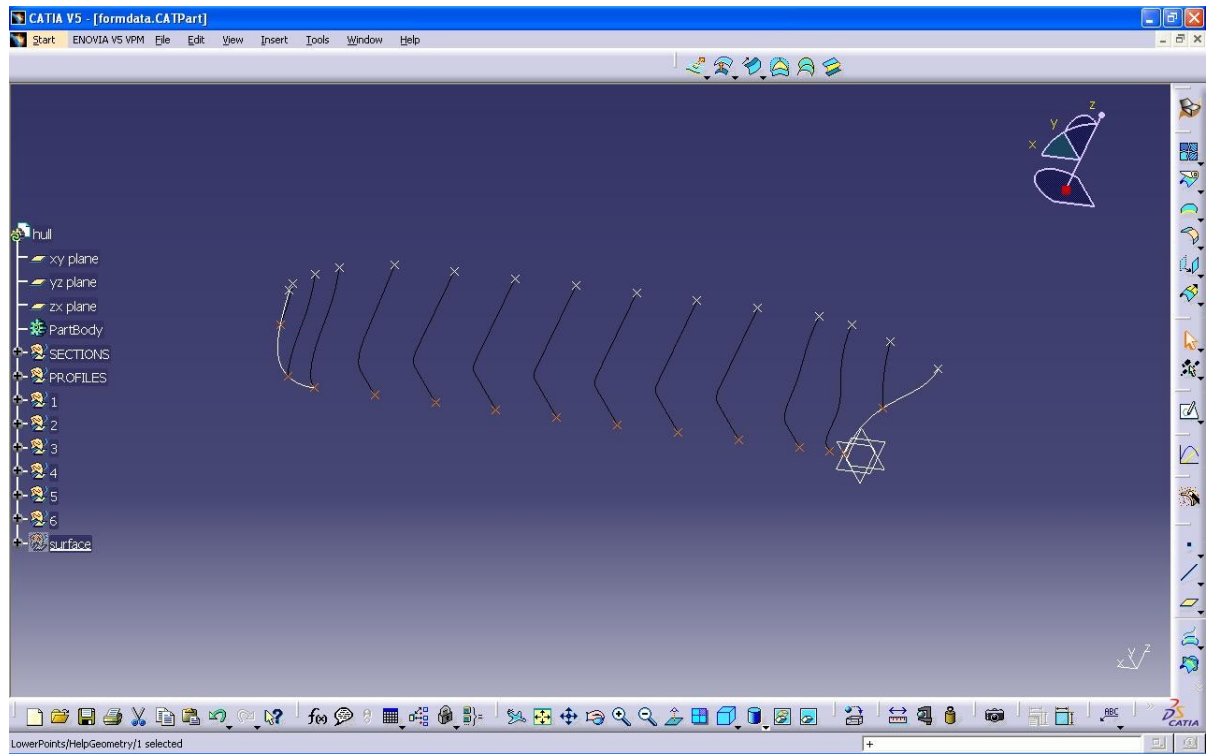
Μετά την αντιγραφή των νομέων και των προφίλ, που έχουν επιλεγεί, στην κατάλληλη θέση, ακολουθεί η δημιουργία (Εικόνες 21-22) των «ανώτερων» (*block: TopPoints CREATION, γραμμές:109-119 του προγράμματος final.py*) και «κατώτερων» (*block: LowerPoints CREATION, γραμμές:121-149 του προγράμματος final.py*) ακραίων σημείων του κάθε νομέα και του κάθε προφίλ στις θέσεις:

hull⇒**category**⇒**HelpGeometry**⇒**TopPoints** και
hull⇒**category**⇒**HelpGeometry**⇒**LowerPoints** αντίστοιχα.

Για την δημιουργία αυτών χρησιμοποιήθηκε το **function: AddNewExtremum** του **object: Extremum** (Παράρτημα 2).



Εικόνα 21: Δημιουργία ανώτερων ακραίων σημείων νομέων και προφίλ

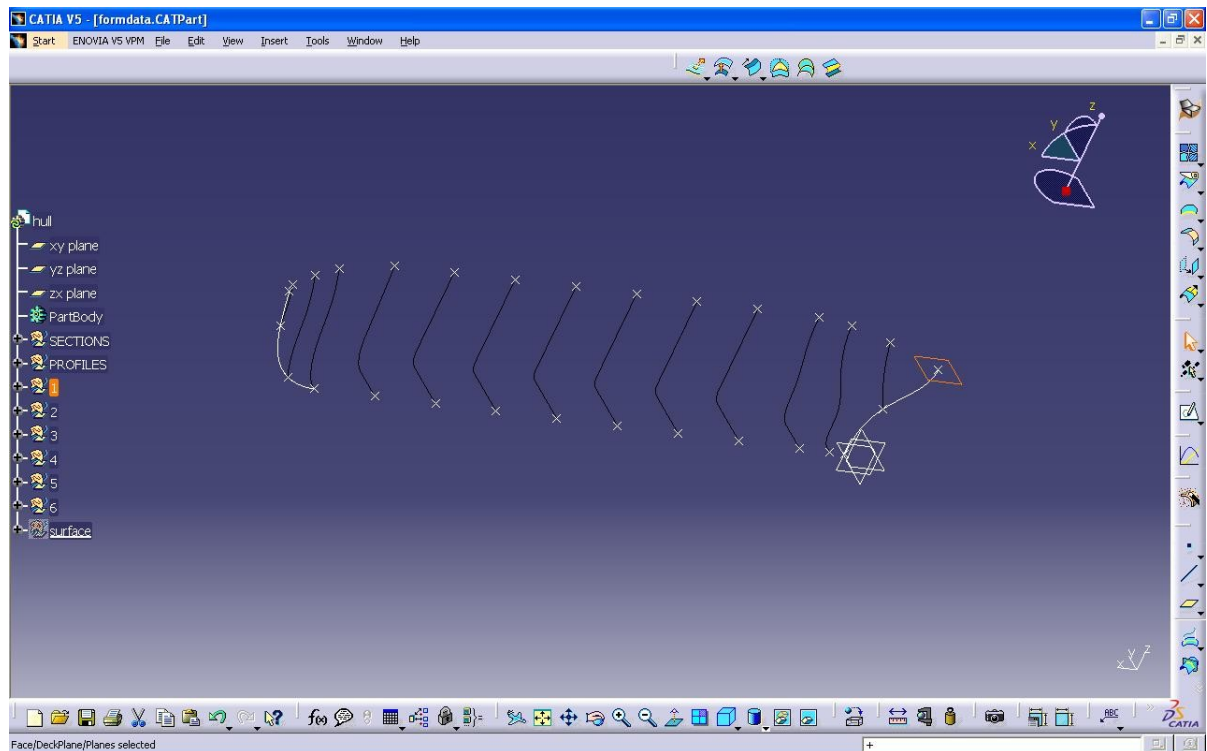


Εικόνα 22: Δημιουργία κατώτερων ακραίων σημείων νομέων και προφίλ

6.4.3. Δημιουργία του Deck Plane

Στη συνέχεια κατασκευάστηκε το επίπεδο του κυρίου καταστρώματος (*block: DeckPlane CREATION, σειρές:151-161 του προγράμματος final.py*) ως ένα παράλληλο επίπεδο του βασικού επιπέδου XY το οποίο διέρχεται από ένα συγκεκριμένο σημείο, στην προκειμένη το ανώτατο ακραίο σημείο του πρυμναίου προφίλ (Εικόνα 23).

Για την δημιουργία αυτών χρησιμοποιήθηκε το **function: AddNewPlaneOffsetPt** του **object: Plane** (Παράρτημα 2).



Εικόνα 23: Κατασκευή επιπέδου κυρίου καταστρώματος

6.4.4. Δημιουργία των σημείων των flat of side και flat of bottom

Έπειτα ακολουθεί η σχεδίαση των σημείων του flat of side και του flat of bottom. Εξετάζεται, για κάθε νομέα, αν είτε στην αρχή του, είτε στο τέλος του υπάρχει ευθιογενές κομμάτι και στην άκρη κάθε ευθείας (εφόσον αυτή υφίσταται) σχεδιάζεται ένα σημείο. Όταν αυτό το σημείο βρίσκεται επί της πλάγιας επιφάνειας της γάστρας το αποκαλούμε **σημείο flat of side** και όταν αυτό βρίσκεται επί του πυθμένα της γάστρας το αποκαλούμε **σημείο flat of bottom**. Αυτό γίνεται μέσω ενός προγράμματος στο **Grasshopper** το οποίο χρησιμοποιεί την εξής ιδιότητα των καμπύλων **B-Splines**: Εάν (k) κορυφές του πολυγώνου De Boor είναι συγγραμμικές, τότε η καμπύλη $Q(t)$ και το πολύγωνο De Boor ταυτίζονται κατά μήκος ενός παραμετρικού διαστήματος. Οι καμπύλες που χρησιμοποιήθηκαν είναι τάξεως $k=4$ συνεπώς αρκεί να ελεγχθεί η ύπαρξη τεσσάρων συνευθειακών σημείων ελέγχου και στα δύο άκρα της κάθε καμπύλης.

6.4.4.1. Κατασκευή αρχείων με τις συντεταγμένες των σημείων των flat of side και flat of bottom στο Grasshopper

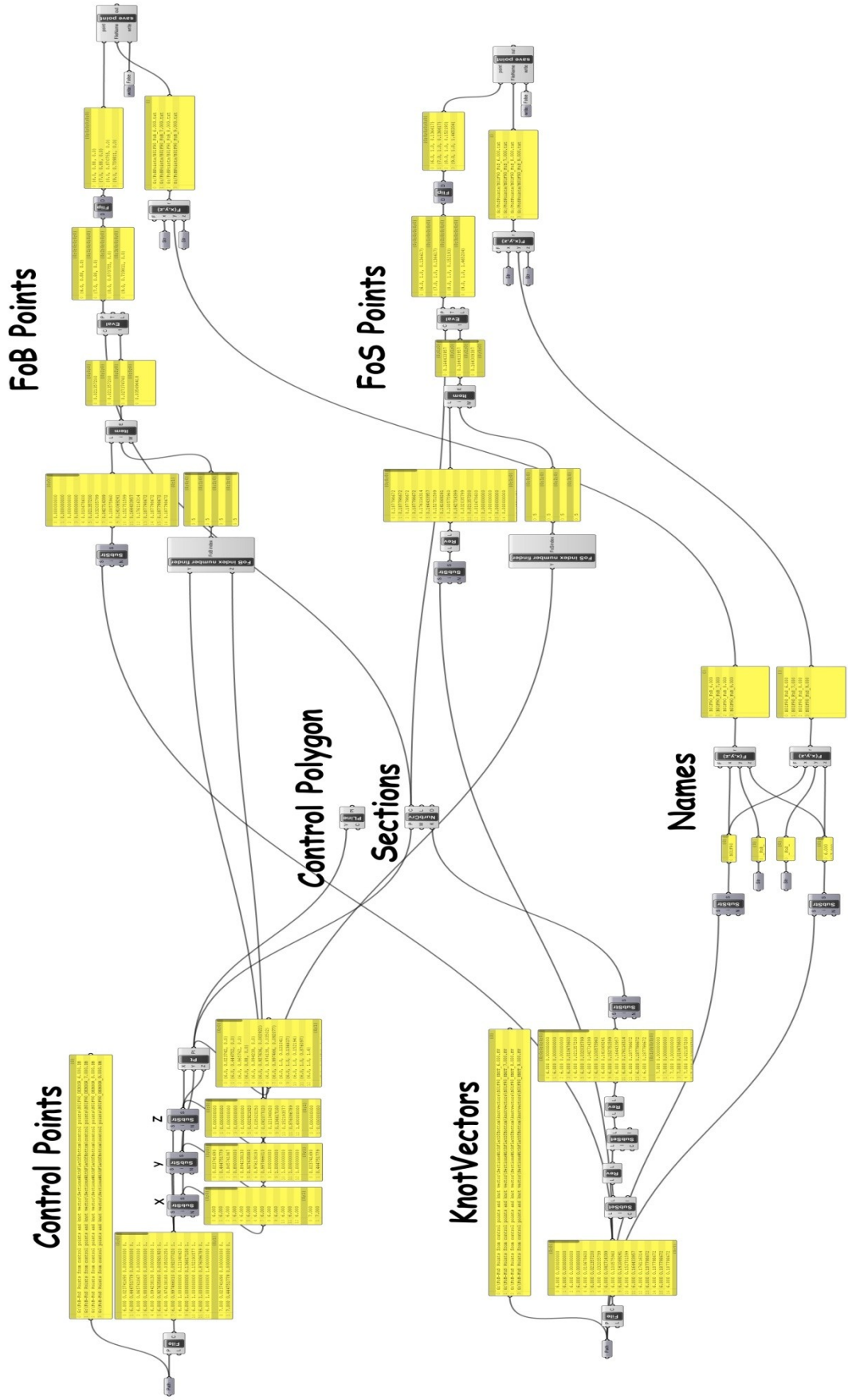
Χρησιμοποιώντας μέρος του προηγούμενου καμβά που δημιουργήθηκε στο Grasshopper (και συγκεκριμένα τα block **control points**, **knotvectors**, **file names** και **curves creation**) δημιουργούμε και πάλι τις καμπύλες. Έπειτα κατασκευάστηκαν δύο νέα block εντολών (FOS POINTS και FOB POINTS) τα οποία αρχικά ελέγχουν την ύπαρξη συνευθειακών control points (και πιο συγκεκριμένα τουλάχιστον τεσσάρων), το ένα στην αρχή της κάθε καμπύλης (σημείο flat of bottom) και το άλλο στο τέλος (σημείο flat of side).

Για την περίπτωση που ελέγχεται η ύπαρξη σημείου flat of side αρκεί να γίνει έλεγχος των Y συντεταγμένων των σημείων ελέγχου. Το επίπεδο του flat of side είναι πάντα παράλληλο στο επίπεδο XZ , συνεπώς αρκεί να ελεγχθεί ποιών σημείων η Y συντεταγμένη είναι ίση με 1. Το επίπεδο του flat of bottom δεν είναι πάντα παράλληλο του XY (περίπτωση κεκλιμένου πυθμένα, ύπαρξη deadrise) συνεπώς είναι απαραίτητος ο έλεγχος για την συνευθειακότητα των σημείων.

Αφού εξακριβωθεί η ύπαρξη τουλάχιστον τεσσάρων συνευθειακών σημείων ελέγχου χρησιμοποιείται η εντολή Eval (Evaluate curve) του Grasshopper για να υπολογιστούν οι συντεταγμένες του σημείου όπου η καμπύλη παύει να ταυτίζεται με το πολύγωνο ελέγχου της. Η παραπάνω εντολή δέχεται ως δεδομένα εισόδου την καμπύλη και μία συγκεκριμένη παράμετρο, εντός του παραμετρικού διαστήματος της καμπύλης, και κατόπιν υπολογίζει το αντίστοιχο σημείο επί της καμπύλης.

Στην περίπτωση που εξετάζεται εδώ, επειδή όλες οι καμπύλες είναι τάξεως $k=4$, αναζητούμε πάντα τέσσερα συνευθειακά σημεία ελέγχου. Συνεπώς, εφόσον ικανοποιείται η προηγούμενη συνθήκη, υπολογίζεται το σημείο επί της καμπύλης στην παράμετρο $n=k+1=4+1=5$. Κατόπιν το σημείο το οποίο υπολογίζεται αποθηκεύεται σε ένα αρχείο κειμένου. Δημιουργούνται δύο διαφορετικά αρχεία ένα για το σημείο του **flat of side** και ένα για το σημείο του **flat of bottom**, της μορφής :

(Όνομα οικογένειας)_(FoS ή FoB)_(όνομα νομέα).txt



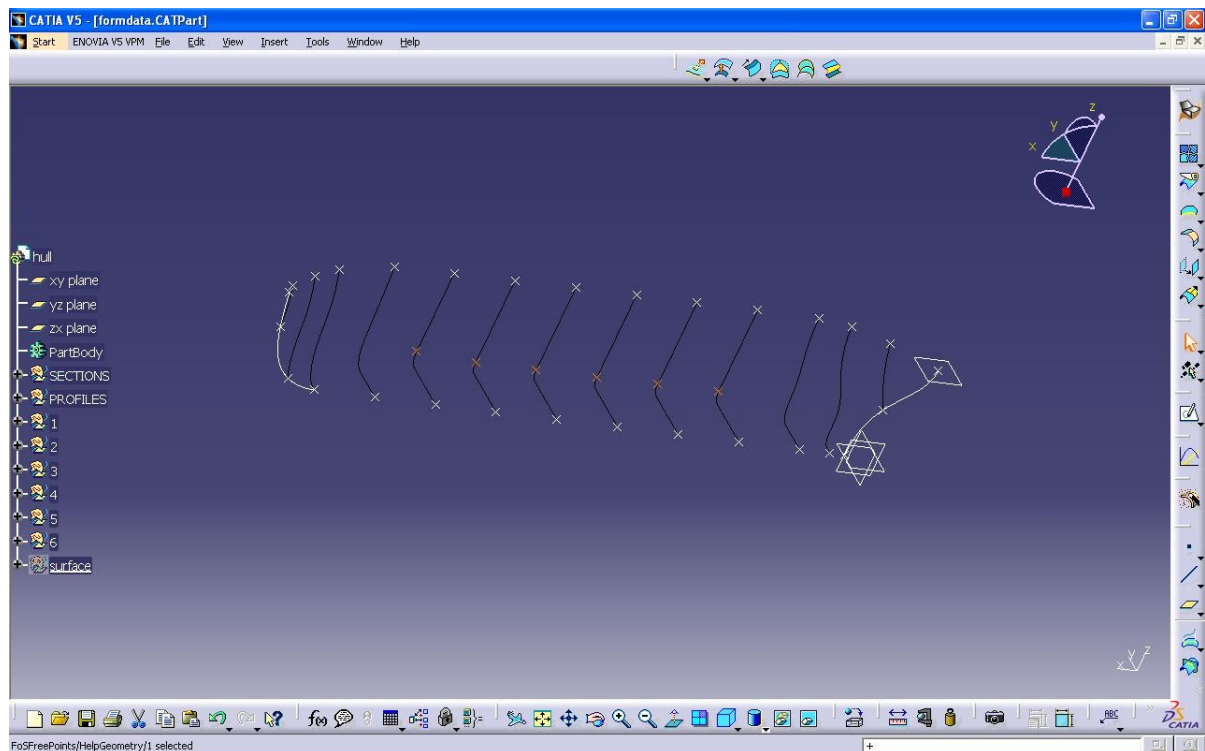
6.4.4.2. Κατασκευή των σημείων των flat of side και flat of bottom στο CATIA

Αφού έχουν κατασκευαστεί τα αρχεία που περιέχουν τις συντεταγμένες των προαναφερθέντων σημείων, ακολουθεί η σχεδίαση τους επί του τρισδιάστατου μοντέλου, όποτε αυτά υφίστανται (ανάλογα με το ποιές οικογένειες πλήρης και πρύμνης έχουν επιλεγεί).

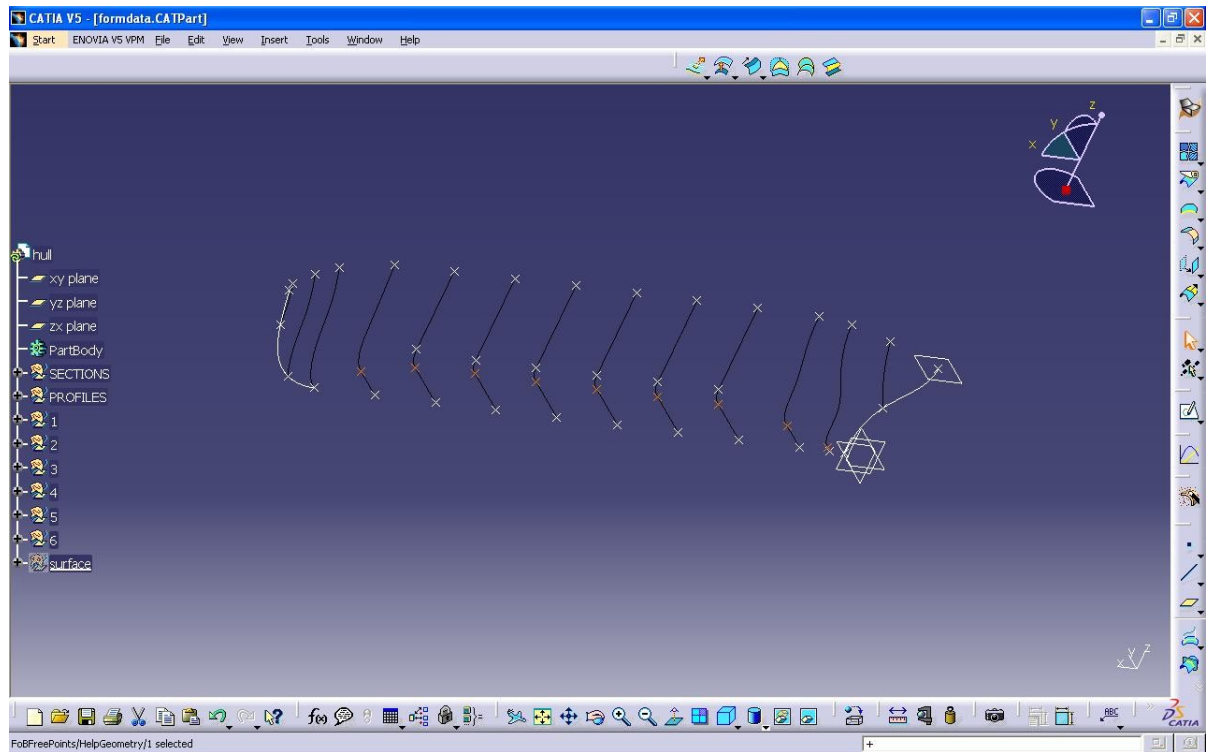
Τα αρχεία που έχουν δημιουργηθεί έχουν ονομαστεί σύμφωνα με την παρακάτω φόρμουλα: **(Όνομα οικογένειας πλήρης ή πρύμνης)_(FoS ή FoB)_(όνομα νομέα).txt** Συνεπώς γνωρίζοντας την οικογένεια πλήρης και πρύμνης μπορούμε να αναζητήσουμε, στα αντίστοιχα directory, εάν υπάρχουν σημεία flat of side και σημεία flat of bottom για τους νομείς που χρησιμοποιούνται κάθε φορά για την κατασκευή του τρισδιάστατου μοντέλου. Εφόσον υπάρχουν τα αντίστοιχα αρχεία (*block: FoB and FoSPoints CREATION, στις σειρές:163-165 του προγράμματος final.py καλείται το module FoBFoSPoints το οποίο περιέχει τα functions: FoSPointsCreation και FoBPointsCreation*) χρησιμοποιούνται οι συντεταγμένες που περιέχουν και σχεδιάζονται (Εικόνες 24-25) τα σημεία, τα οποία τοποθετούνται στις θέσεις:

hull⇒category⇒HelpGeometry⇒FoSFreePoints και

hull⇒category⇒HelpGeometry⇒FoBFreePoints αντίστοιχα.



Εικόνα 24 : Δημιουργία σημείων Flat of Side



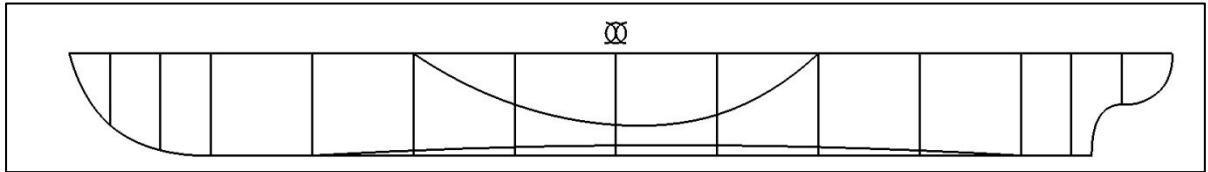
Εικόνα 25 : Δημιουργία σημείων Flat of Bottom

Για την δημιουργία αυτών χρησιμοποιήθηκε το **function: AddNewPointCoord** του **object: Point** (Παράρτημα 2).

Στη συνέχεια γίνεται έλεγχος της θέσης των ακραίων σημείων του Flat of Side (*block: REPLACE FSS POINTS TOO CLOSE TO DECK PLANE, στις σειρές:166-167 του προγράμματος final.py καλείται το module: replacefsspoints το οποίο περιέχει το function: replace*) και συγκεκριμένα της Z συντεταγμένης τους. Εάν το σημείο βρίσκεται σε απόσταση μικρότερη του 0.1mm ($Z_{coord} > 1.5\text{mm}$) από το επίπεδο του Deck, τότε αυτό αντικαθίσταται με νέο σημείο στην κορυφή του αντίστοιχου νομέα, δηλαδή επί του επιπέδου του Deck. Ο λόγος αυτού του ελέγχου και της αντικατάστασης η μη του σημείου με νέο επί του επιπέδου του Deck, εξηγείται σε επόμενη παράγραφο όπου περιγράφεται η κατασκευή της επιφάνειας της γάστρας.

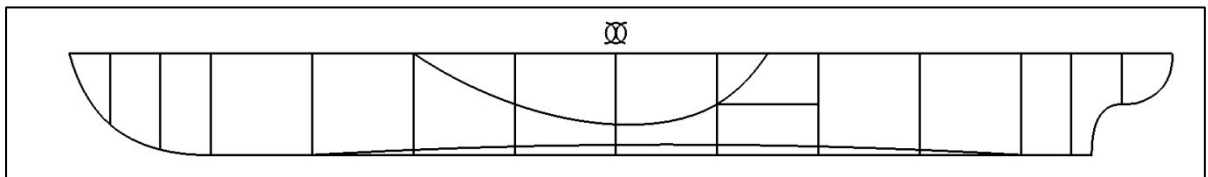
Έπειτα ακολουθεί ο προσδιορισμός της κατηγορίας fos (*block:foscategoroy στις σειρές:168-170 του προγράμματος final.py καλείται το module: foscategoroycheck το οποίο περιέχει το function:foscategoroy*). Ο προσδιορισμός αυτών των κατηγοριών βοηθάει μετέπειτα, στην ευκολότερη κατασκευή της επιφάνειας της γάστρας. Διακρίνονται τέσσερις κατηγορίες (Εικόνες 26 a-b-c-d) ανάλογα με το που βρίσκονται τα ακραία σημεία του Flat of Side (επί του επιπέδου του Deck ή όχι):

Κατηγορία both : και το προραίο και το πρυμναίο σημείο του Flat of Side βρίσκονται επί του επιπέδου του Deck.



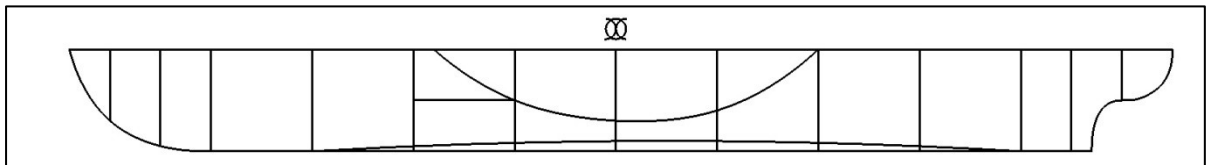
Εικόνα 26α : foscateory=both

Κατηγορία fore : μόνο το προραίο σημείο του Flat of Side βρίσκεται επί του επιπέδου του Deck.



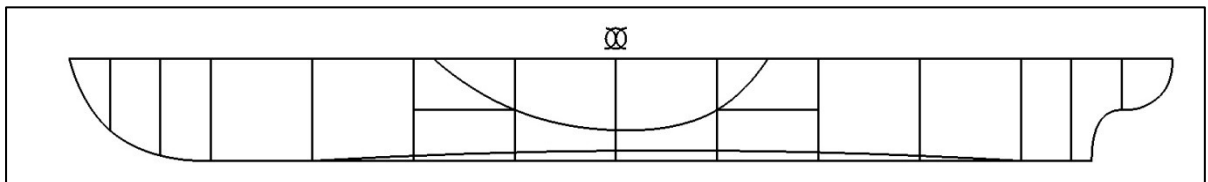
Εικόνα 26b : foscateory=fore

Κατηγορία aft : μόνο το πρυμναίο σημείο του Flat of Side βρίσκεται επί του επιπέδου του Deck.



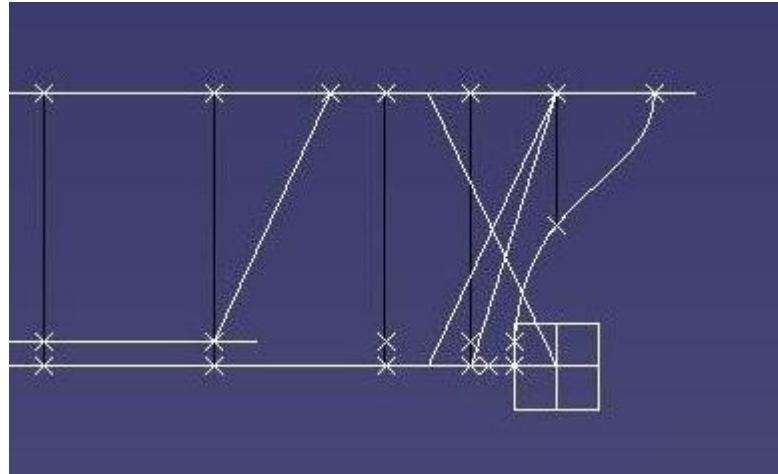
Εικόνα 26c : foscateory=aft

Κατηγορία none : ούτε το προραίο ούτε το πρυμναίο σημείο του Flat of Side βρίσκονται επί του επιπέδου του Deck.



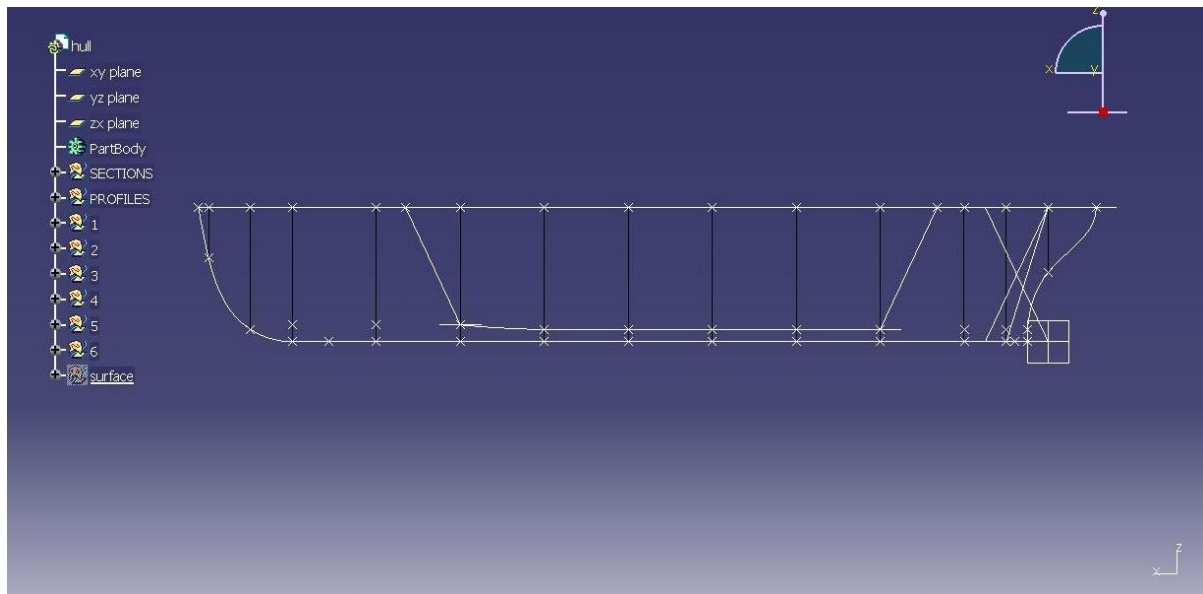
Εικόνα 26d : foscateory=none

Για την περίπτωση όπου κάποιο από τα ακραία σημεία του Flat of Side δεν βρίσκεται επί του deck plane είναι αναγκαίο να προσδιοριστούν δύο σημεία (ένα στην αρχή του FoS και ένα στο τέλος) επί του επιπέδου του Deck (*block:FoSForeTopPoint AND FoSAftTopPoint CREATION*, στις σειρές:171-172 του προγράμματος *final.py* καλείται το *module: FoSForeAft* το οποίο περιέχει το *function:FoS*). Χρησιμοποιώντας δύο ευθείες (Εικόνα 27a) με συγκεκριμένη κλίση, μία για την κλίση της πλώρης και μία για την κλίση της πρύμνης, κατασκευάζουμε δύο ευθείες παράλληλες σε αυτές, με αρχή τα ακραία σημεία FoS μέχρι το επίπεδο του Deck (Εικόνα 27b) καθώς και τα κορυφαία ακραία σημεία αυτών (Εικόνα 27c).

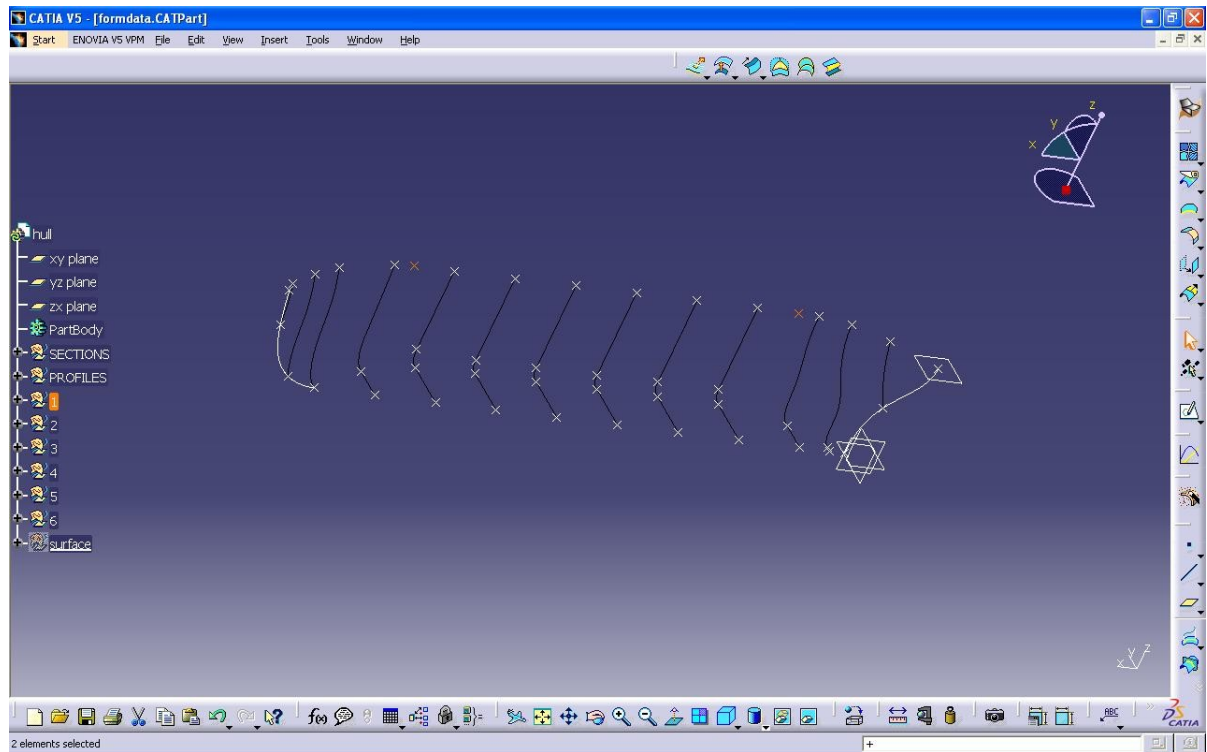


Εικόνα 27a: Οι ευθείες που ορίζουν την κλίση

Τα σημεία αυτά τοποθετήθηκαν στη θέση: **hull**⇒**category**⇒**HelpGeometry**⇒**FoS**



Εικόνα 27b: Οι κατασκευαζόμενες ευθείες

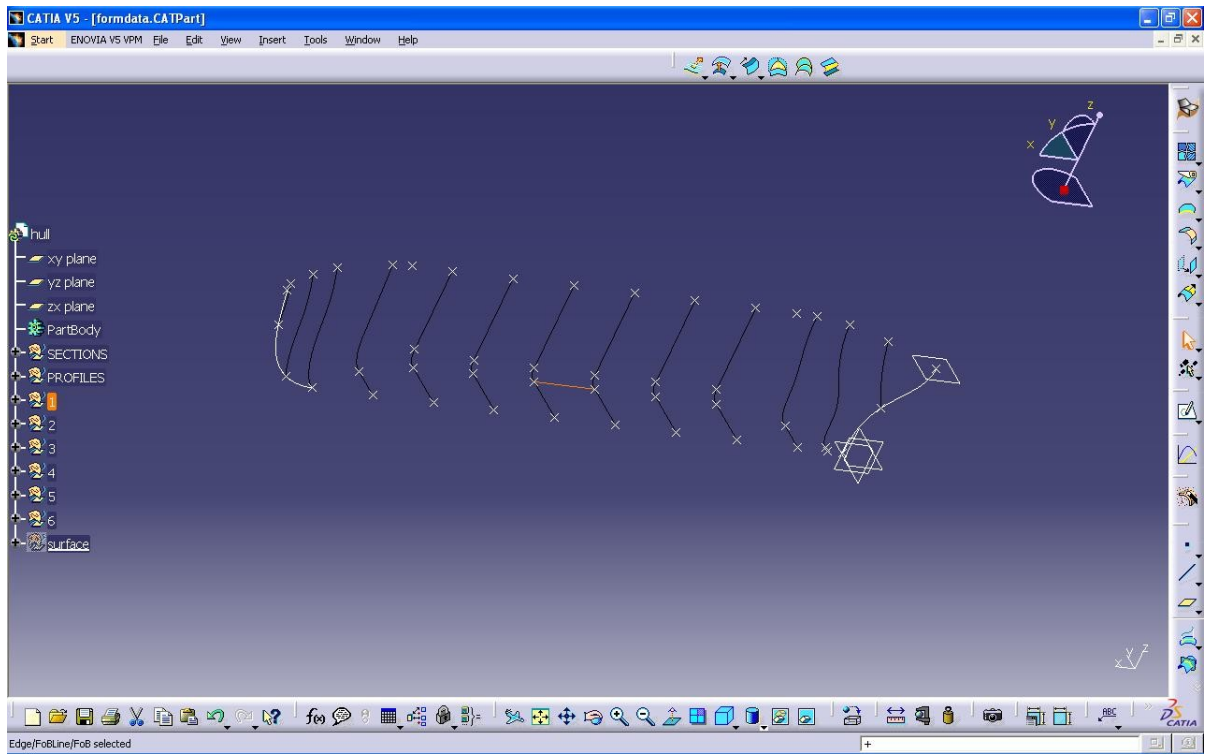


Εικόνα 27c: Δημιουργία των ακραίων κορυφαίων σημείων του FoS

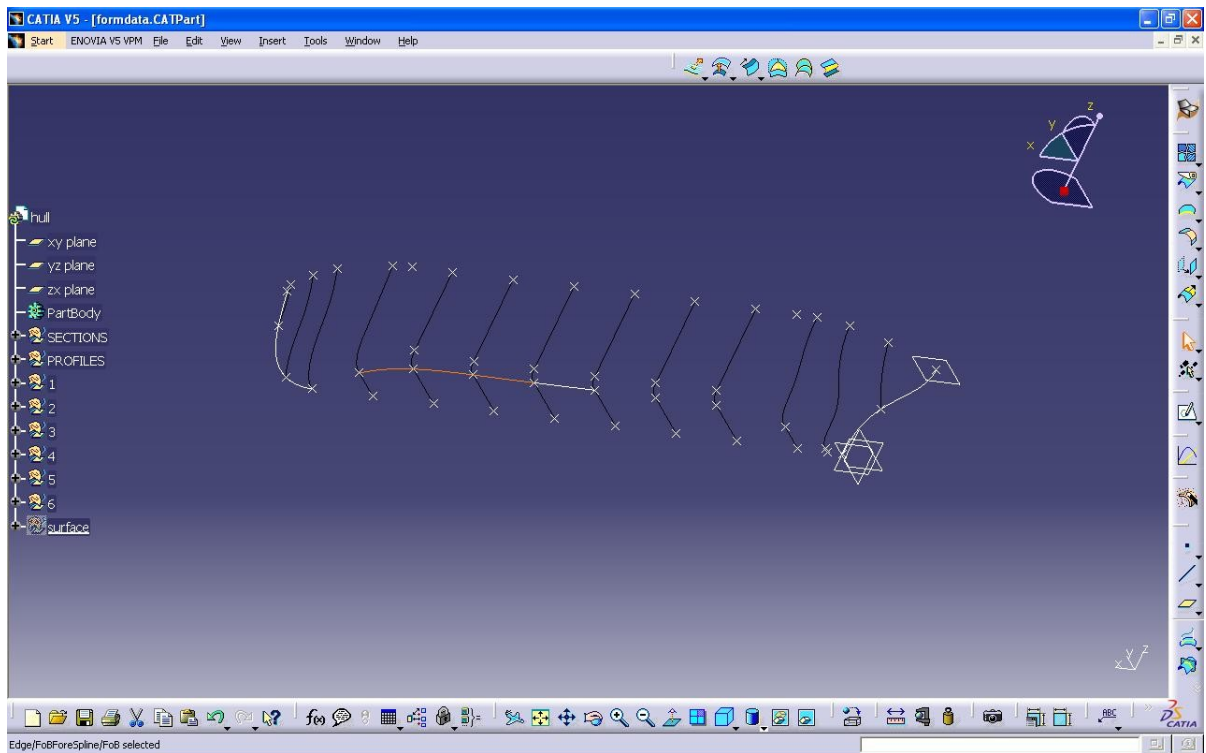
6.4.5. Δημιουργία της καμπύλης του Flat of Bottom

Στη συνέχεια (*block:FoB CREATION*, στις σειρές:173-174 του προγράμματος *final.py* καλείται το *module: FoBCreation* το οποίο περιέχει το *function:FoB*) σχεδιάζεται η καμπύλη που περνάει από τα σημεία του Flat of Bottom και τα άκρα της βρίσκονται επί της Center Line.

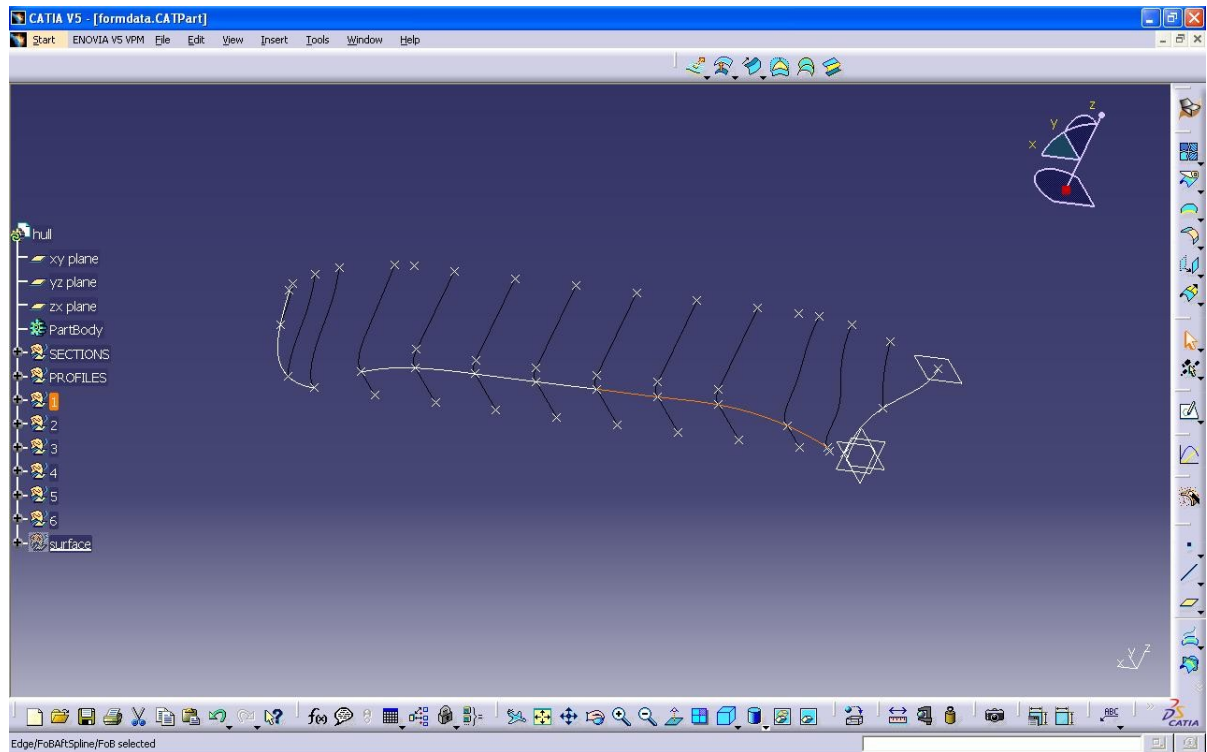
Αρχικά γίνεται έλεγχος για την ύπαρξη συνευθειακών σημείων και εφόσον αυτά υπάρχουν παρεμβάλλονται με μία ευθεία (Εικόνα 28a). Εάν δεν υπάρχουν τότε όλα τα σημεία παρεμβάλλονται με ένα Spline. Στην περίπτωση που σχεδιαστεί ευθεία τότε σχεδιάζονται και δύο Splines (Εικόνες 28b-c), ένα για να παρεμβάλει τα σημεία του Flat of Bottom πρόωραθεν της ευθείας και άλλο ένα για να παρεμβάλει αυτά πρύμνηθεν της ευθείας. Και τα δύο Splines ενώνονται εραπτομενικά με την ευθεία.



Εικόνα 28a: Δημιουργία ευθειογενούς τμήματος της καμπύλης του Flat of Bottom



Εικόνα 28b: Δημιουργία προβαίου τμήματος της καμπύλης του Flat of Bottom



Εικόνα 28c: Δημιουργία πρυμναίου τμήματος της καμπύλης του Flat of Bottom

Σε αυτή τη φάση πρέπει να οριστούν τα σημεία επί της CenterLine στα οποία θα «κλείνει» η καμπύλη του Flat of Bottom. Για την επίτευξη ομαλής καμπύλης στο σύνολο της, τα δύο αυτά σημεία (Εικόνες 28d-e) δημιουργήθηκαν με τον εξής τρόπο:

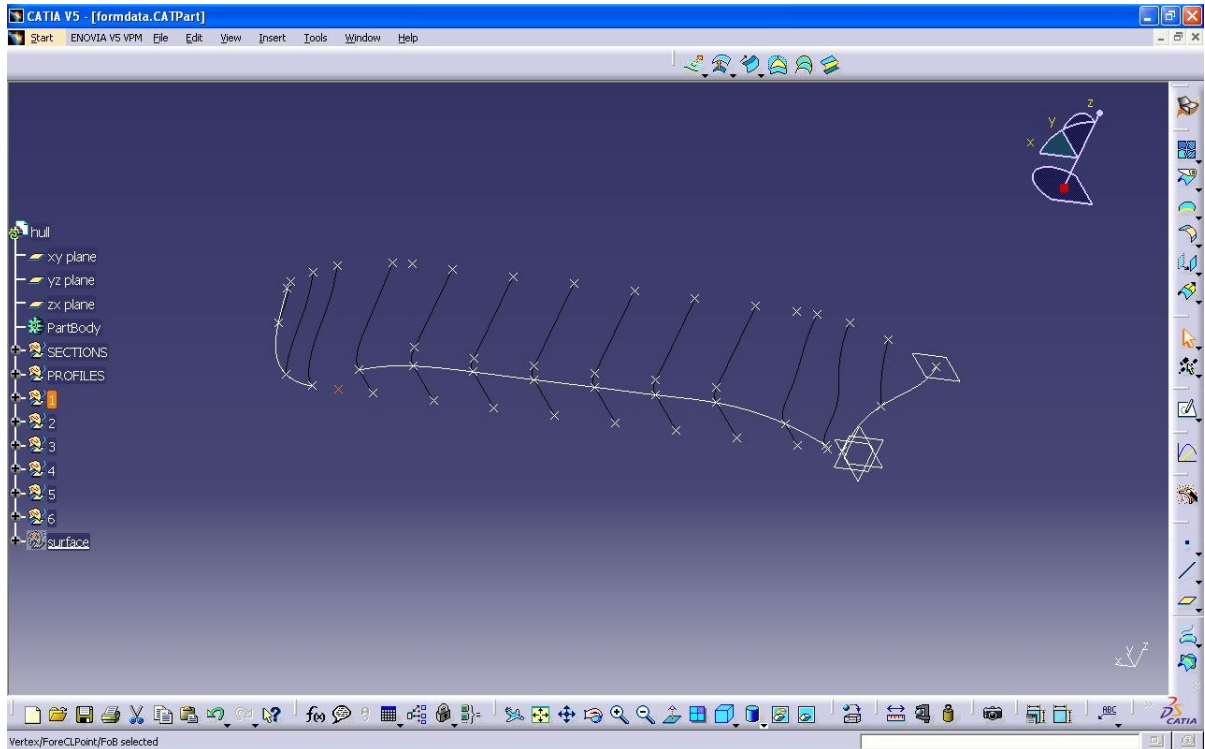
Υπολογίστηκε η Y συντεταγμένη (Ycoord) του προραίου και του πρυμναίου σημείου FoB. Έπειτα σχεδιάστηκε ένα σημείο επί της Center Line πρόραθεν και πρύμνηθεν αντίστοιχα σε απόσταση x από το κατώτατο ακραίο σημείο του αντίστοιχου ακραίου νομέα. Η απόσταση x ορίστηκε ως εξής:

$$\begin{aligned} \text{για } 0.00 < Y_{\text{coord}} \leq 0.25 & \quad \rightarrow \quad x = 4 * Y_{\text{coord}} \\ \text{για } 0.25 < Y_{\text{coord}} \leq 0.50 & \quad \rightarrow \quad x = 2 * Y_{\text{coord}} \\ \text{για } 0.50 < Y_{\text{coord}} \leq 1.00 & \quad \rightarrow \quad x = Y_{\text{coord}} \end{aligned}$$

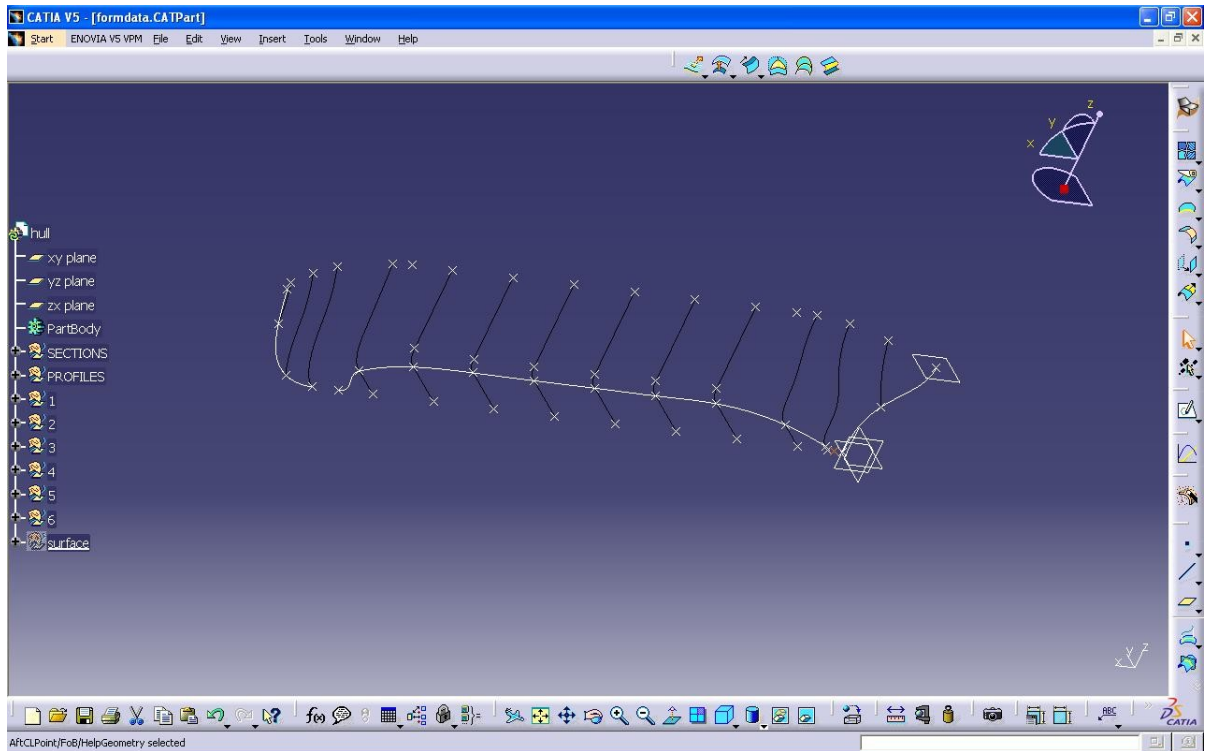
Ακολούθως κατασκευάζονται τα δύο ακραία Splines (Εικόνες 28f-g) για να ολοκληρωθεί η καμπύλη του Flat of Bottom. Στα άκρα και των δύο αυτών Splines επιβλήθηκαν εφαπτομενικά διανύσματα ίδια με αυτά των άκρων της ήδη υπάρχουσας καμπύλης του Flat of Bottom.

Τέλος ενοποιήθηκαν και τα πέντε κομμάτια που απαρτίζουν την καμπύλη σε ένα. Όλα τα παραπάνω γεωμετρικά στοιχεία όπως και η τελική καμπύλη τοποθετήθηκαν στη θέση: **hull**⇒**category**⇒**HelpGeometry**⇒**FoB**

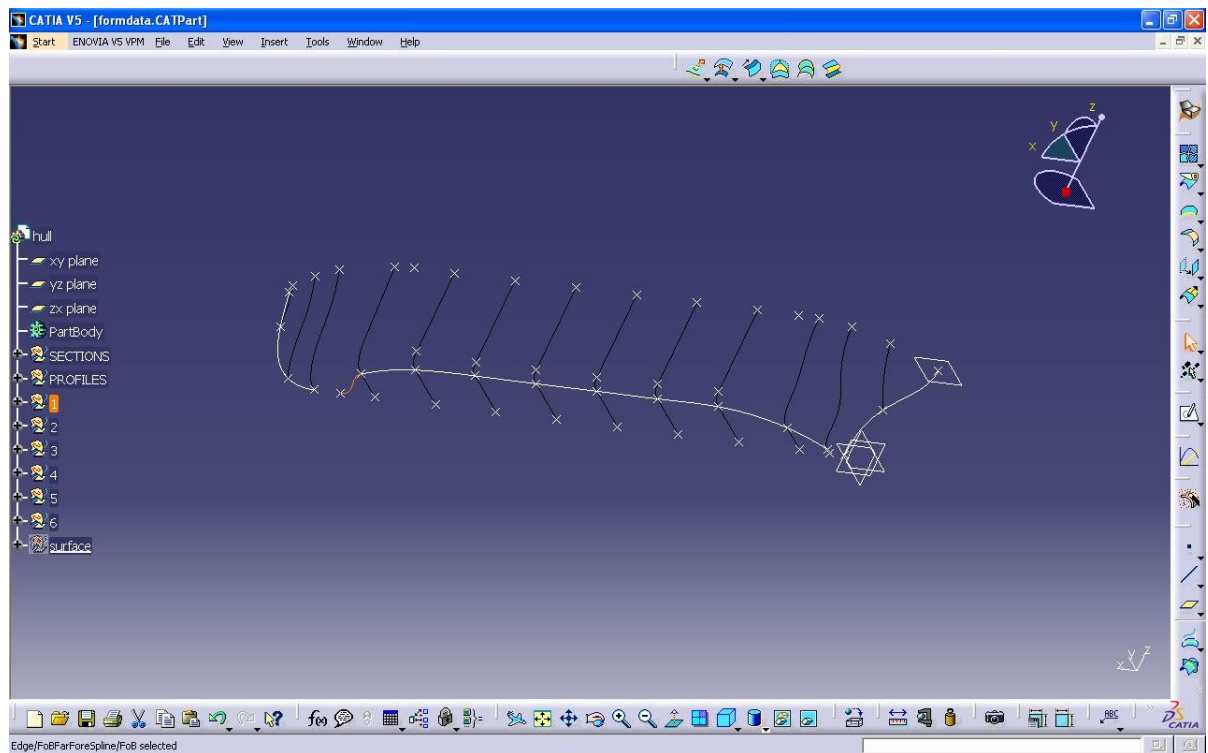
Για την δημιουργία αυτών χρησιμοποιήθηκαν τα **function: AddNewLinePtPt** του **object: Line** και **function: AddNewSpline** του **object: Spline** (Παράρτημα 2).



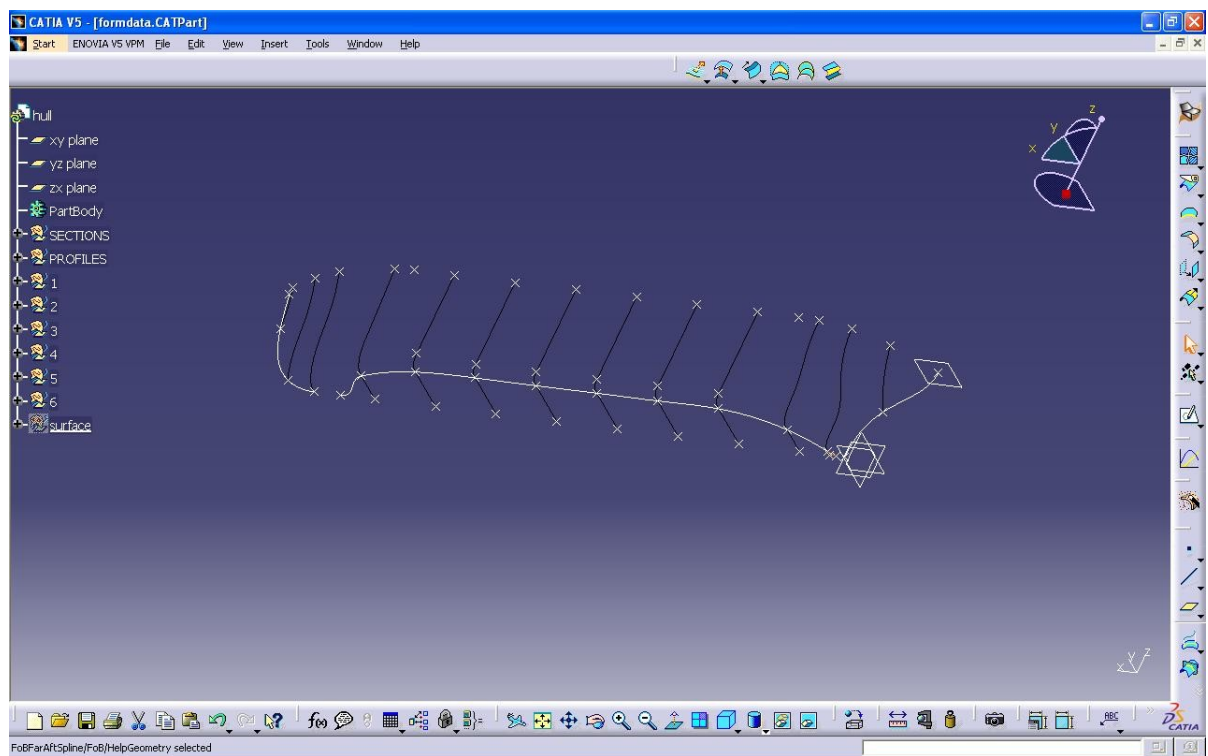
Εικόνα 28d: Δημιουργία προωραίου σημείου κατάληξης της καμπύλης του Flat of Bottom



Εικόνα 28e: Δημιουργία πρυμναίου σημείου κατάληξης της καμπύλης του Flat of Bottom



Εικόνα 28f: Δημιουργία ακραίου προωραίου τμήματος της καμπύλης του Flat of Bottom



Εικόνα 28g: Δημιουργία ακραίου πρυμναίου τμήματος της καμπύλης του Flat of Bottom

6.4.6. Δημιουργία πρύμνης άβακα τύπου D

Στην περίπτωση που η κατηγορία στην οποία ανήκει το υπό κατασκευή τρισδιάστατο μοντέλο είναι είτε η 3 είτε η 6 (έχει δηλαδή πρύμνη άβακα τύπου D) κατασκευάζεται η πρύμνη (*block: TRANSOM STERN DESIGN TYPE D, στις σειρές: 176-289 του προγράμματος final.py*) ως εξής.

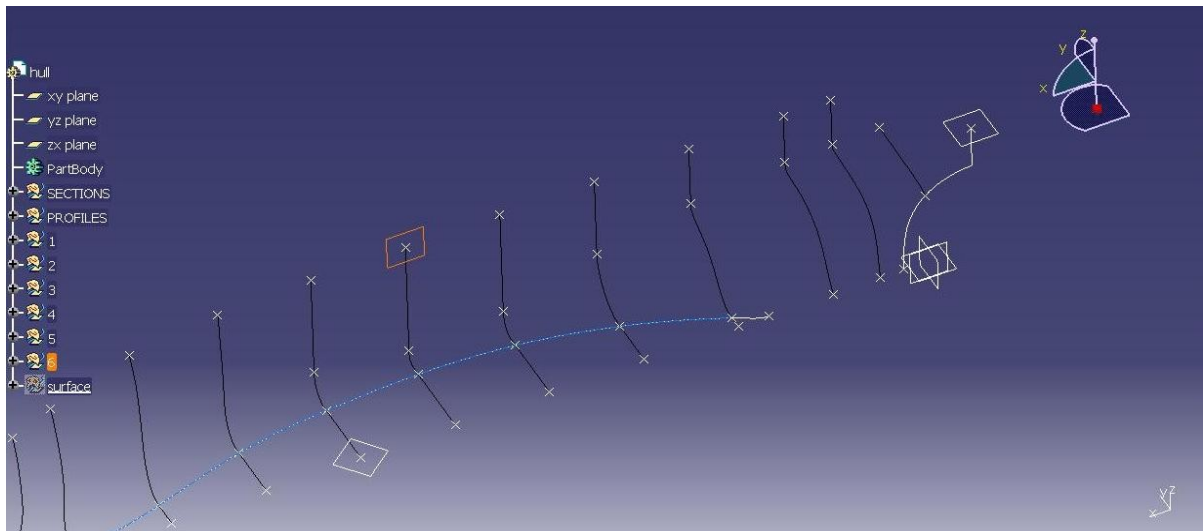
Αρχικά κατασκευάζεται το επίπεδο του flat of side (**FoSPlane**-Εικόνα 29a) και στη συνέχεια σχεδιάζεται μία ευθεία παράλληλη στον Y άξονα (**TransomDeckLine**) με αρχικό σημείο το ανώτατο ακραίο σημείο του προφίλ της πρύμνης μέχρι το επίπεδο του flat of side (Εικόνα 29b). Έπειτα δημιουργείται το ακραίο σημείο αυτής της ευθείας (**DeckLineTransomEndPoint**) κατά τη διεύθυνση του άξονα των Y (Εικόνα 29c). Στη συνέχεια σχεδιάζεται μία ευθεία παράλληλη στον Z άξονα (**0_extention**) με αρχή το ανώτατο ακραίο σημείο του νομέα “0.000” μέχρι το επίπεδο του Deck (Εικόνα 29d) και κατόπιν δημιουργείται και το ανώτατο ακραίο σημείο αυτής (**0_extention_TopPoint** - Εικόνα 29e). Πάνω στην ευθεία που σχεδιάστηκε τελευταία (**0_extention**) και συγκεκριμένα στο 80% του μήκους της και δημιουργείται ένα σημείο (**Move Point**-Εικόνα 29f). Έπειτα από το ακραίο σημείο της TransomDeckLine με διεύθυνση παράλληλη στον άξονα των Z και φορά προς τα κάτω σχεδιάζεται μία ευθεία μέχρι το επίπεδο στο οποίο ανήκει το MovePoint (**TransomVerticalPart**-Εικόνα 29g), καθώς και το κατώτατο ακραίο σημείο αυτής (**TransomVerticalPart_LowerPoint**-Εικόνα 29h). Στη συνέχεια σχεδιάζεται ένα σημείο επί του προφίλ της πρύμνης και συγκεκριμένα ακριβώς στην ακμή του (**TransomPoint**-Εικόνα 29i). Τελικά σχεδιάζεται ένα Spline με αρχή το σημείο **TransomPoint**, όπου και επιβλήθηκε εφαπτομενικό διάνυσμα παράλληλο στον άξονα των Y, και τέλος το σημείο **TransomVerticalPart_LowerPoint** (**TransomSpline**-Εικόνα 29j).

Όλα τα παραπάνω γεωμετρικά στοιχεία τοποθετήθηκαν στη θέση: **hull⇒category⇒HelpGeometry⇒Transom**, εκτός από το FoSPlane που τοποθετήθηκε στη θέση: **hull⇒category⇒HelpGeometry⇒Planes**.

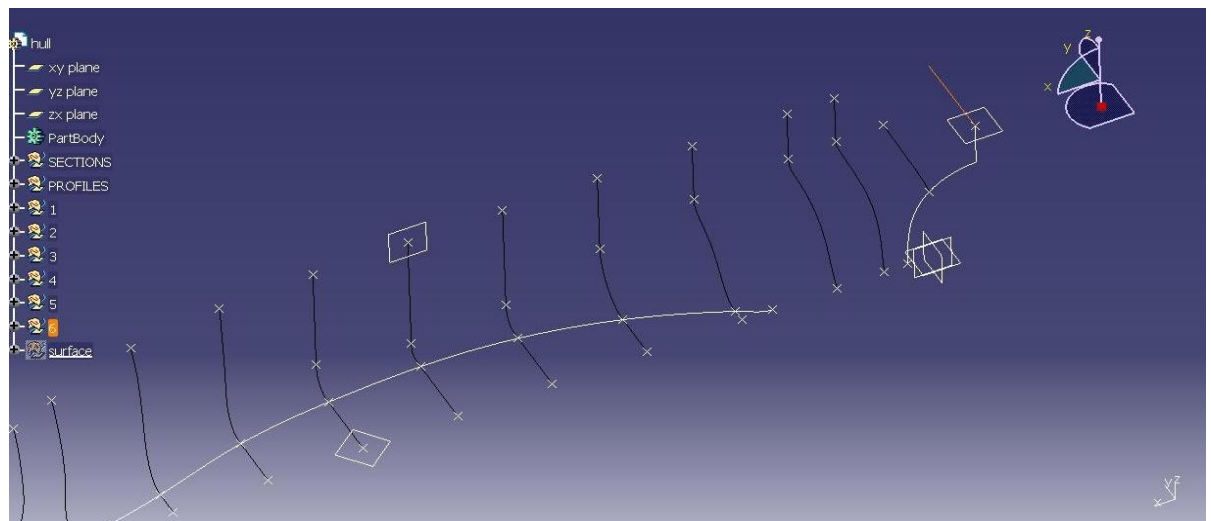
Για την δημιουργία αυτών χρησιμοποιήθηκαν τα:

function: <u>AddNewPlaneOffsetPt</u>	του	object: <u>Plane</u>
function: <u>AddNewExtremum</u>	του	object: <u>Extremum</u>
function: <u>AddNewPointOnCurveFromPercent</u>	του	object: <u>Point</u>
function: <u>AddNewLinePtDir</u>	του	object: <u>Line</u>

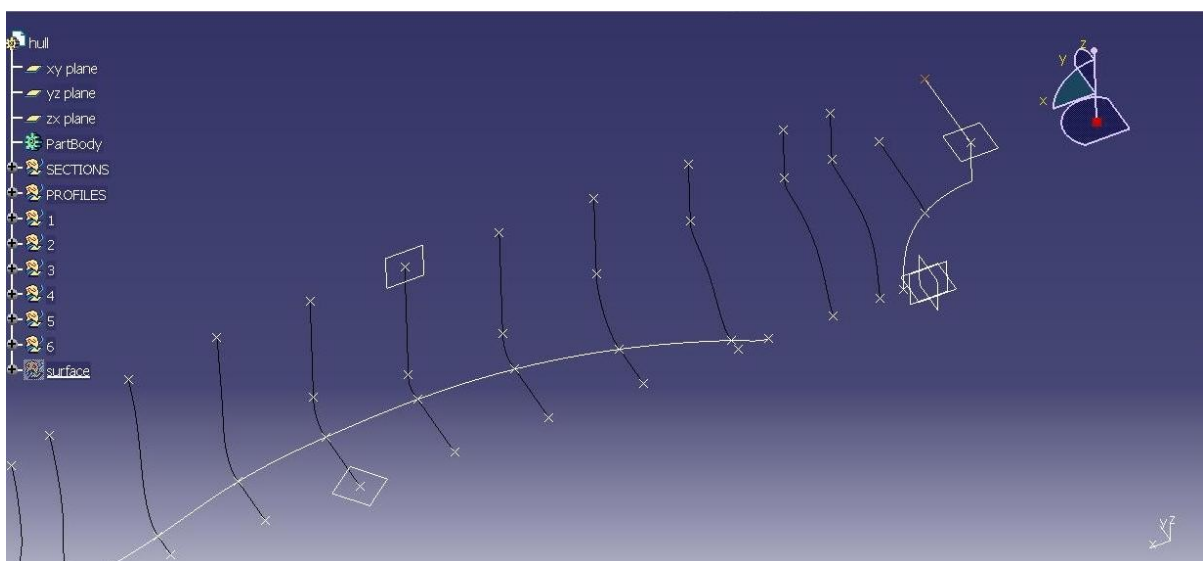
(Παράρτημα 2)



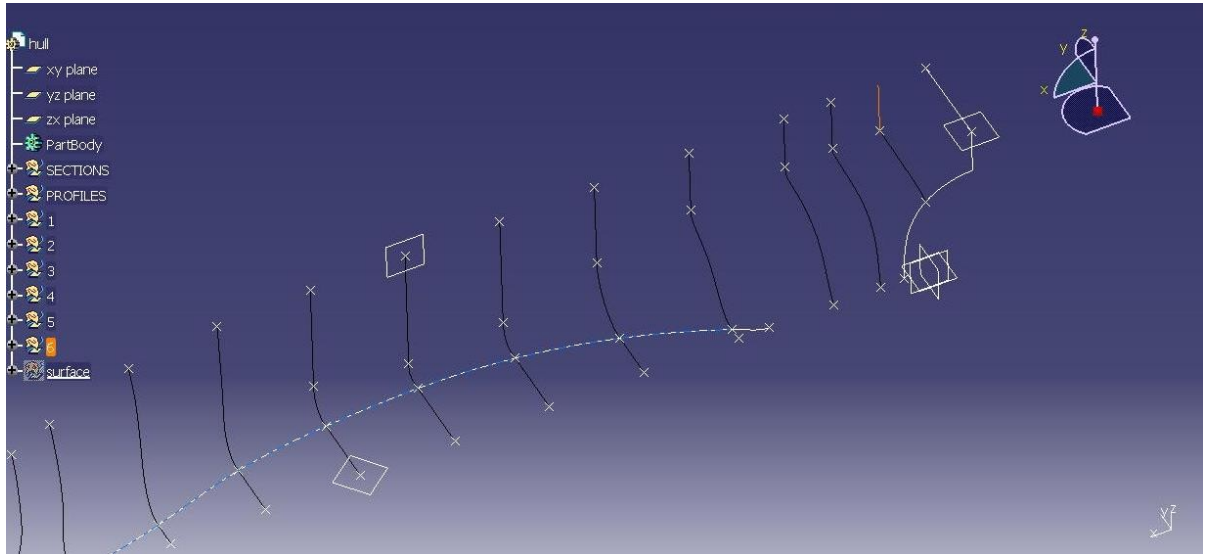
Εικόνα 29α: Κατασκευή του επιπέδου του Flat of Side



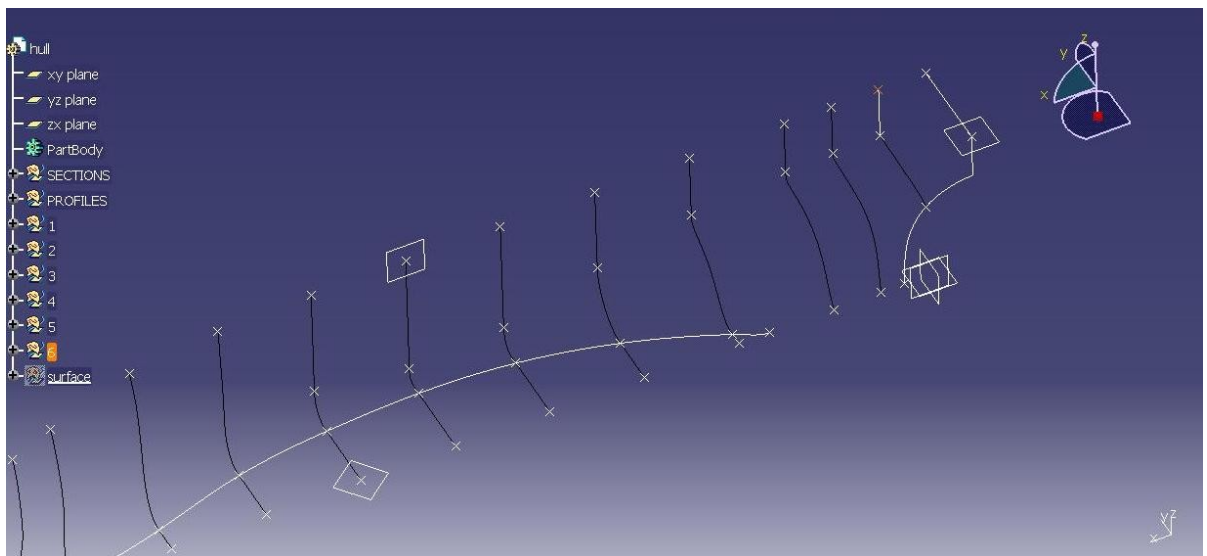
Εικόνα 29b: Κατασκευή της Transom Deck Line



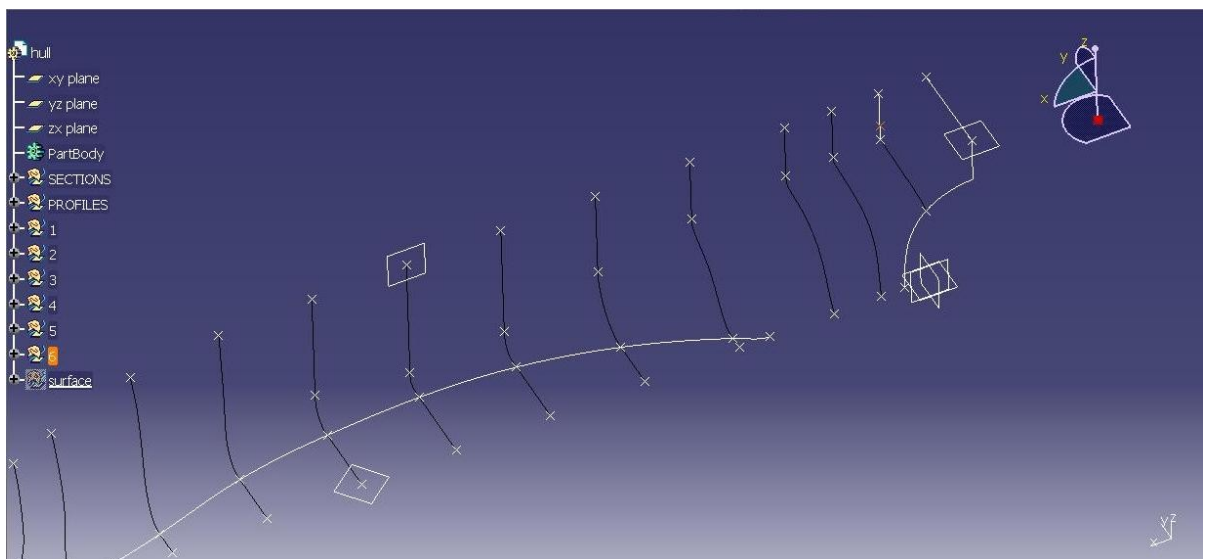
Εικόνα 29c: Κατασκευή του DeckLine Transom End Point



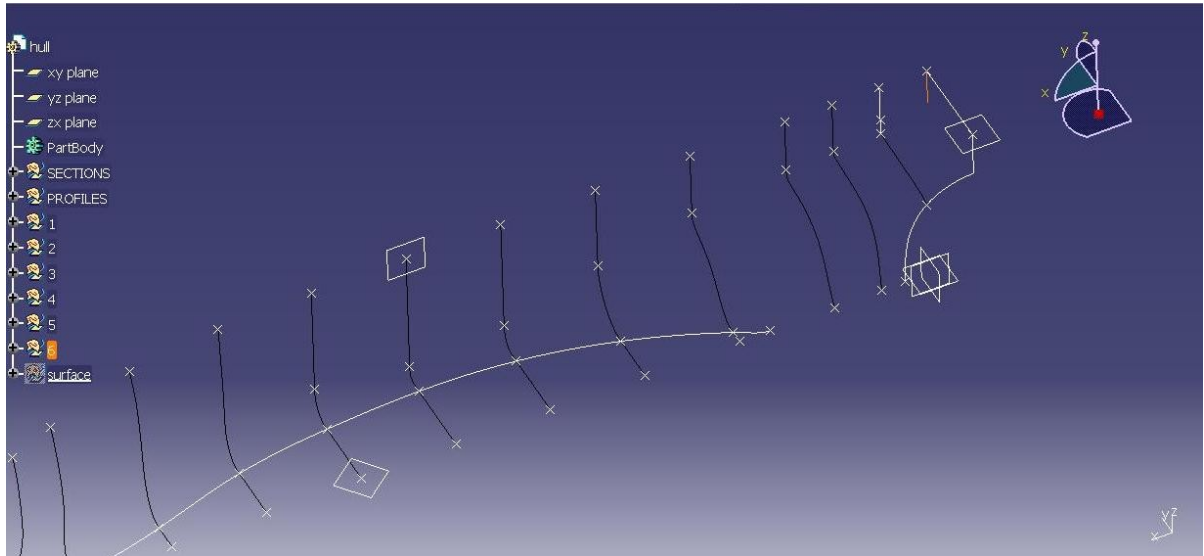
Εικόνα 29d: Κατασκευή της 0_extention



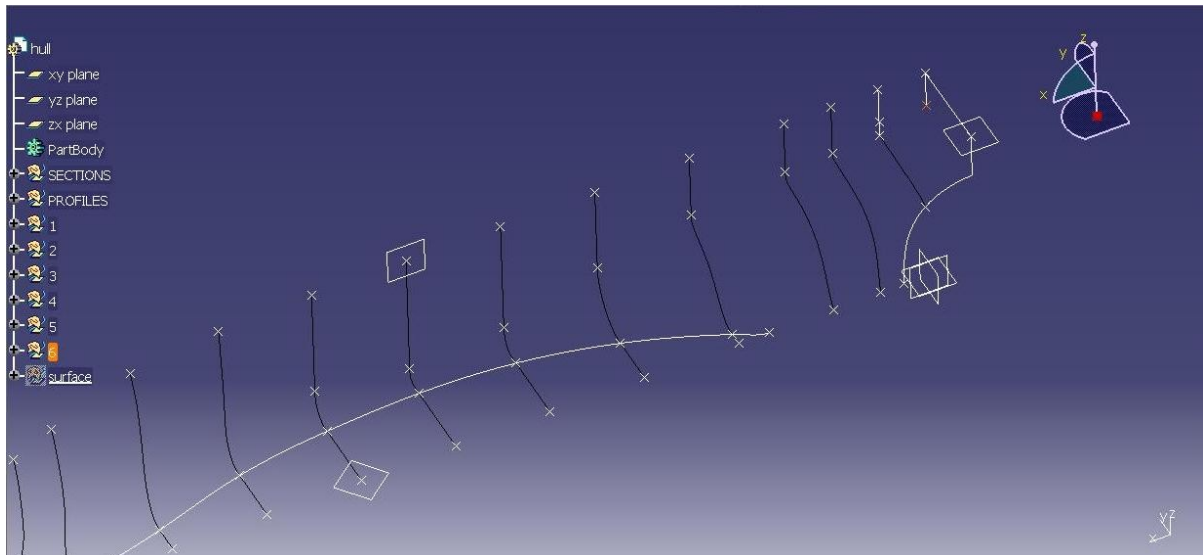
Εικόνα 29e: Κατασκευή του 0_extention Top Point



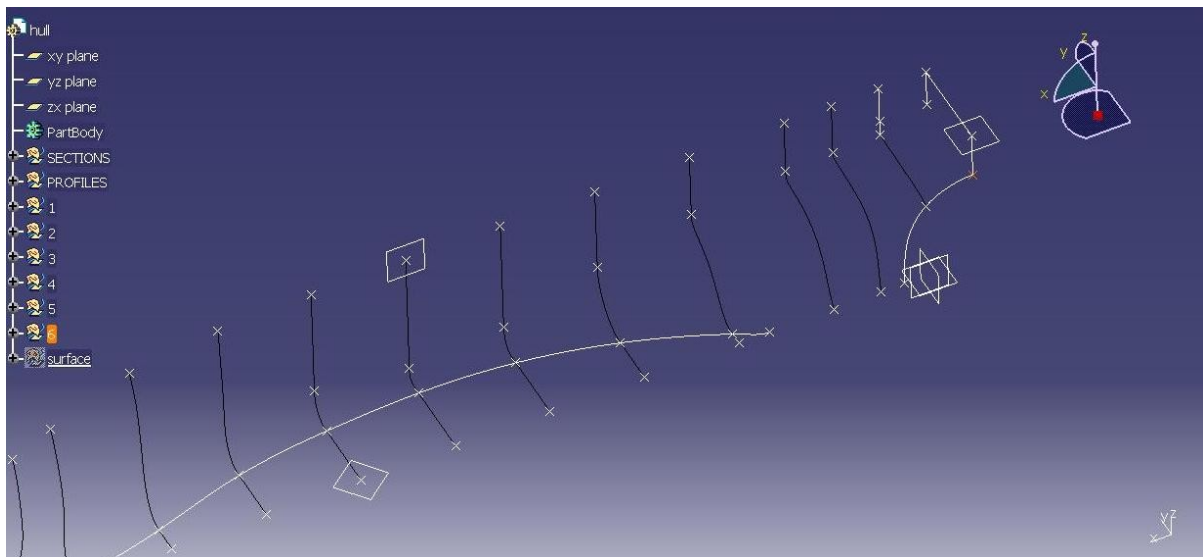
Εικόνα 29f: Κατασκευή του Move Point



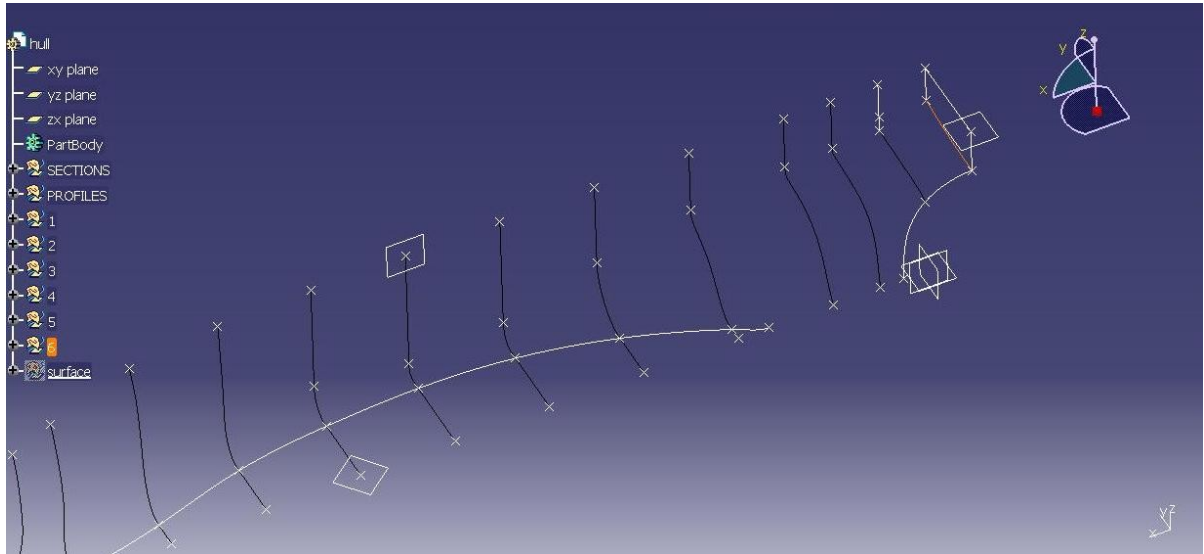
Εικόνα 29g: Κατασκευή της Transom Vertical Part



Εικόνα 29h: Κατασκευή του Transom Vertical Part Lower Point



Εικόνα 29i: Κατασκευή του Transom Point



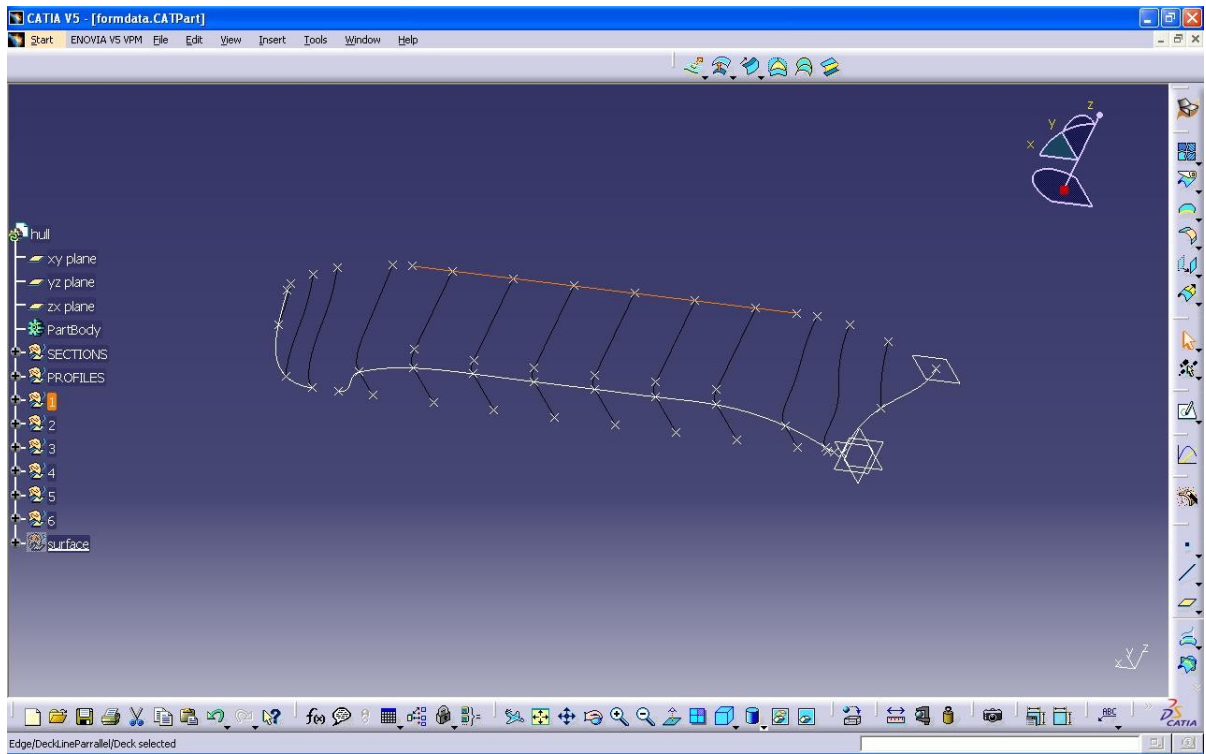
Εικόνα 29j: Κατασκευή του Transom Spline

6.4.7. Δημιουργία της Deckline

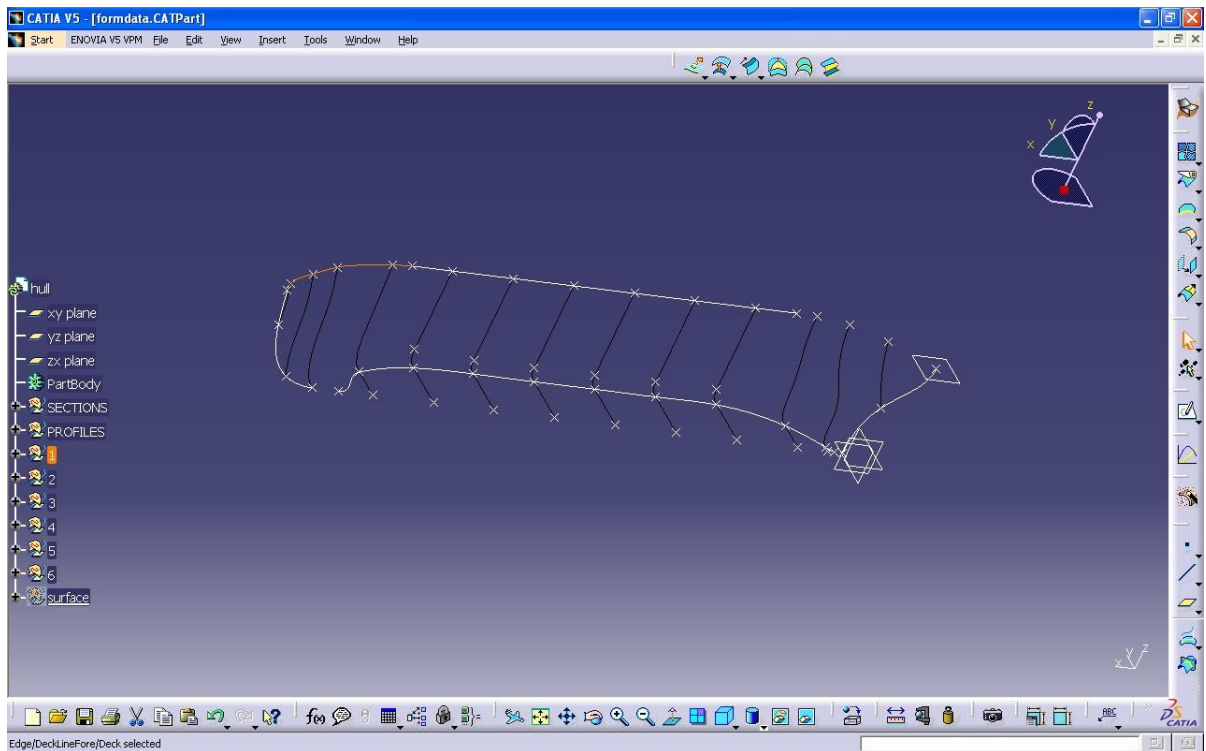
Ακολούθως σχεδιάζεται η Deckline (*block: DeckLine CREATION, στις σειρές: 290-291 του προγράμματος: final.py καλείται το module: DeckLineCreation το οποίο περιέχει το function: DECKLINE*). Διακρίνουμε τις εξής περιπτώσεις ανάλογα με την κατηγορία 1 έως 6 και την fos κατηγορία. Για τις κατηγορίες 1,2,4,5 καθώς και οι υποκατηγορίες τους both,fore,aft,none πρώτα σχεδιάζουμε το παράλληλο τμήμα (Εικόνα 30a) το οποίο ανάλογα την υποκατηγορία έχει και διαφορετικά ακραία σημεία, στη συνέχεια σχεδιάζεται ένα Spline το οποίο παρεμβάλει τα ακραία κορυφαία σημεία των υπόλοιπων προραίων νομέων (Εικόνα 30b) και τέλος σχεδιάζεται άλλο ένα Spline (Εικόνα 30c) που παρεμβάλει τους υπόλοιπους πρυμναίους νομείς. Και στα δύο Splines επιβάλλονται συνθήκες εφαιπτομενικότητας στα άκρα τους. Αντίστοιχη διαδικασία ακολουθείται και για τις κατηγορίες 3 και 6 και τις υποκατηγορίες τους μόνο που σε αυτή την περίπτωση δεν σχεδιάζεται πρυμναίο Spline λόγω της ύπαρξης της πρύμνης άβακα τύπου D το οποίο συνεπάγεται ότι το παράλληλο τμήμα εκτείνεται μέχρι την πρύμνη.

Όλα τα παραπάνω γεωμετρικά στοιχεία όπως και η τελική καμπύλη τοποθετήθηκαν στη θέση: **hull⇒category⇒HelpGeometry⇒Deck**

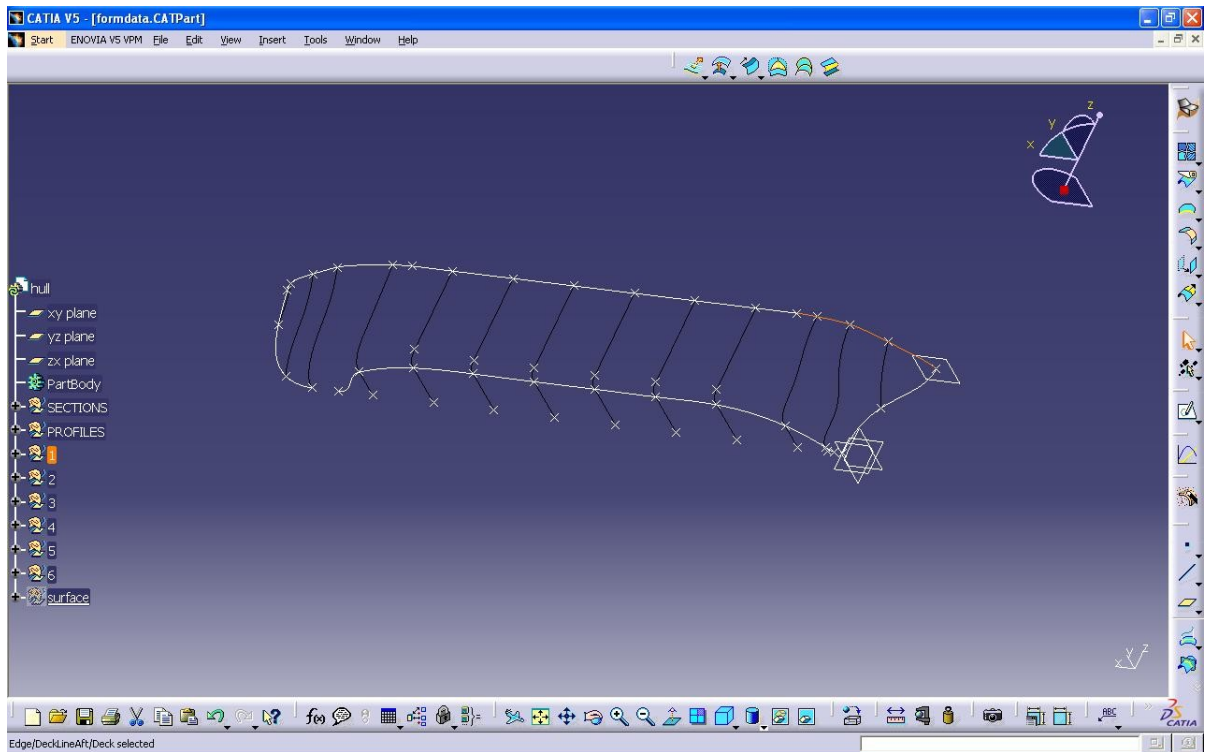
Για την δημιουργία αυτών χρησιμοποιήθηκαν τα **function: AddNewLinePtPt** του **object: Line** και **function: AddNewSpline** του **object: Spline** (Παράρτημα 2).



Εικόνα 30α: Δημιουργία παράλληλου τμήματος



Εικόνα 30b: Δημιουργία προφραίου τμήματος της γραμμής καταστρώματος



Εικόνα 30b: Δημιουργία πρυμναίου τμήματος της γραμμής καταστρώματος

6.4.8. Δημιουργία πρύμνης άβακα τύπου ABC

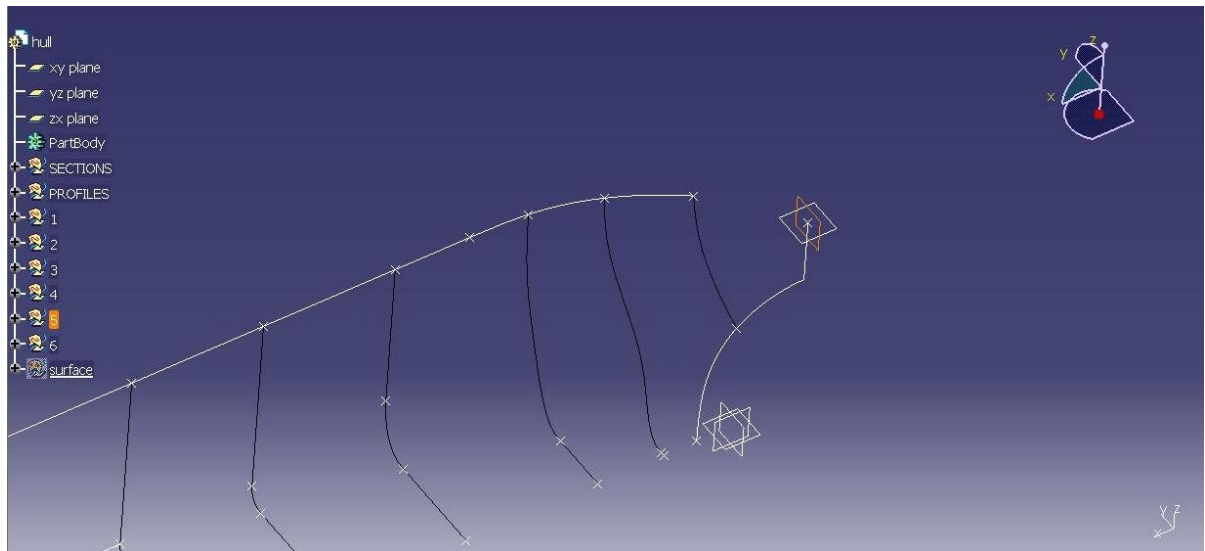
Στην περίπτωση που η κατηγορία στην οποία ανήκει το υπό κατασκευή τρισδιάστατο μοντέλο είναι είτε η 2 είτε η 5 (έχει δηλαδή πρύμνη άβακα τύπου ABC) κατασκευάζεται η πρύμνη (*block: TRANSOM STERN CREATION TYPE ABC, στις σειρές: 293-378 του προγράμματος: final.py*) ως εξής:

Αρχικά κατασκευάζεται το επίπεδο του καθρέφτη (TransomPlane-Εικόνα 31a), ένα επίπεδο παράλληλο στο επίπεδο YZ το οποίο διέρχεται από το κορυφαίο ακραίο σημείο του προφίλ της πρύμνης. Έπειτα σχεδιάζεται μία προέκταση της DeckLineAft από το κορυφαίο ακραίο σημείο του νομέα 0.000 μέχρι το επίπεδο του καθρέφτη (Extrapolation-Εικόνα 31b). Στη συνέχεια σχεδιάζεται το ακραίο σημείο αυτής της προέκτασης κατά τον άξονα των X (ουσιαστικά το σημείο τομής της προέκτασης και του επιπέδου του καθρέφτη, DeckEndPoint-Εικόνα 31c) και κατόπιν άλλο ένα σημείο, πάνω στο πρυμναίο προφίλ και συγκεκριμένα στο κατώτατο σημείο του καθρέφτη (TransomPoint-Εικόνα 31d). Τέλος σχεδιάζεται ένα spline το οποίο παρεμβάλλει τα δύο παραπάνω σημεία και έχει εφαπτομενικά διανύσματα, στα άκρα του, παράλληλα στους άξονες Z και Y αντίστοιχα (TransomSpline-Εικόνα 31e). Επίσης κατασκευάζεται και μία ευθεία με αρχή το ανώτατο ακραίο σημείο του πρυμναίου προφίλ και τέλος το DeckEndPoint (TransomDeckLine-Εικόνα 31f).

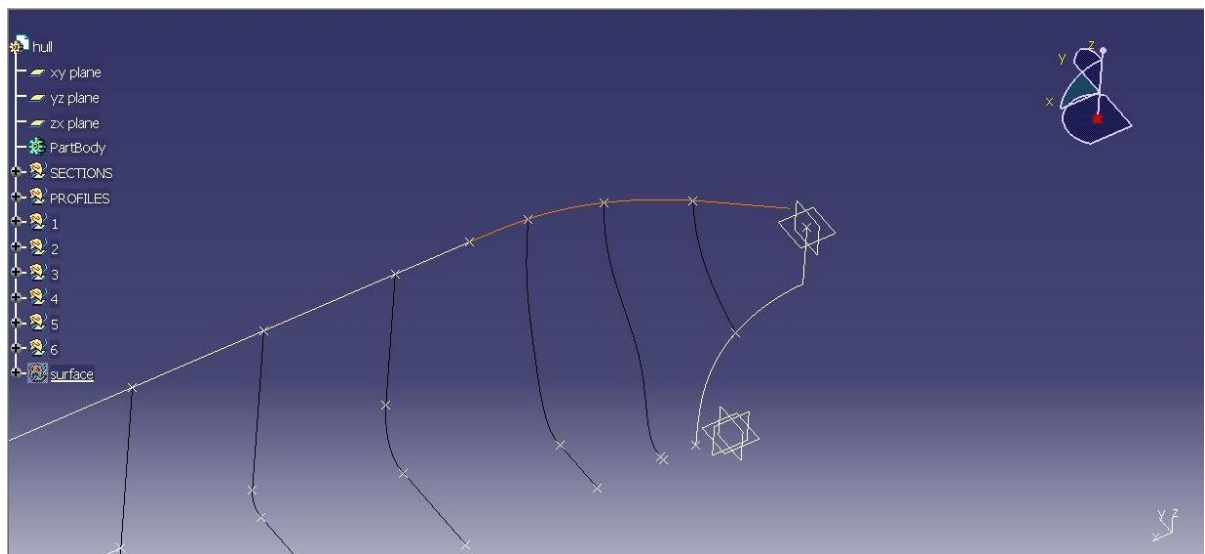
Όλα τα παραπάνω γεωμετρικά στοιχεία τοποθετήθηκαν στη θέση: **hull**⇒**category**⇒**HelpGeometry**⇒**Transom**. Για την δημιουργία αυτών χρησιμοποιήθηκαν τα functions(Παράρτημα 2):

function: AddNewExtrapolUntil του **object:** Extrapol, **function:** AddNewPlaneOffsetPt του **object:** Plane, **function:** AddNewExtremum του **object:** Extremum, **function:**

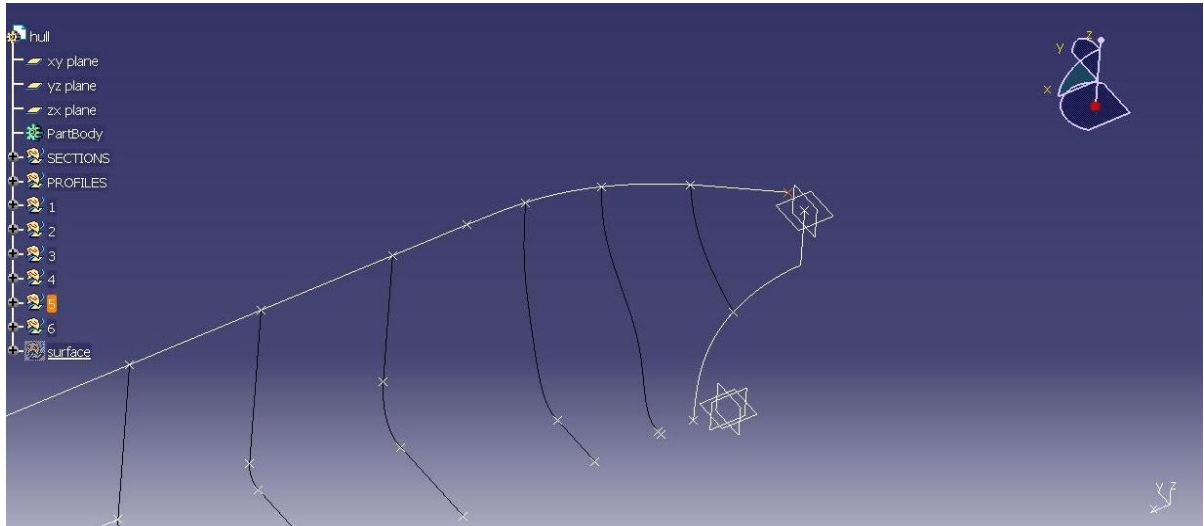
AddNewPointOnCurveFromPercent του **object**: Point, **function**: AddNewSpline του **object**: Spline, **function**: AddNewLinePtPt του **object**: Line.



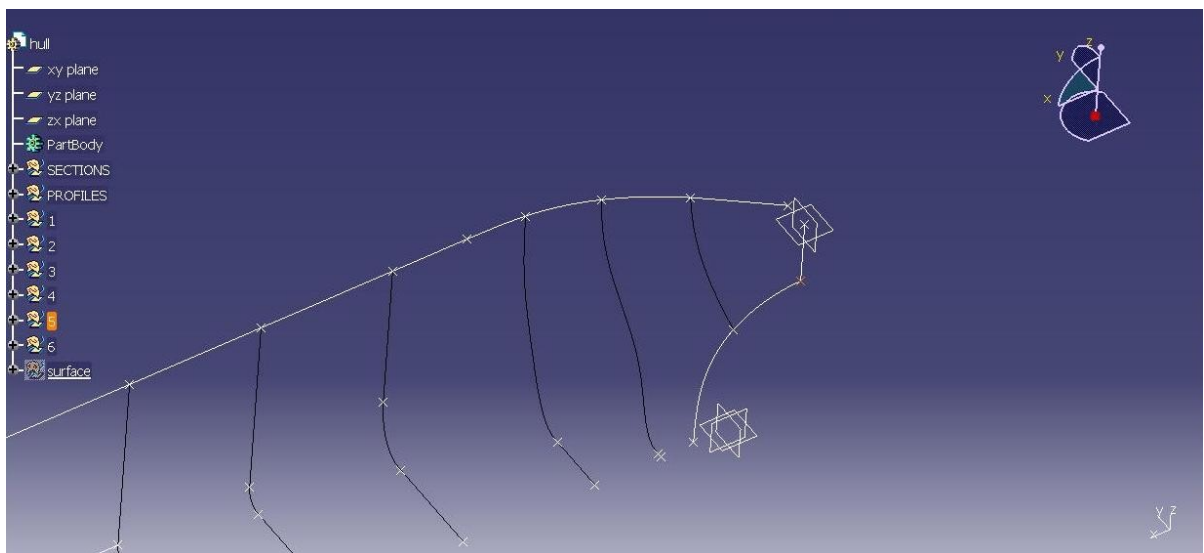
Εικόνα 31a: Δημιουργία του TransomPlane



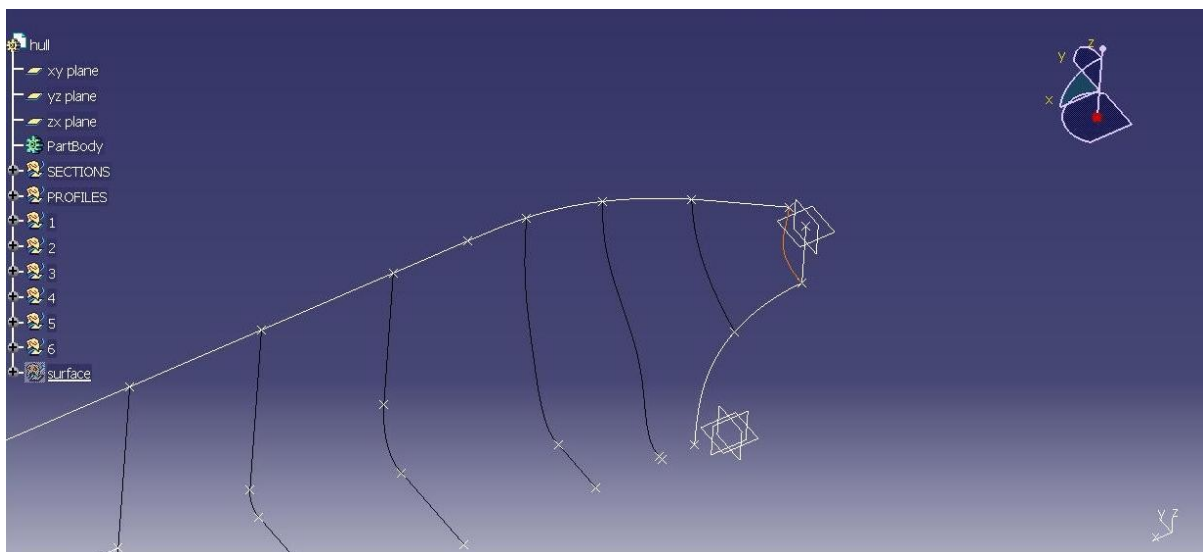
Εικόνα 31b: Δημιουργία της προέκτασης της DeckLineAft



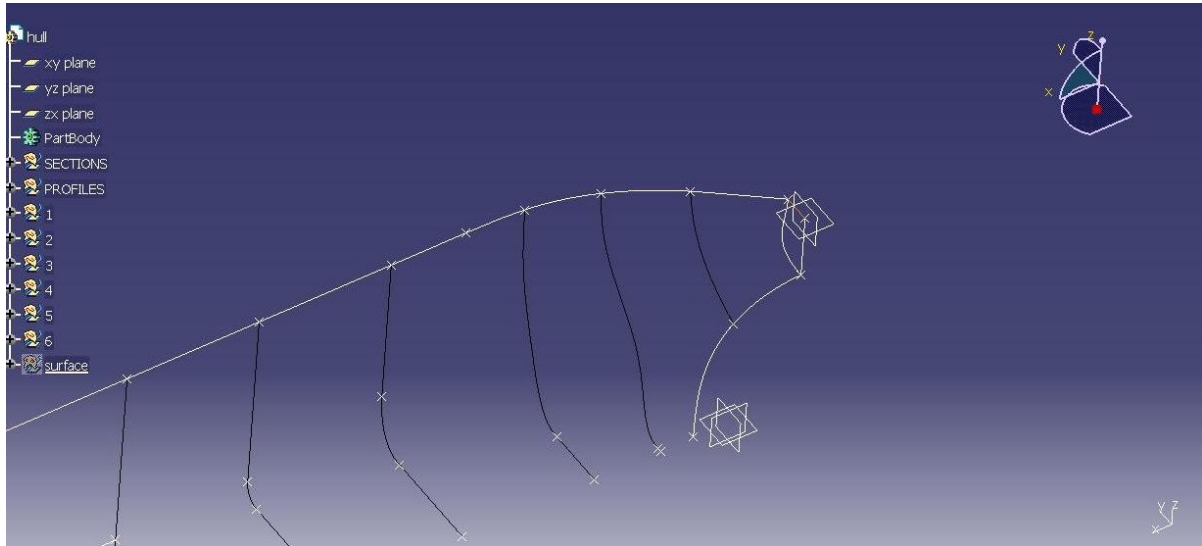
Εικόνα 31c: Δημιουργία ακραίου σημείου της νέας DeckLine



Εικόνα 31d: Δημιουργία του TransomPoint



Εικόνα 31e: Δημιουργία του TransomSpline



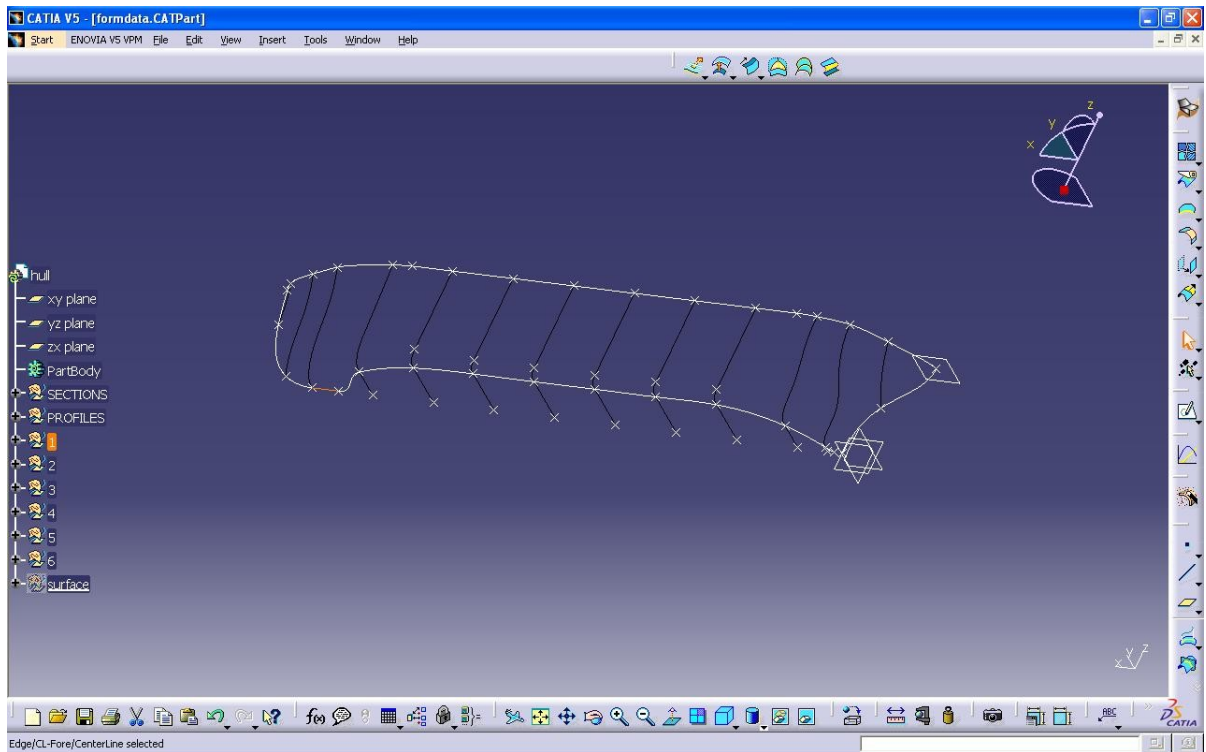
Εικόνα 31f: Δημιουργία της TransomDeckLine

6.4.9. Δημιουργία της Center Line

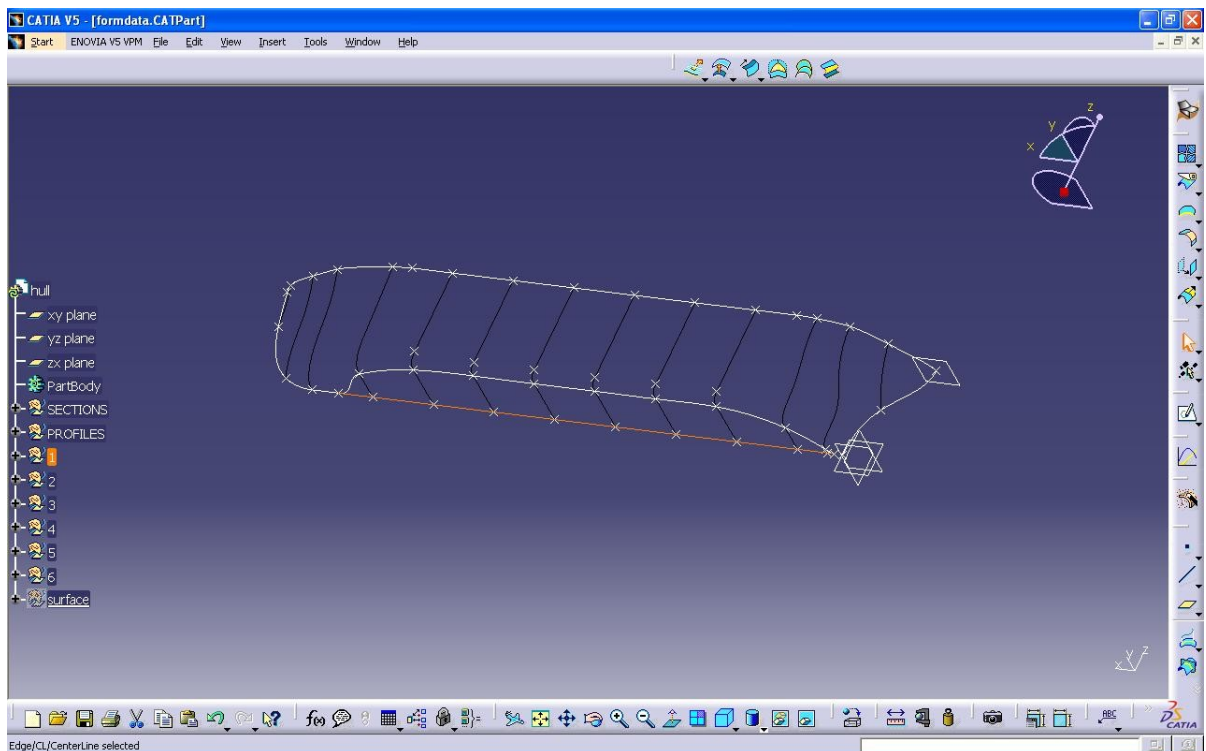
Ακολούθως σχεδιάζεται η Centerline (*block: CenterLine CREATION, στις σειρές: 379-425 του προγράμματος: final.py*). Πρώτα σχεδιάζεται το προωαίο κομμάτι της CL, από το κατώτερο ακραίο σημείο του προφίλ της πλήρης, μέχρι το προωαίο ακραίο σημείο της καμπύλης του Flat of Bottom (Εικόνα 32a). Έπειτα σχεδιάζεται το μεσαίο κομμάτι, από το προωαίο ακραίο σημείο της καμπύλης του Flat of Bottom μέχρι το πρυμναίο ακραίο σημείο της καμπύλης του Flat of Bottom (Εικόνα 32b). Και τελικά σχεδιάζεται το πρυμναίο κομμάτι, από το πρυμναίο ακραίο σημείο της καμπύλης του Flat of Bottom, μέχρι το κατώτερο ακραίο σημείο του προφίλ της πρύμνης (Εικόνα 32c).

Όλα τα παραπάνω γεωμετρικά στοιχεία όπως και η τελική καμπύλη τοποθετήθηκαν στη θέση: **hull⇒category⇒HelpGeometry⇒CenterLine**

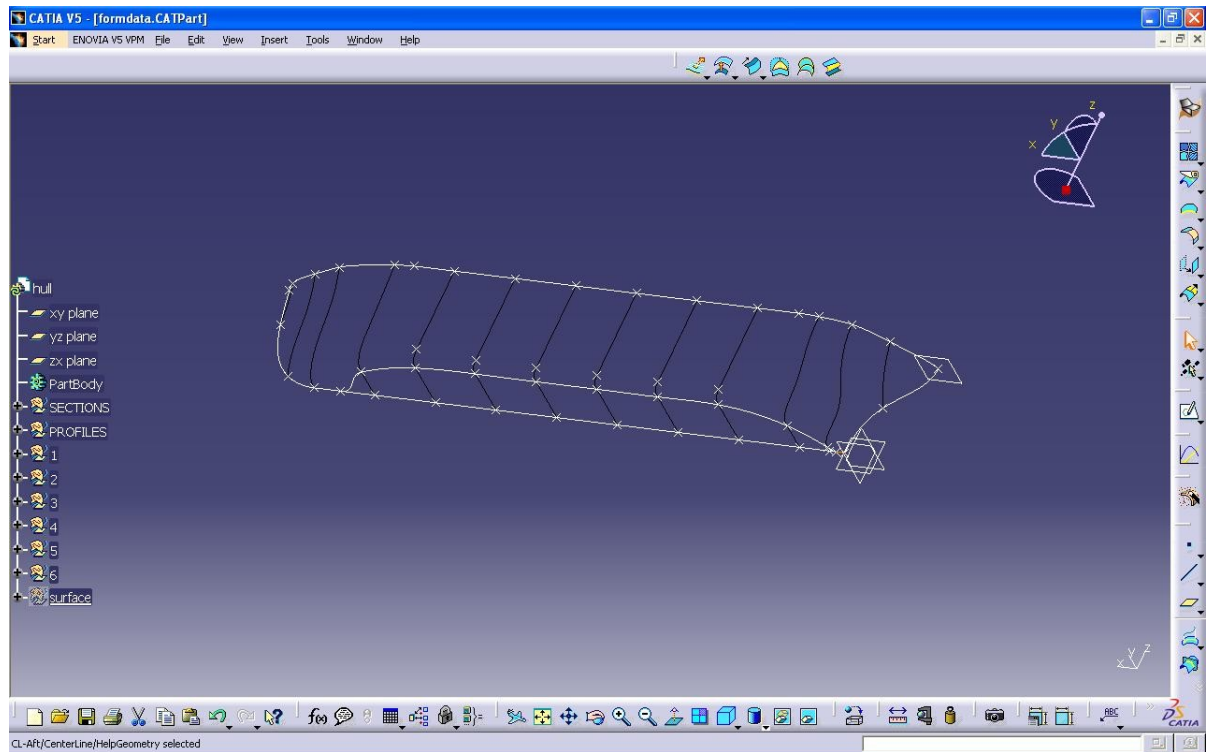
Για την δημιουργία αυτών χρησιμοποιήθηκε το **function: AddNewLinePtPt** του **object: Line** (Παράρτημα 2).



Εικόνα 32α: Δημιουργία προβαίου τμήματος της Center Line



Εικόνα 32b: Δημιουργία μεσαίου τμήματος της Center Line



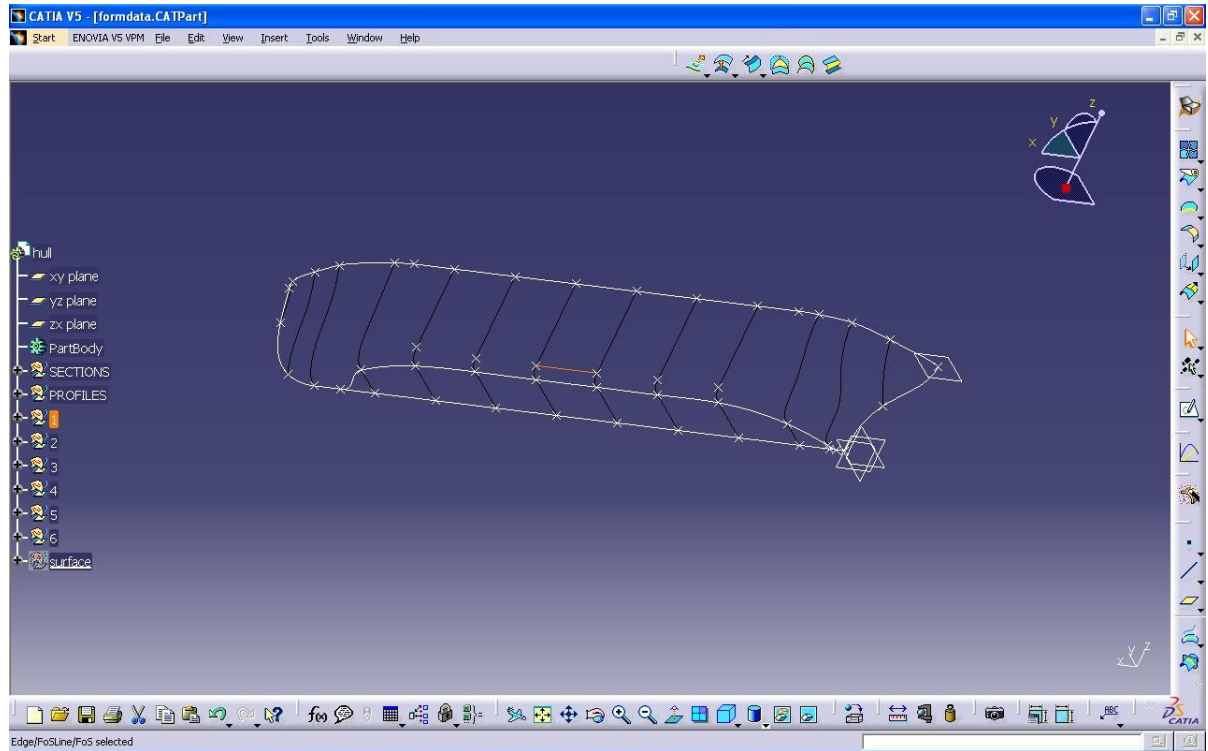
Εικόνα 32c: Δημιουργία πρυμναίου τμήματος της Center Line

6.4.10. Δημιουργία της καμπύλης του Flat of Side

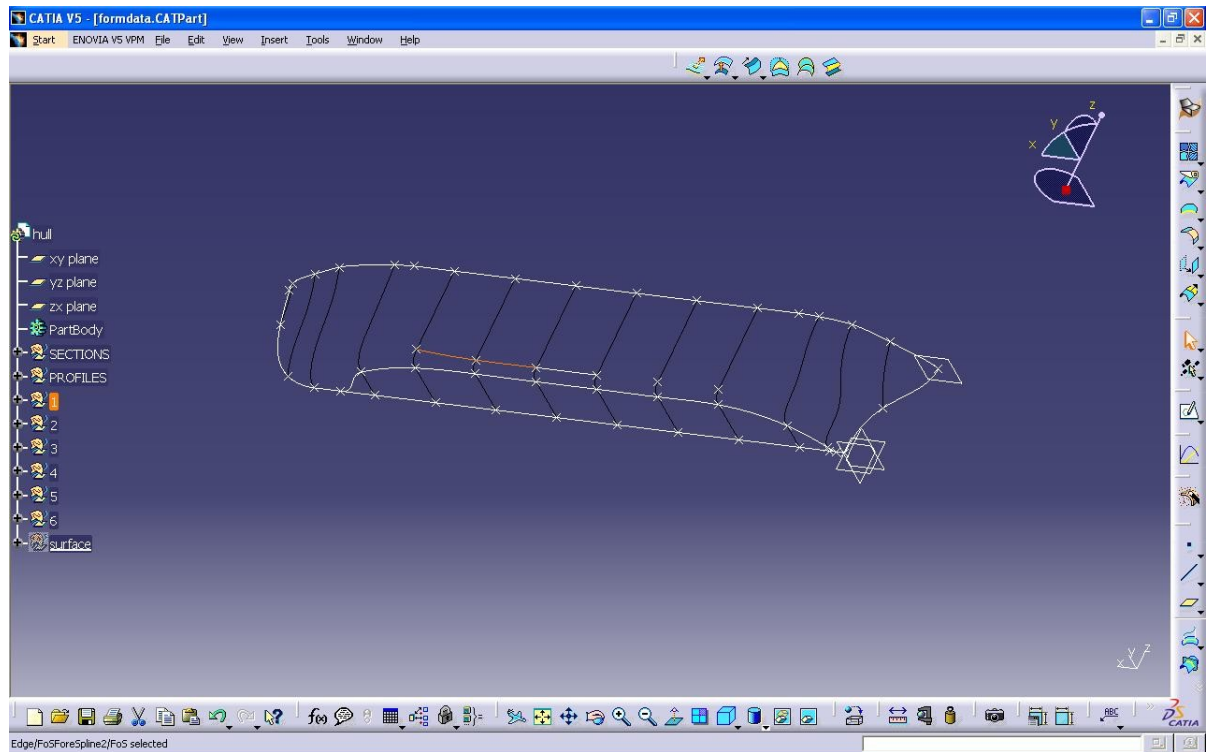
Στη συνέχεια σχεδιάζεται η καμπύλη του Flat of Side (*block: FoS CREATION, στις σειρές: 427-434 του προγράμματος: final.py καλείται το module: FoScreation και τα functions: categories1245, categories36*). Διακρίνονται δύο διαφορετικές κατασκευές ανάλογα με την ύπαρξη πρύμνης άβακα τύπου D στο υπό σχεδίαση μοντέλο.

Στην περίπτωση όπου δεν υπάρχει πρύμνη άβακα τύπου D (πρύμνη καταδρομικού ή άβακας τύπου A,B,C – κατηγορίες 1,2,4,5) καλείται το module: FoScreation και ποιό συγκεκριμένα το function: categories1245, το οποίο είναι υπεύθυνο για την σχεδίαση της καμπύλης του Flat of Side. Πιο συγκεκριμένα αρχικά γίνεται έλεγχος των Z συντεταγμένων των σημείων του flat of side και ελέγχεται ποιές είναι ίσες. Εφόσον βρεθούν σημεία με ίσες Z συντεταγμένες, παρεμβάλλονται με μία ευθεία γραμμή (FoSLine-Εικόνα 33a). Έπειτα δημιουργούμε δύο splines, ένα για να παρεμβάλουμε τους νομείς, που ανήκουν στο flat of side, πρώταθεν της ευθείας που μόλις σχεδιάστηκε (Εικόνα 33b) και άλλο ένα για να παρεμβάλλουμε τους αντίστοιχους νομείς πρύμνηθεν (Εικόνα 33c). Και στα δύο αυτά splines επιβλήθηκαν, στα άκρα τους που ενώνονται με την ευθεία FoSLine, εφαπτομενικά διανύσματα παράλληλα με την FoSLine. Στη συνέχεια σχεδιάζονται τα ακραία τμήματα της καμπύλης του Flat of Side δημιουργώντας δύο splines ένα στην αρχή και ένα στο τέλος της ήδη σχεδιασμένης καμπύλης. Το πρώτο ακραίο spline παρεμβάλλει το πρώτο σημείο του flat of side (fss[0]) και το πρώτο ακραίο κορυφαίο σημείο (FoSForeTopPoint-Εικόνα 33d). Αντίστοιχα το πρυμναίο ακραίο spline παρεμβάλλει το πρυμναίο σημείο του flat of side (fss[-1]) και το πρυμναίο ακραίο κορυφαίο σημείο (FoSAftTopPoint- Εικόνα 33e). Και στα δύο αυτά splines επιβλήθηκαν εφαπτομενικά διανύσματα παράλληλα με την ήδη

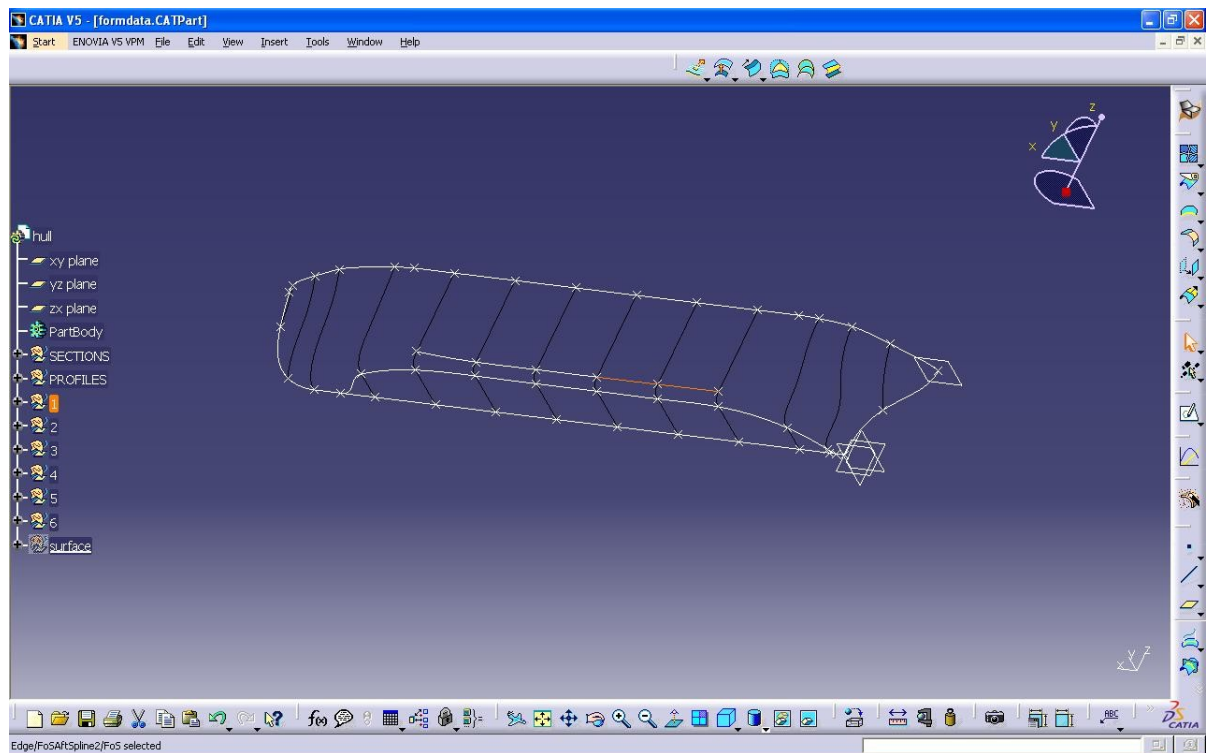
υπάρχουσα καμπύλη του flat of side, έτσι ώστε να διατηρηθεί η C1 γεωμετρική συνέχεια της καμπύλης. Στα άκρα αυτών των spline που εφάπτονται της DeckLine επιβλήθηκαν εφαπτομενικά διανύσματα παράλληλα με αυτά του πρωραίου και πρυμναίου spline της DeckLine. Αυτό συνέβη για να βελτιωθεί η ποιότητα της επιφάνειας που θα δημιουργηθεί αργότερα. Τα εργαλεία του Catia για κατασκευή επιφανειών που μπορούσαμε να χρησιμοποιήσουμε παρήγαγαν καλύτερη επιφάνεια στο συγκεκριμένο σημείο, αν διαμορφωνόταν έτσι η καμπύλη του Flat of Side.



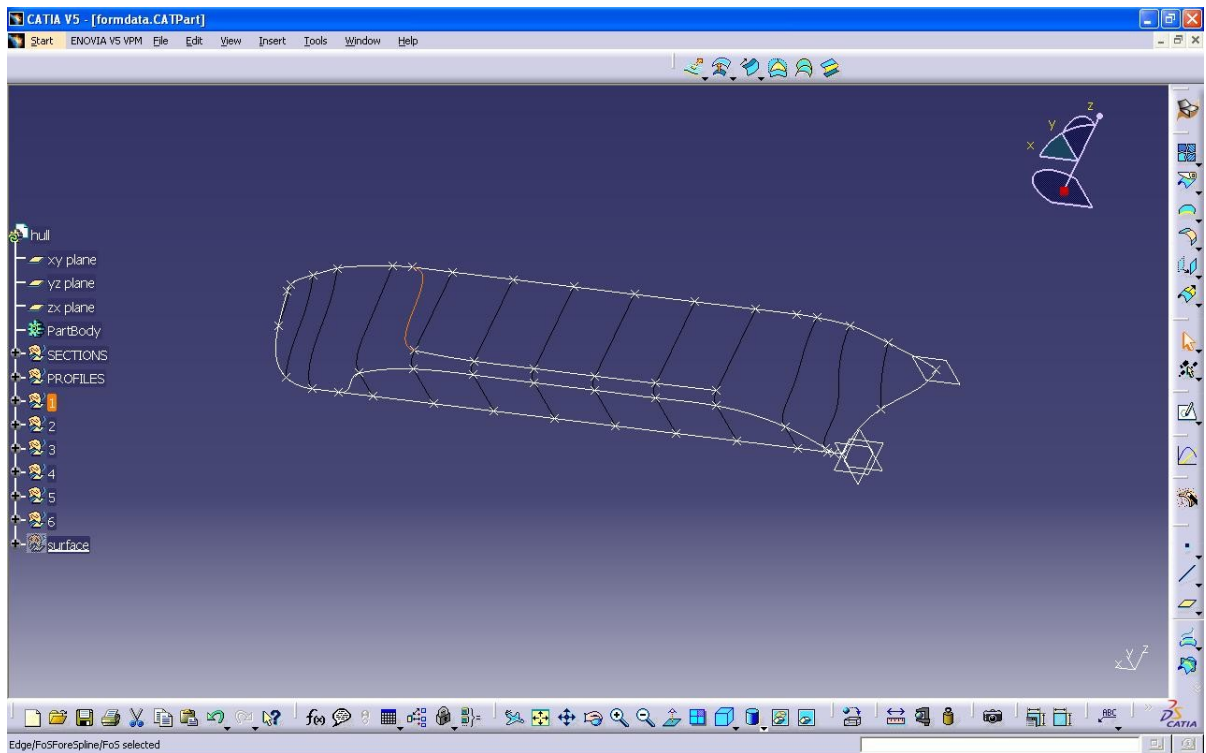
Εικόνα 33α: Κατασκευή ευθιογενούς τμήματος της καμπύλης του Flat of Side (πρύμνη καταδρομικού)



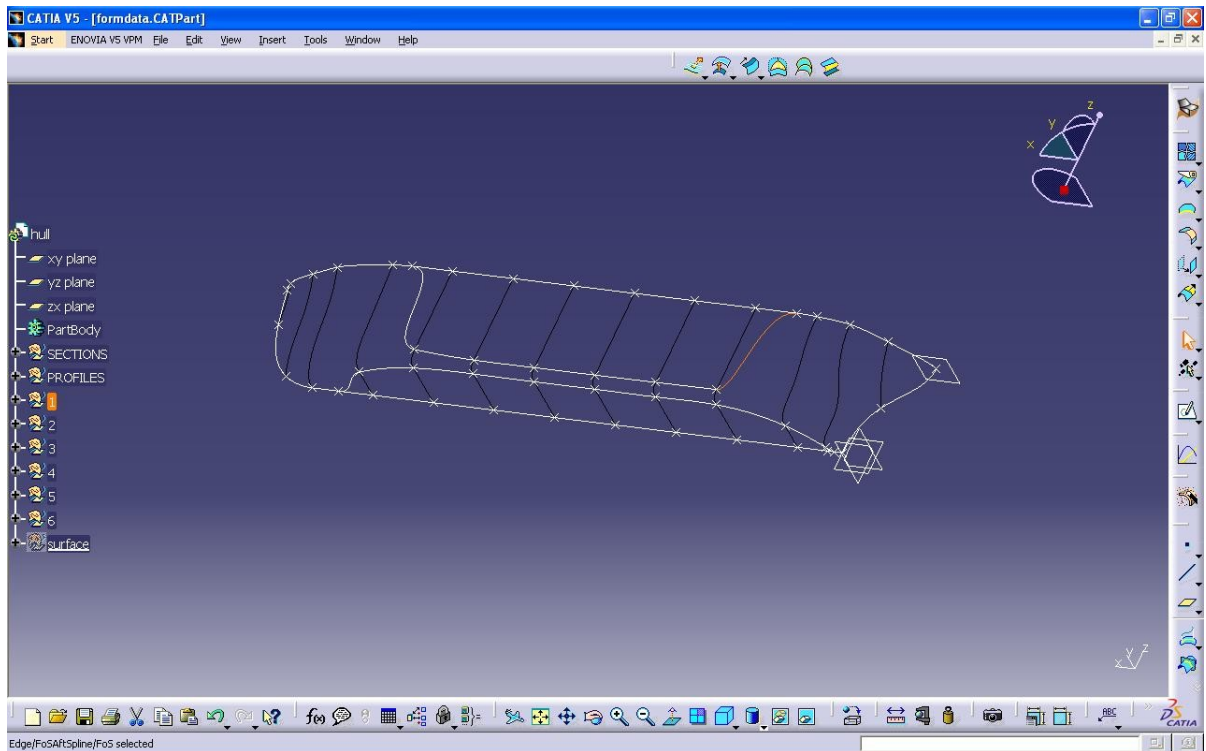
Εικόνα 33b: Κατασκευή προβαίου spline της καμπύλης του Flat of Side (πρύμνη καταδρομικού)



Εικόνα 33c: Κατασκευή πρυμναίου spline της καμπύλης του Flat of Side (πρύμνη πρύμνη καταδρομικού)



Εικόνα 33d: Κατασκευή προωραίου ακραίου spline της καμπύλης του Flat of Side (πρύμνη πρύμνη καταδρομικού)

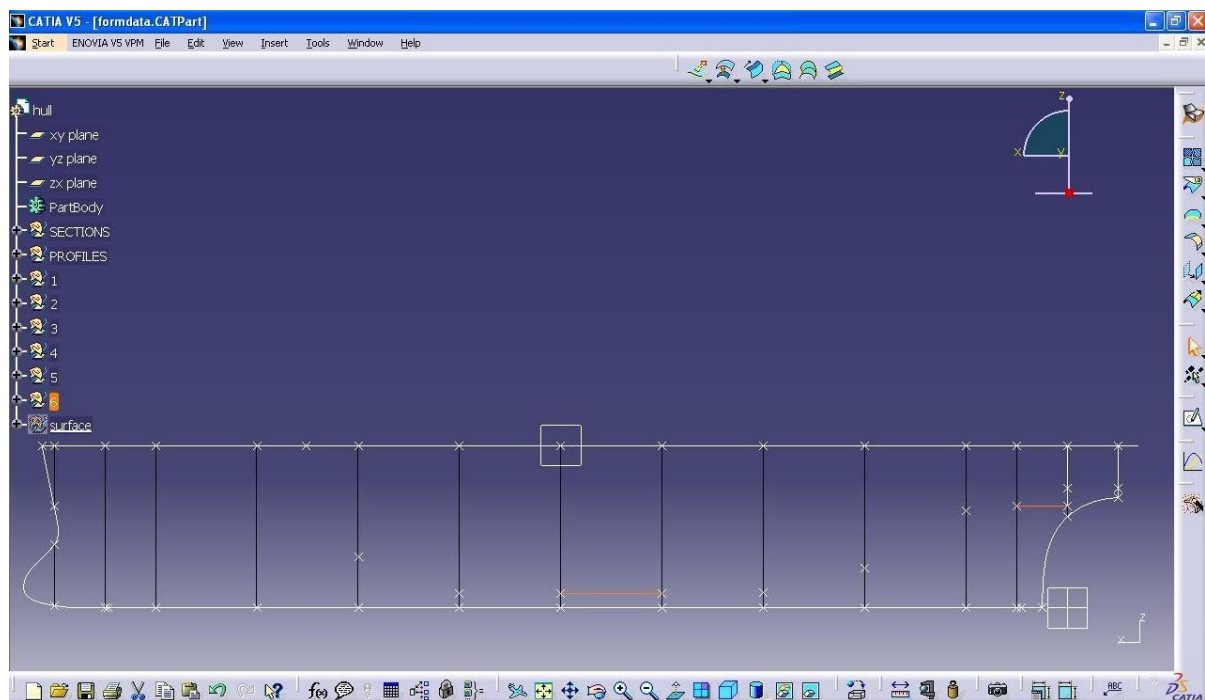


Εικόνα 33ε: Κατασκευή πρυμναίου ακραίου spline της καμπύλης του Flat of Side (πρύμνη καταδρομικού)

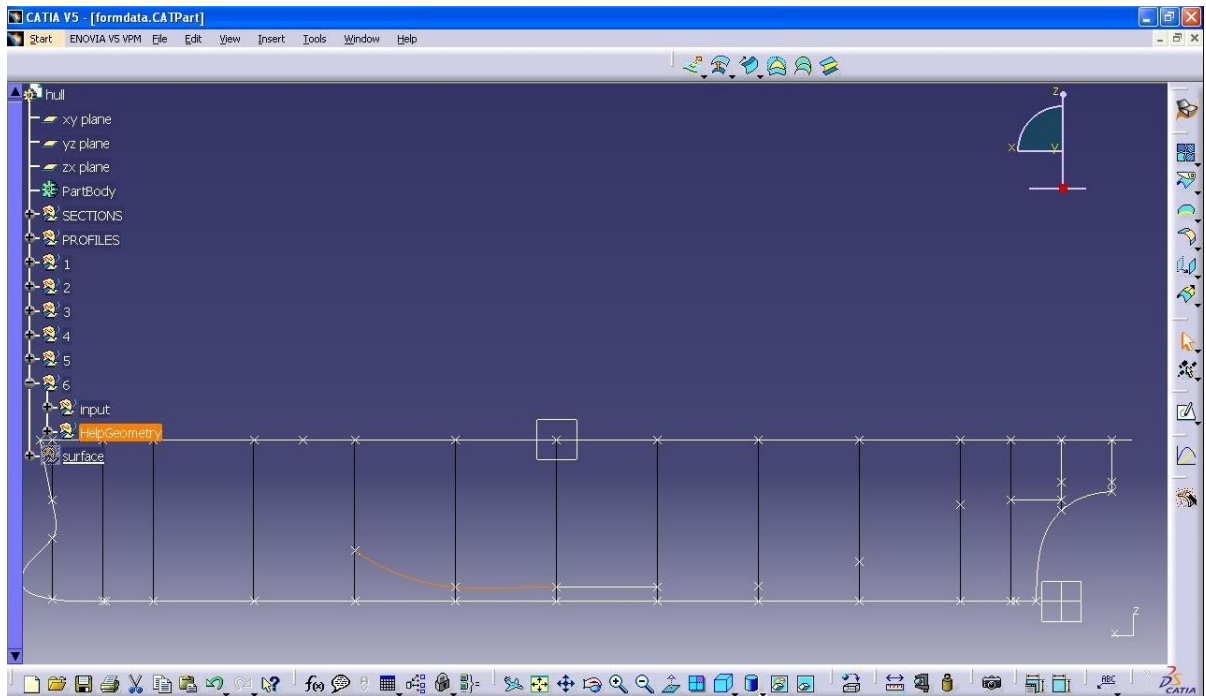
Για την περίπτωση όπου το υπό σχεδίαση μοντέλο διαθέτει πρύμνη άβακα τύπου D, η σχεδίαση της καμπύλης του flat of side διαφοροποιείται από την προηγούμενη περίπτωση ως εξής. Σε αυτή την περίπτωση είναι πιθανόν να υπάρχουν δύο ευθειογενή τμήματα στην καμπύλη του flat of side ,συνεπώς σχεδιάζονται δύο ευθείες (FoSLine1,FoSLine2-Εικόνα 33f),εφόσον υπάρχουν. Στη συνέχεια σχεδιάζεται το πρωαίο spline (Εικόνα 33g) , καθώς και το πρυμναίο spline (Εικόνα 33i), όπως και στην προηγούμενη περίπτωση. Εφόσον υπάρχουν δύο ευθείες είναι απαραίτητο να δημιουργηθεί άλλο ένα spline το οποίο θα παρεμβάλλει τα σημεία ανάμεσα στις δύο ευθείες (MiddleFssSpline-Εικόνα 33h). Τέλος δημιουργείται, όπως και πριν το ακραίο πρωαίο spline(Εικόνα 33j).Λόγω της διαμόρφωσης της πρύμνης άβακα τύπου D δεν χρειάζεται να σχεδιαστεί πρυμναίο ακραίο spline της καμπύλης του Flat of Side.

Όλα τα αντικείμενα που δημιουργήθηκαν όπως περιγράφηκε παραπάνω, τοποθετήθηκαν στη θέση: **hull**⇒**category**⇒**HelpGeometry**⇒**FoS**

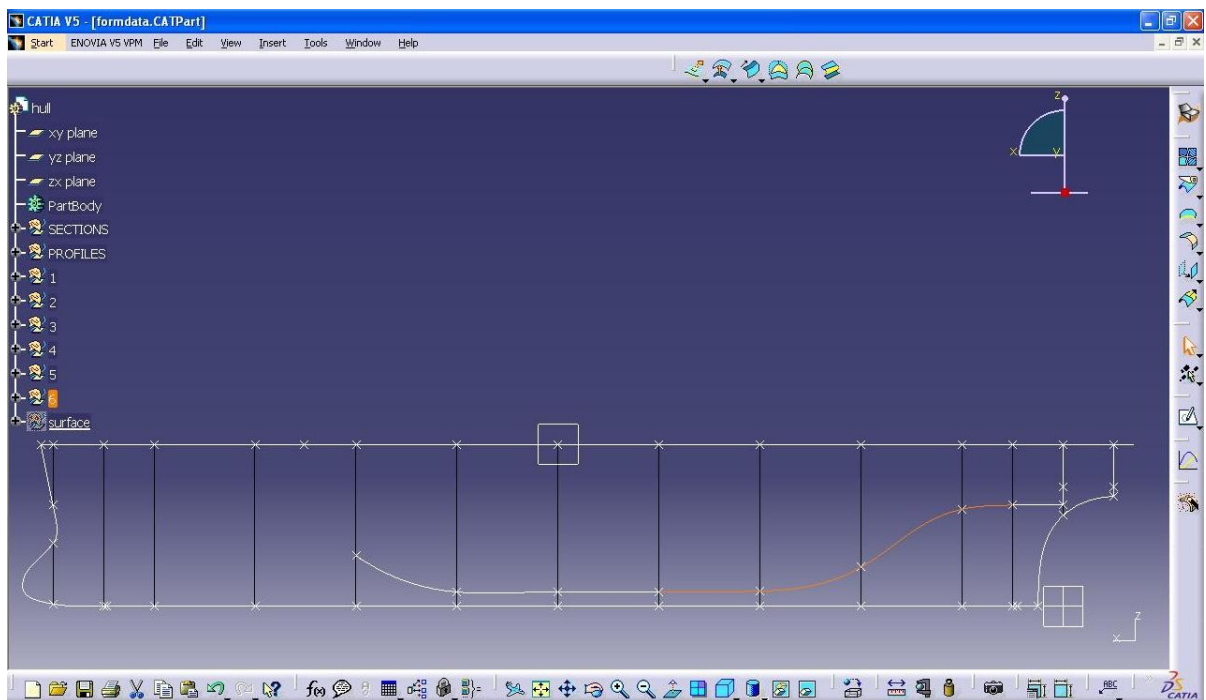
Για την δημιουργία αυτών χρησιμοποιήθηκαν τα **function: AddNewLinePtPt** του **object: Line** και **function: AddNewSpline** του **object: Spline** (Παράρτημα 2).



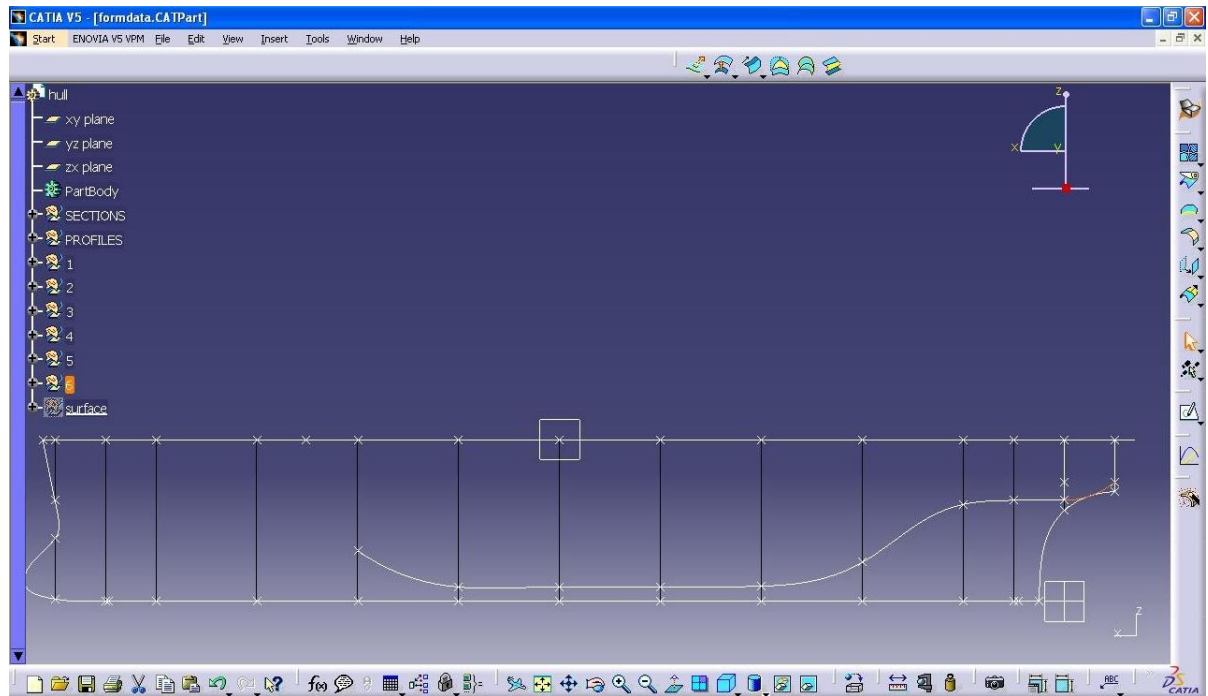
Εικόνα 33f: Κατασκευή ευθειογενών τμημάτων της καμπύλης του Flat of Side (πρύμνη άβακας τύπου D)



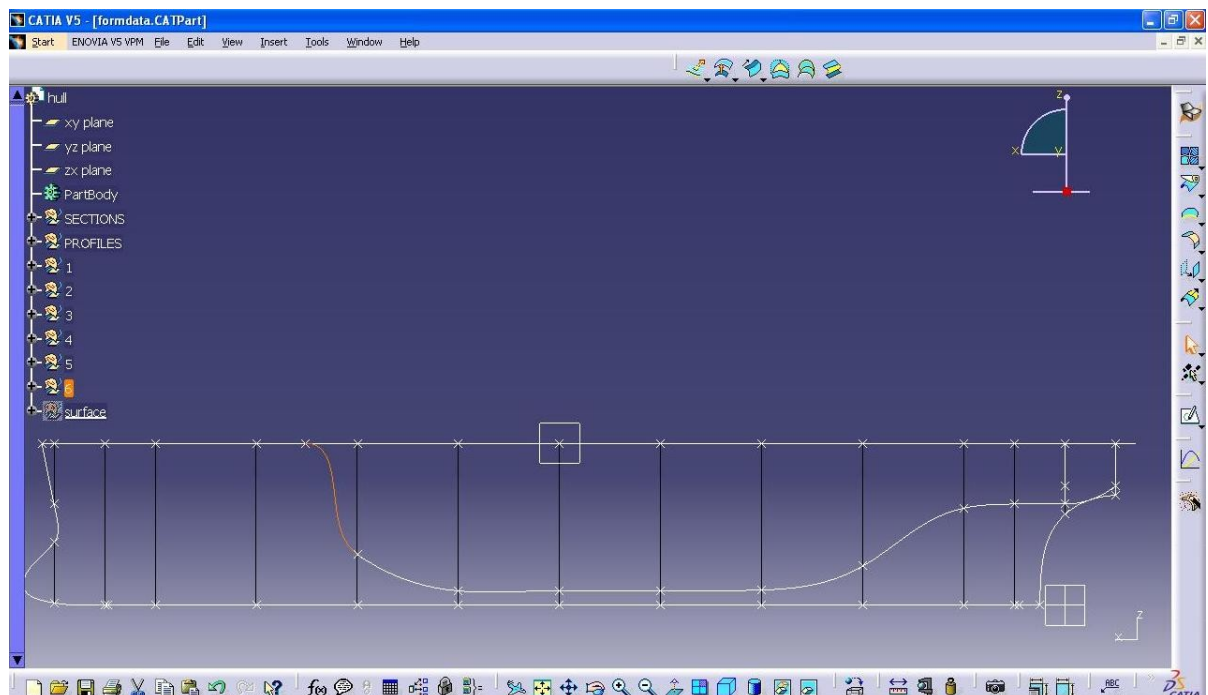
Εικόνα 33g: Κατασκευή προβαίου spline της καμπύλης του Flat of Side (πρόμνη άβακας τύπου D)



Εικόνα 33h: Κατασκευή μεσαιού spline της καμπύλης του Flat of Side (πρόμνη άβακας τύπου D)



Εικόνα 33i: Κατασκευή πρυμναίου spline της καμπύλης του Flat of Side (πρύμνη άβακας τύπου D)



Εικόνα 33j: Κατασκευή ακραίου προωαίου spline της καμπύλης του Flat of Side (πρύμνη άβακας τύπου D)

6.4.11. Δημιουργία των Waterlines του Flat of Side

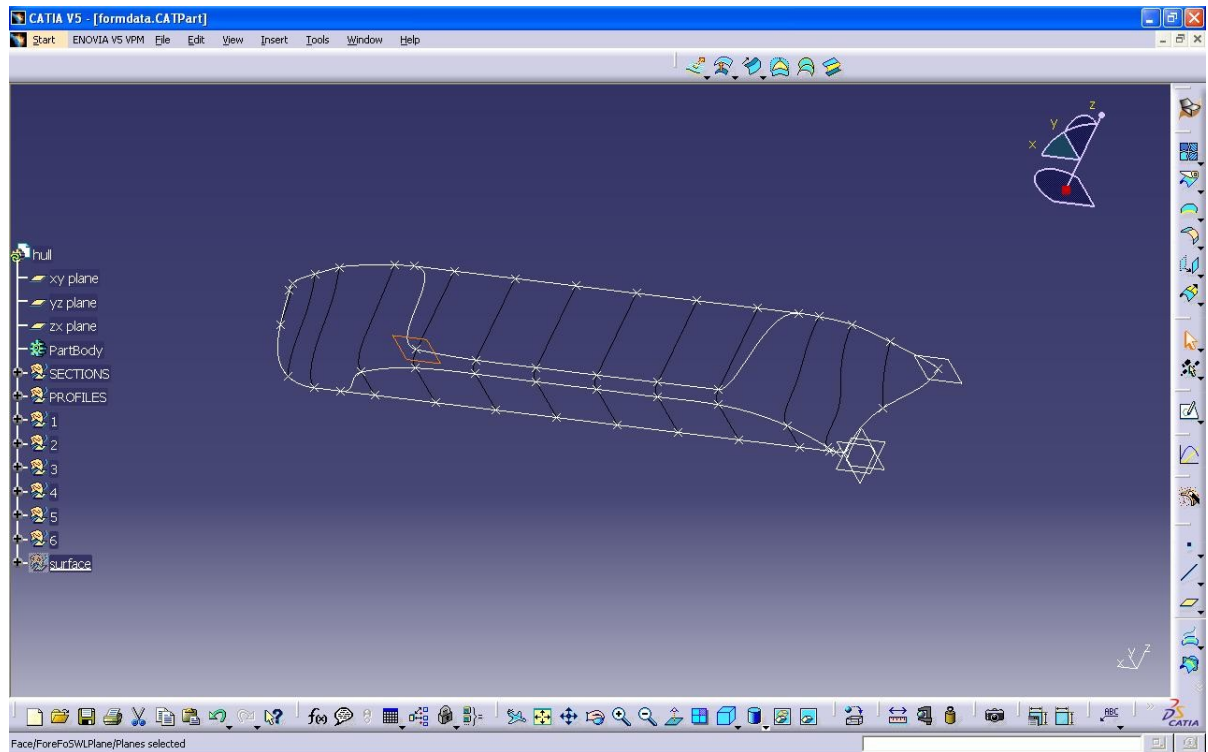
Στη συνέχεια (*block: FoSWaterlines CREATION, στις σειρές: 437-443 του προγράμματος: final.py καλείται το module: FoSWaterlines και τα functions: Fore, Aft*) σχεδιάζονται είτε δύο ίσαλοι (κατηγορίες 1,2,4,5 και υποκατηγορία none) ,είτε μία ίσαλος (κατηγορίες 3 και 6 ,υποκατηγορίες fore, aft) στα άκρα του Flat of Side. Στην περίπτωση όπου και οι δύο ακραίοι νομείς του flat of side καταλήγουν στη deckline (υποκατηγορία both) προφανώς δεν σχεδιάζεται κάποια ίσαλος. Οι συγκεκριμένες ίσαλοι σχεδιάζονται για να διευκολύνουν την κατασκευή της επιφάνειας στη συνέχεια. Συγκεκριμένα στις περιοχές στα άκρα του flat of side, χωρίς την παρουσία των συγκεκριμένων ισάλων, θα έπρεπε να δημιουργηθεί μία επιφάνεια η οποία θα περικλείεται από πέντε καμπύλες (δύο νομείς, την καμπύλη του flat of side, την deckline και είτε την καμπύλη του flat of bottom είτε την Center Line). Δημιουργώντας τις συγκεκριμένες δύο ισάλους χωρίζεται αυτή η περιοχή σε δύο υποπεριοχές οι οποίες περικλείονται από τέσσερις καμπύλες η κάθε μία, πράγμα το οποίο καθιστά ευκολότερη και απλούστερη την κατασκευή επιφάνειας στη συγκεκριμένη περιοχή.

Αρχικά δημιουργείται ένα επίπεδο παράλληλο με το επίπεδο XY (fos fore waterline plane) το οποίο διέρχεται από τον ακραίο (προς την πλώρη) νομέα του Flat of Side (Εικόνα 34a). Στη συνέχεια δημιουργούνται σημεία στις τομές του προηγούμενου επιπέδου με όλους του νομείς πύραθεν του ακραίου νομέα του Flat of Side (fos fore waterline intersections-Εικόνα 34b) μέχρι και το νομέα '9.000'. Τέλος σχεδιάζεται ένα spline (fore FoS waterline) το οποίο ξεκινώντας από το πύραίο ακραίο σημείο του flat of side (με εφαπτομενικό διάνυσμα παράλληλο στον άξονα των X) παρεμβάλει τα σημεία-τομές που δημιουργήθηκαν νωρίτερα (Εικόνα 34c).

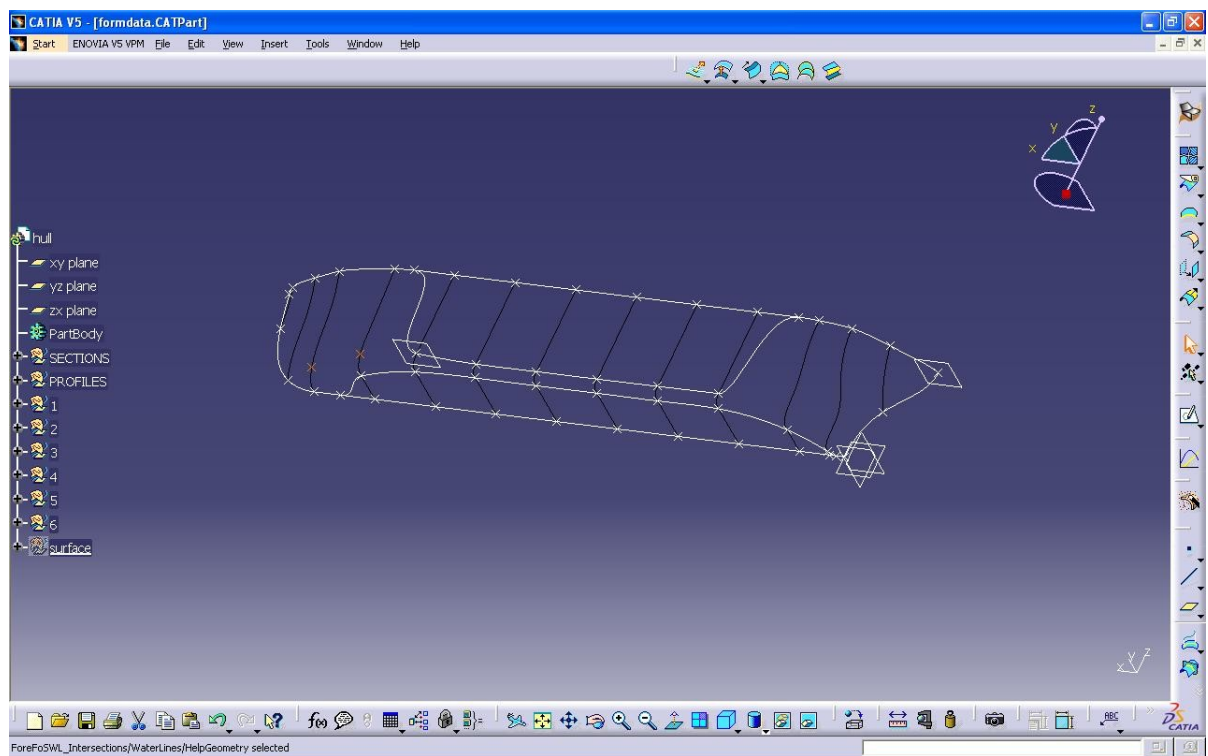
Αντίστοιχα σχεδιάζεται και η πρυμναία waterline στην άκρη του flat of side, εφόσον αυτό είναι εφικτό ,δηλαδή το τρισδιάστατο μοντέλο έχει είτε πρύμνη καταδρομικού είτε άβακα τύπου A,B,C και το πρυμναίο ακραίο σημείο του flat of side δεν βρίσκεται επί του επιπέδου του κυρίου καταστρώματος. Πρώτα δημιουργείται το επίπεδο FoS Aft Waterline Plane (Εικόνα 34d) , στη συνέχεια τα σημεία στις τομές του επιπέδου αυτού με τους νομείς πρύμνηθεν του ακραίου πρυμναίου νομέα του flat of side (Aft FoS Waterline Intersections-Εικόνα 34e) και τέλος το spline που παρεμβάλει τα σημεία αυτά (Aft FoS Waterline-Εικόνα 34f).

Όλα τα αντικείμενα που δημιουργήθηκαν όπως περιγράφηκε παραπάνω, τοποθετήθηκαν στη θέση: **hull⇒category⇒HelpGeometry⇒Waterlines**

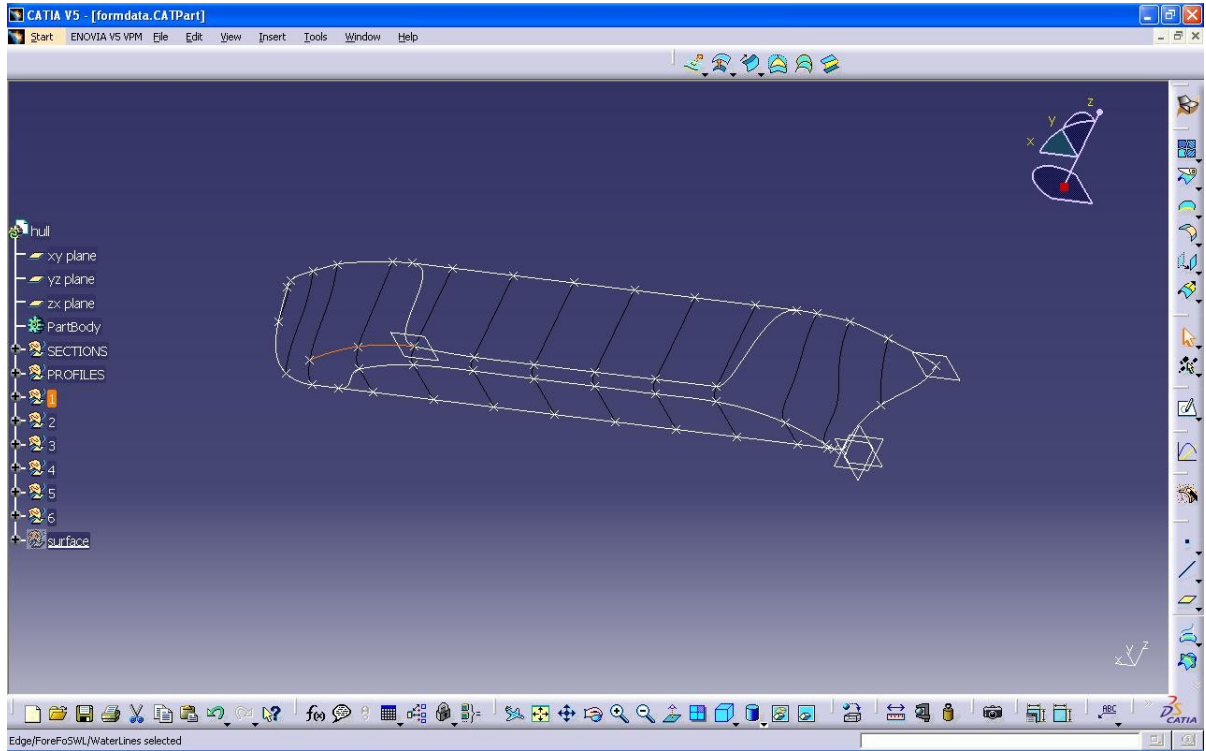
Για την δημιουργία αυτών χρησιμοποιήθηκαν τα **function: AddNewPlaneOffsetPt** του **object: Plane** , **function: AddNewIntersection** του **object: Intersection** και **function: AddNewSpline** του **object: Spline** (Παράρτημα 2).



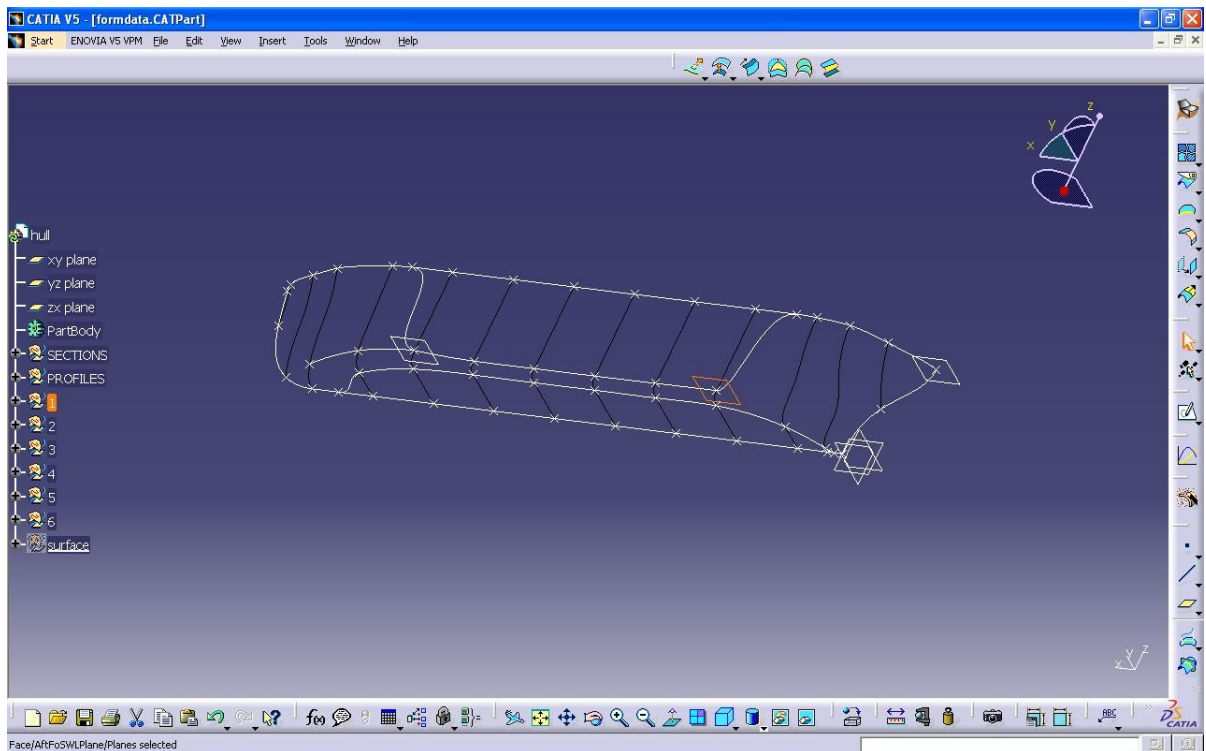
Εικόνα 34a: Δημιουργία επιπέδου fore foos waterline plane



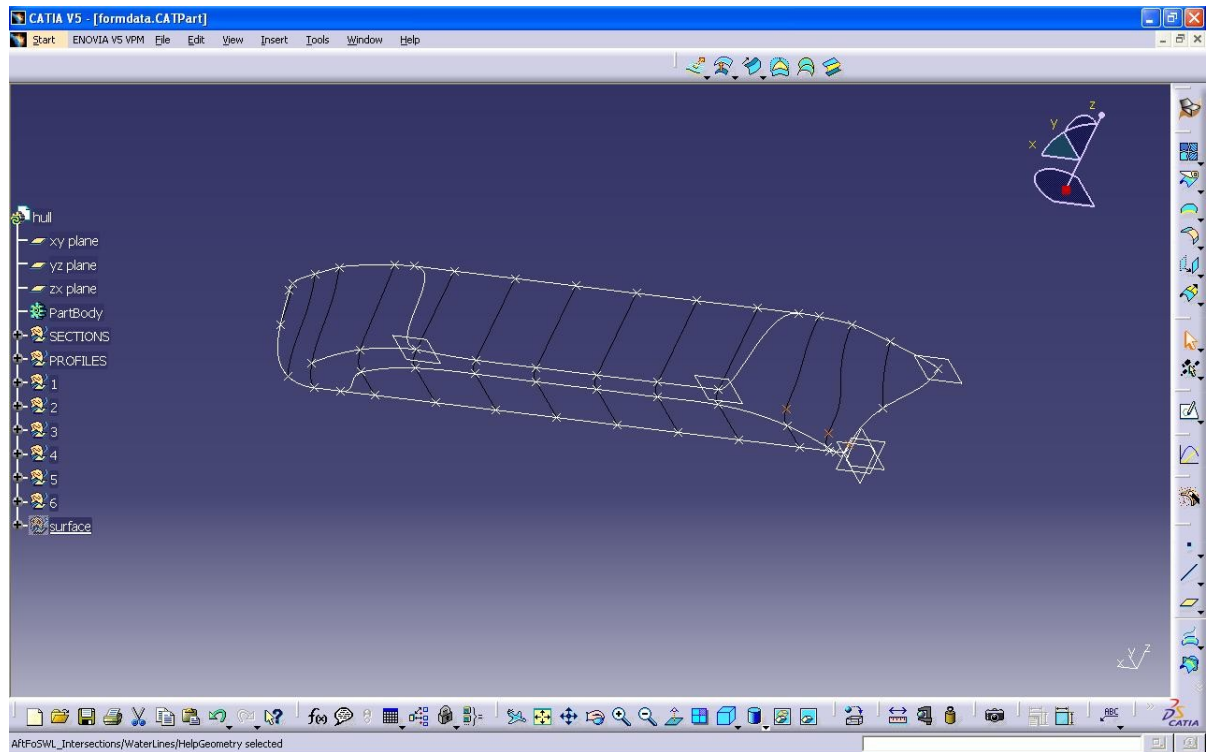
Εικόνα 34b: Δημιουργία τομών του επιπέδου fore foos waterline plane και των προφαιών νομέων



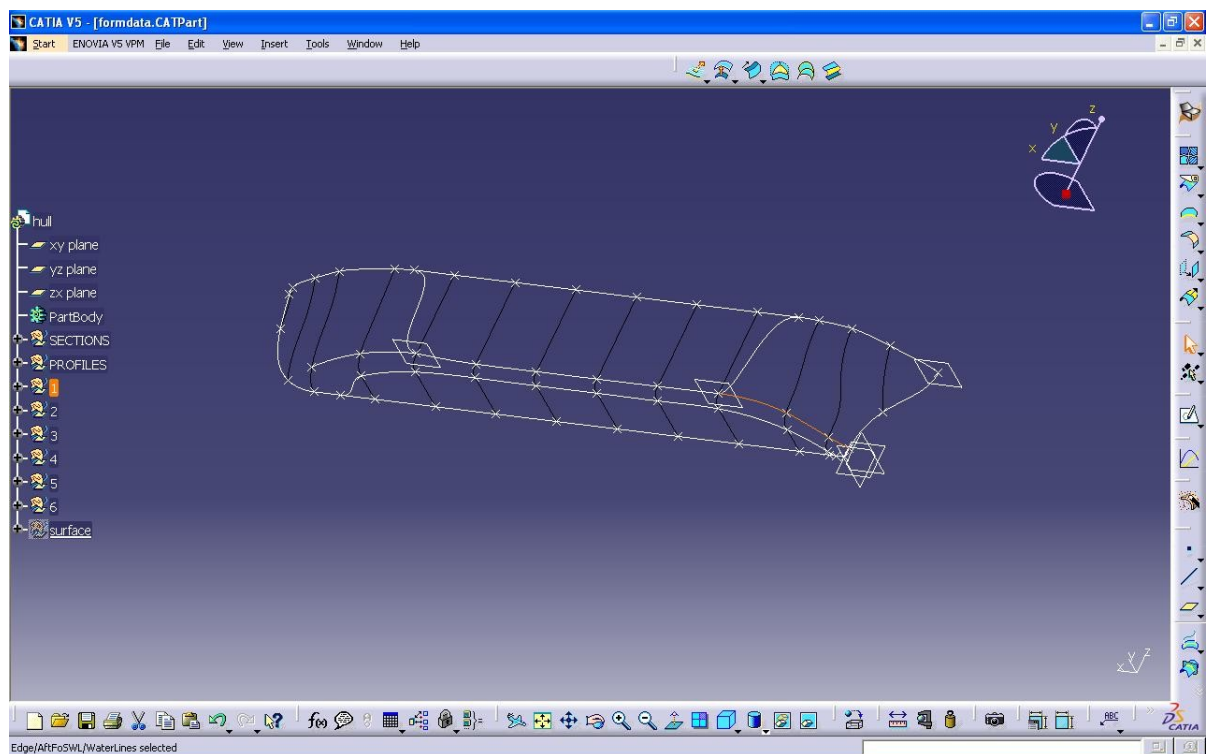
Εικόνα 34c: Δημιουργία προωρίας FoS ισάλου (fore fos waterline)



Εικόνα 34d: Δημιουργία επιπέδου aft fos waterline plane



Εικόνα 34e: Δημιουργία τομών του επιπέδου aft foS waterline plane και των πρυμναίων νομέων



Εικόνα 34f: Δημιουργία πρυμναίας FoS ισάλου (aft foS waterline)

6.4.Διαστατοποίηση αδιάστατου μοντέλου

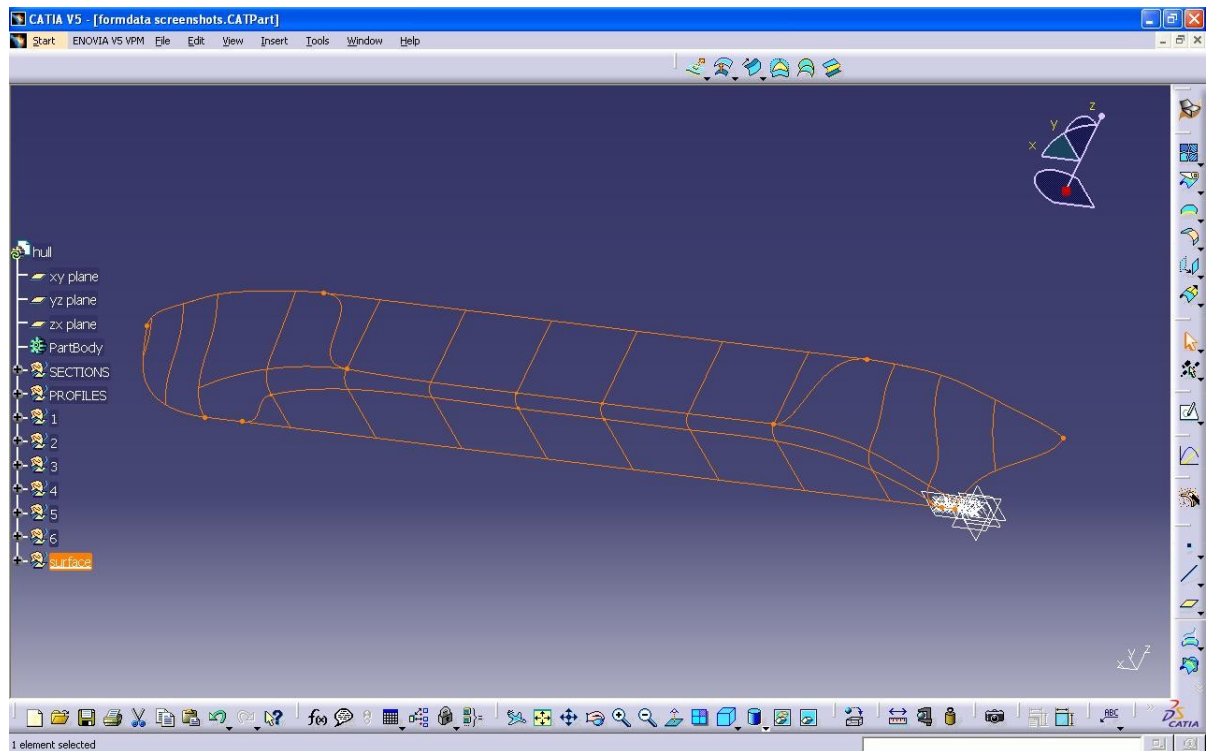
Κατόπιν ακολουθεί η διαστατοποίηση του τρισδιάστατου μοντέλου (*block: SCALE, στις σειρές: 444-445 του προγράμματος: final.py καλείται το module: Affinity και το function:Scale*). Τα αρχικά δεδομένα (νομείς της Formdata) ήταν αδιάστατα. Όλοι οι νομείς είχαν πλάτος 1 και ύψος 1.6 και η ισαπόσταση τους ήταν 1. Συνεπώς το τρισδιάστατο μοντέλο που έχει κατασκευαστεί μέχρι στιγμής και απεικονίζει τη μισή γάστρα πλοίου, έχει διαστάσεις $(L_{BP}, B, D) = (10, 1, 1.6)$. Γνωρίζοντας τις κύριες διαστάσεις του υπό σχεδίαση πλοίου, μας δίνεται η δυνατότητα να διαστατοποιήσουμε το υπάρχον τρισδιάστατο μοντέλο ώστε αυτό να αποκτήσει τελικά τις επιθυμητές διαστάσεις. Εκτός από τους νομείς διαστατοποιήθηκαν και οι εξής βοηθητικές καμπύλες:

- Τα προφίλ πλώρης και πρύμνης.
- Η καμπύλες του Flat of Bottom και Flat of Side
- Η Deck Line
- Η Center Line
- Οι ίσαλοι στα άκρα του Flat of Side (FoS WaterLines)

Οι οποίες δημιουργήθηκαν όπως περιγράφηκε σε προηγούμενες παραγράφους, για να διευκολύνουν την κατασκευή της επιφάνειας της γάστρας στη συνέχεια. Στην Εικόνα 35 φαίνεται το αρχικό τρισδιάστατο μοντέλο καθώς και το διαστατοποιημένο.

Οι διαστατοποιημένες καμπύλες που δημιουργήθηκαν όπως περιγράφηκε παραπάνω, τοποθετήθηκαν στη θέση: **hull**⇒**surface**⇒**input**

Για την διαστατοποίηση του τρισδιάστατου μοντέλου χρησιμοποιήθηκε το **function:AddNewAffinity** του **object: HybridShapeAffinity** (Παράρτημα 2).



Εικόνα 35: Διαστατοποίηση τρισδιάστατου μοντέλου

6.5.Κατασκευή επιφανειών

Η κατασκευή της επιφάνειας της γάστρας έχει χωριστεί σε τρία μέρη (*block: SURFACE DESIGN, στις σειρές: 455-469 του προγράμματος: final.py* καλούνται τα *module:surface* και το *function:creation, module:aft* και τα *functions:cruiser,transom* και *transomABC ,module:bulb* και το *function:surface*). Αρχικά κατασκευάζεται η επιφάνεια ανάμεσα στους νομείς 0.500 και 9.000, όταν το μοντέλο έχει πλώρη χωρίς βολβό ή 0.500 και 9.500 όταν το μοντέλο διαθέτει βολβοειδή πλώρα (ενδιάμεσο τμήμα). Στη συνέχεια ακολουθεί η κατασκευή της πρύμνης , και τέλος η κατασκευή της πλώρης.

6.5.1.Κατασκευή ενδιάμεσου τμήματος της επιφάνειας της γάστρας

Η κατασκευή της επιφάνειας ανάμεσα στους νομείς 9.000 (ή 9.500) και 0.500 απαρτίζεται από πέντε επιμέρους κομμάτια:

- Την επιφάνεια του Flat of Side
- Την επιφάνεια του Flat of Bottom
- Την επιφάνεια ανάμεσα σε αυτές του Flat of Side και του Flat of Bottom
- Το προωαίο κομμάτι
- Το πρυμναίο κομμάτι

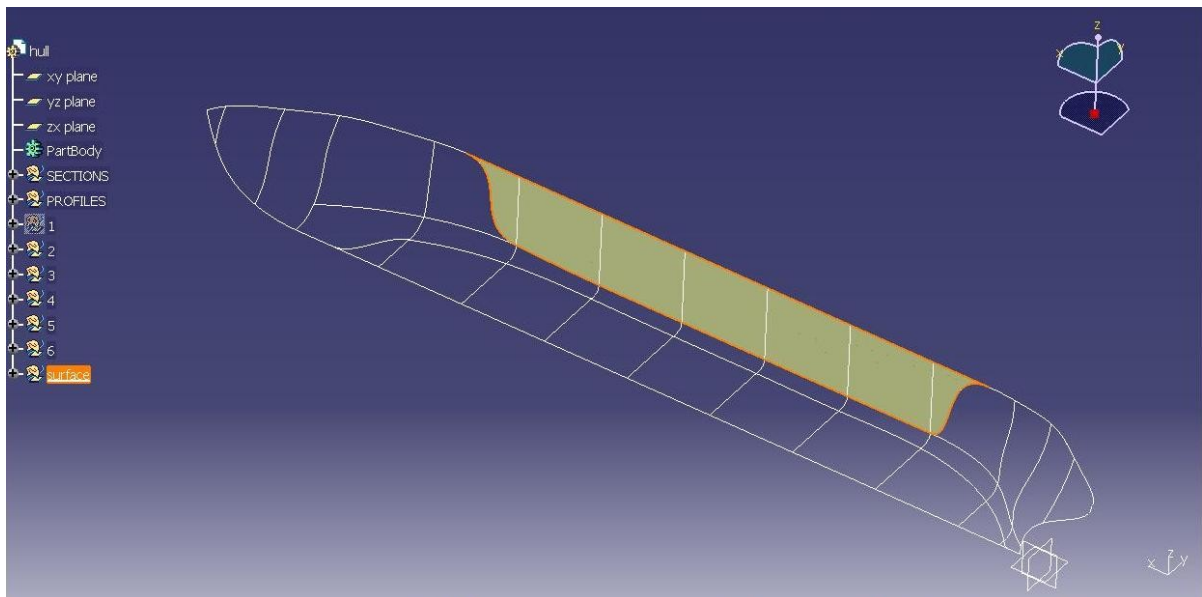
Αρχικά δημιουργείται μία σειρά από λίστες (*block: SECTIONS CATEGORIZATION στις γραμμές: 9-73 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*), με τις οποίες κατηγοριοποιούνται οι νομείς ανάλογα με την θέση τους.

Όνομα λίστας	Περιγραφή
Fbsandfss	Περιέχει τους νομείς που έχουν και Flat of Side και Flat of Bottom
Fbsnotfss	Περιέχει τους νομείς που έχουν Flat of Bottom και δεν έχουν Flat of Side
Forefbsnotfss	Περιέχει τους προωαίους νομείς που έχουν Flat of Bottom και δεν έχουν Flat of Side
Aftfbsnotfss	Περιέχει τους πρυμναίους νομείς που έχουν Flat of Bottom και δεν έχουν Flat of Side
Notfbsnotfss	Περιέχει τους νομείς που δεν έχουν ούτε Flat of Side ούτε Flat of Bottom
Forenotfbsnotfss	Περιέχει τους προωαίους νομείς που δεν έχουν ούτε Flat of Side ούτε Flat of Bottom
Aftnotfbsnotfss	Περιέχει τους πρυμναίους νομείς που δεν έχουν ούτε Flat of Side ούτε Flat of
Aftfssnotfbs	Περιέχει τους πρυμναίους νομείς που έχουν Flat of Side και δεν έχουν Flat of Bottom
Aftnotfss	Περιέχει τους πρυμναίους νομείς που δεν έχουν Flat of Side
Forenotfss	Περιέχει τους προωαίους νομείς που δεν έχουν Flat of Side
Forenotfbs	Περιέχει τους προωαίους νομείς που δεν έχουν Flat of Bottom
Aftnotfbs	Περιέχει τους πρυμναίους νομείς που δεν έχουν Flat of Bottom

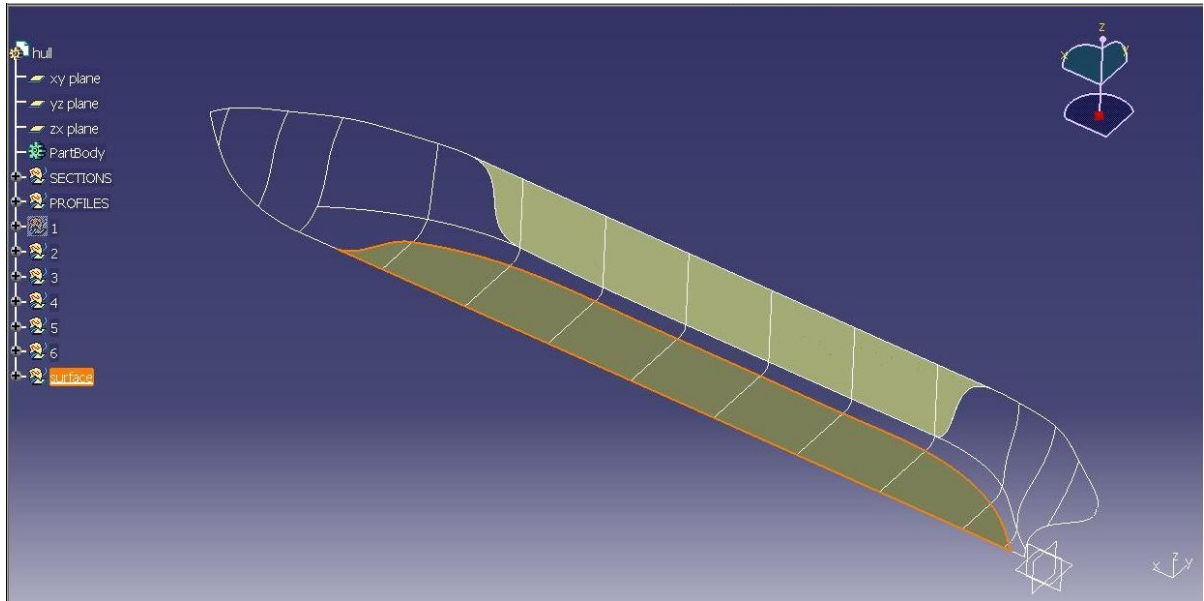
Στη συνέχεια είναι απαραίτητο να ενωθούν (εντολή Join-Παράρτημα 2) ορισμένες καμπύλες που απαρτίζονται από περισσότερα κομμάτια. Έτσι ακολουθεί ή ένωση της CenterLine (*block: JOIN CENTER LINE στις γραμμές: 75-90 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*) που αποτελείται από τρία κομμάτια (CL-Fore,CL,CL-Aft), της DeckLine (*block: JOIN DECKLINE στις γραμμές: 91-106 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*) η οποία αποτελείται επίσης από τρία κομμάτια (DeckLineFore, DeckLineParrallel, DeckLineAft) και τέλος της καμπύλης του Flat of Side (*block:JOIN FOS PARTS στις γραμμές: 106-153 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*) η οποία αποτελείται από τρία κομμάτια συνήθως, ανάλογα με την FoS κατηγορία στην οποία ανήκει το υπό σχεδίαση μοντέλο.

Πρώτα κατασκευάζεται η επιφάνεια του Flat of Side (Εικόνα 36), χρησιμοποιώντας την εντολή Fill (Παράρτημα 2) με όρια τις καμπύλες DeckLineParrallel , και FoS *block:Flat of Side CREATION στις γραμμές: 154-175 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py* . Στην περίπτωση που το τρισδιάστατο μοντέλο διαθέτει πρύμνη άβακα τύπου D ,προστίθεται και ένα τρίτο οριο, η καμπύλη TransomVerticalPart. Επειτα , κατασκευάζεται με αντίστοιχο τρόπο η επιφάνεια του Flat of Bottom(Εικόνα 37), με όρια τις καμπύλες FoB και CL *block: Flat of Bottom CREATION στις γραμμές: 176-193 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*).

Και οι δύο αυτές επιφάνειες τοποθετήθηκαν στη θέση: **hull⇒surface**.

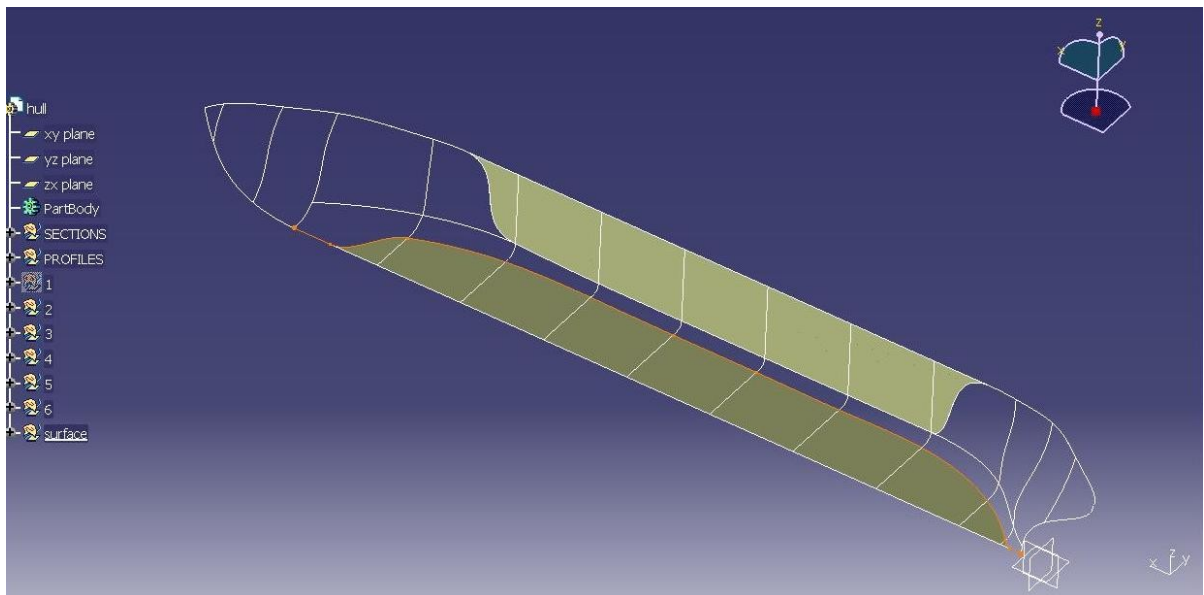


Εικόνα 36: Κατασκευή επιφάνειας του Flat of Side

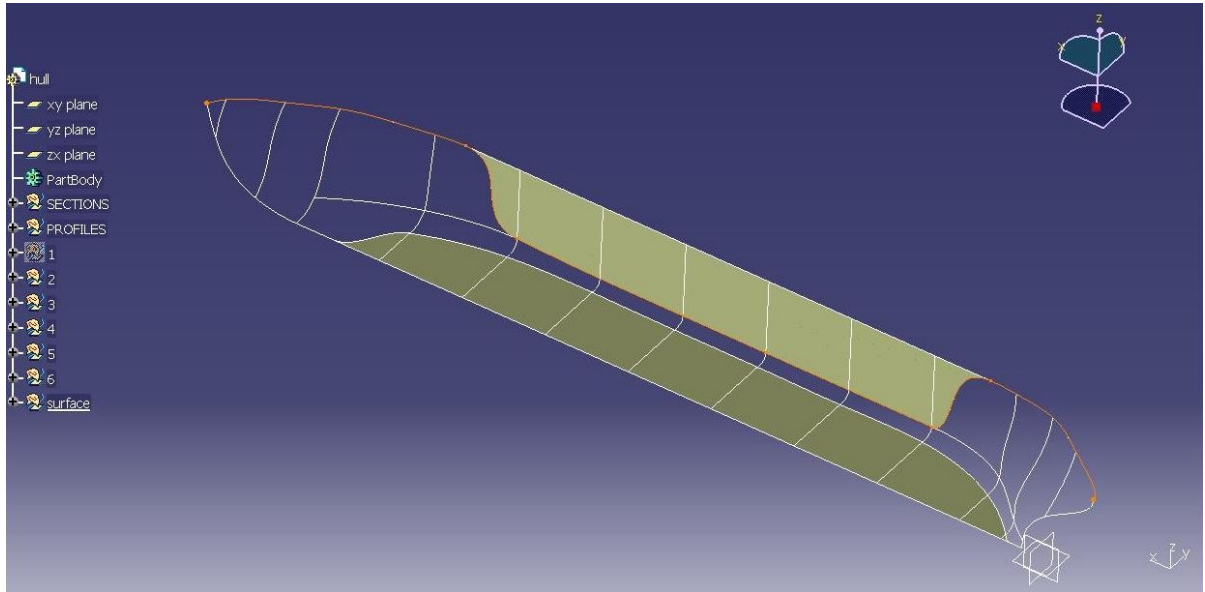


Εικόνα 37: Κατασκευή επιφάνειας του Flat of Bottom

Στη συνέχεια ακολουθεί η δημιουργία δυο νέων καμπύλων μέσω της ένωσης ήδη υπάρχοντων καμπύλων. Πρώτα δημιουργείται η καμπύλη **GeneralCL** (Εικόνα 38), η οποία αποτελείται από τις **CL-Fore**, **FoB** και **CL-Aft** (*block:GENERALCL CREATION στις γραμμές: 194-212 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*) και εν συνεχεία η καμπύλη **GeneralDeckLine** (Εικόνα 39) η οποία αποτελείται από τις **DeckLineFore**, **FoS** και **DeckLineAft** (*block: GENERAL DECKLINE CREATION στις γραμμές: 213-228 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*).

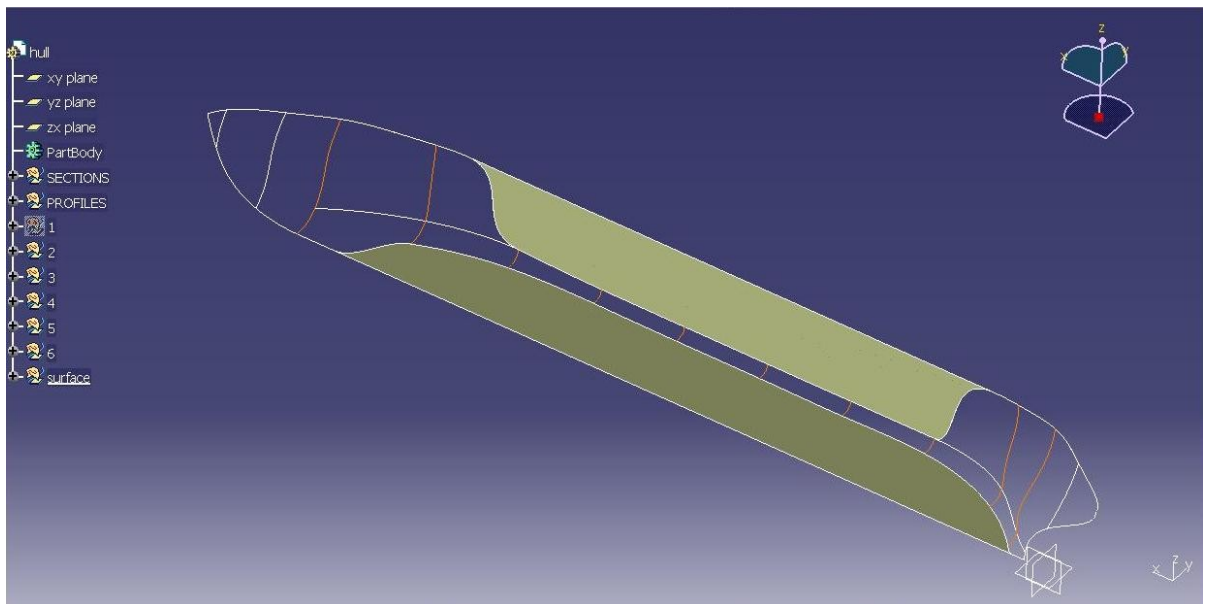


Εικόνα 38: Δημιουργία καμπύλης **GeneralCL**



Εικόνα 39: Δημιουργία καμπύλης **GeneralDeckLine**

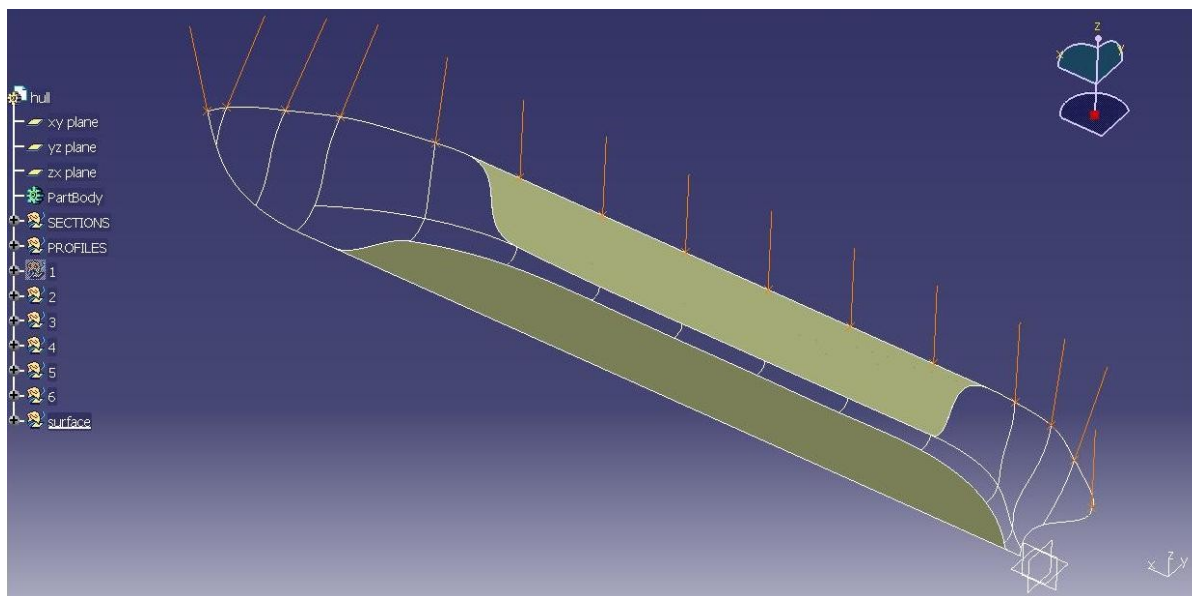
Έπειτα ακολουθεί η διαίρεση (εντολή Split-Παράρτημα 2) των νομέων (Εικόνα 40) με στοιχεία τομής τις δύο καμπύλες που δημιουργήθηκαν νωρίτερα (**GeneralCL** και **GeneralDeckLine**). Η διαίρεση των νομέων (*block: SPLIT SECTIONS στις γραμμές: 229-251 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*) είναι απαραίτητη για την κατασκευή των επόμενων επιφανειών όπου θα γίνει χρήση της εντολής Loft. Η συγκεκριμένη εντολή απαιτεί οι τομές που χρησιμοποιεί να αρχίζουν και να τελειώνουν πάνω στις καμπύλες οδηγούς, και όχι απλά να έχουν από ένα σημείο τομής.



Εικόνα 40: Διαίρεση των νομέων

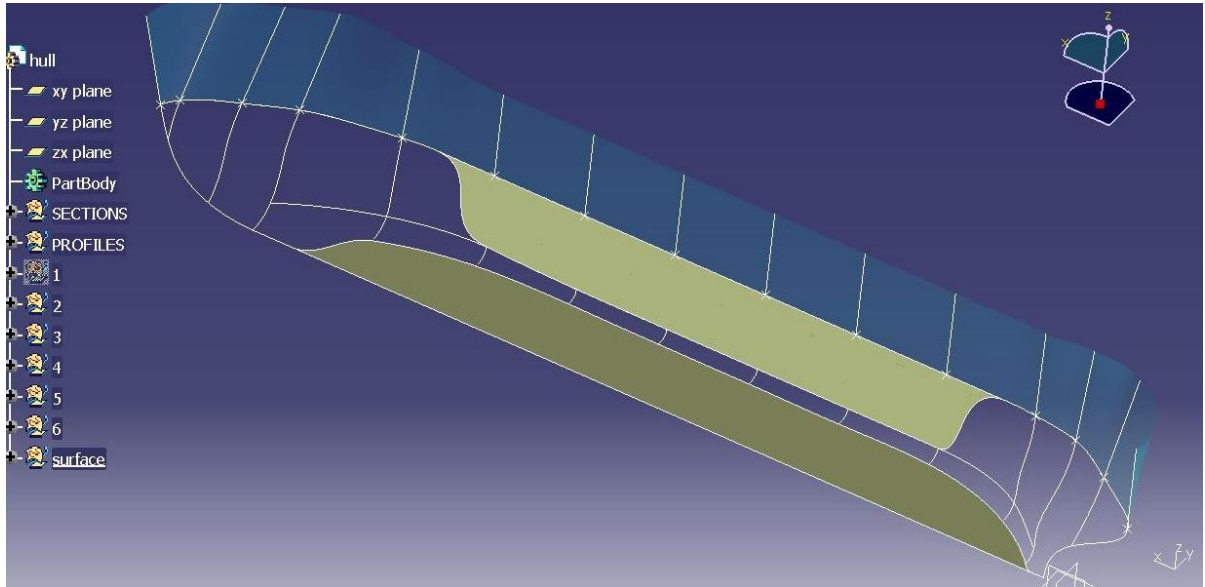
Στη συνέχεια περιγράφεται η κατασκευή δύο βοηθητικών επιφανειών (*block: SUPPORT SURFACES στις γραμμές: 252-492 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*), οι οποίες θα παίξουν τον ρόλο των επιφανειών στήριξης (support surfaces) για τις επιφάνειες που θα σχεδιαστούν στη συνέχεια και θα απαρτίζουν την επιφάνεια της γάστρας του μοντέλου.

Η πρώτη επιφάνεια, θα κατασκευαστεί άνωθεν του περιγράμματος του κυρίου καταστρώματος. Αρχικά σχεδιάζονται ξανά τα ανώτατα ακραία σημεία των νομέων και των προφίλ (όπως περιγράφηκε νωρίτερα για το αδιάστατο μοντέλο) και κατόπιν σχεδιάζεται, στην κορυφή κάθε νομέα και προφίλ, με αρχή το ανώτερο ακραίο σημείο και φορά προς τα πάνω, μία ευθεία μήκους 20m η οποία εφάπτεται της εκάστοτε καμπύλης στο συγκεκριμένο σημείο (Εικόνα 41).



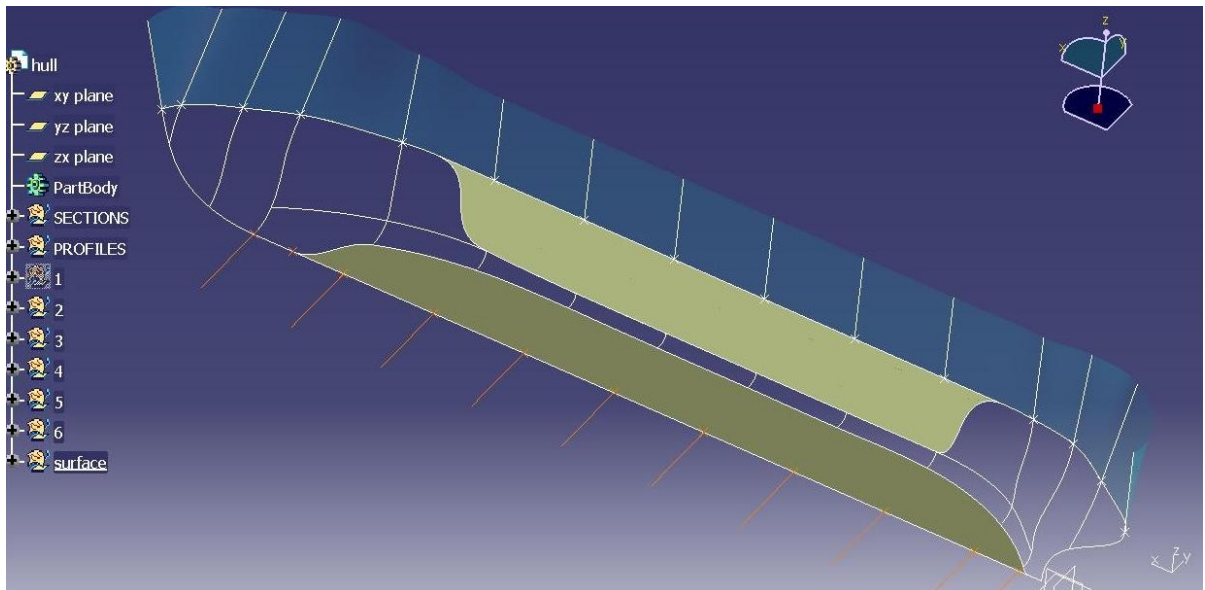
Εικόνα 41: Δημιουργία κορυφαίων ακραίων σημείων και ευθειών των νομέων και των προφίλ

Στη συνέχεια χρησιμοποιώντας την εντολή AddNewLoft (Παράρτημα 2) και πιο συγκεκριμένα χρησιμοποιώντας τις ευθείες, που σχεδιάστηκαν ακριβώς πριν, ως τομές και την καμπύλη του περιγράμματος του κυρίου καταστρώματος ως οδηγό, κατασκευάζεται μία επιφάνεια (DeckLine Support Surface-Εικόνα 42). Η συγκεκριμένη επιφάνεια, καθώς και η επόμενη, κατασκευάστηκαν για να βοηθήσουν την κατασκευή των επιφανειών που θα απαρτίζουν την επιφάνεια της γάστρας. Τα εργαλεία που μας παρέχει το CATIA έχουν ορισμένες ιδιαιτερότητες, συγκεκριμένα τα δύο εργαλεία που χρησιμοποιήθηκαν ως εκ το πλείστον για την κατασκευή των επιφανειών (Fill και Loft) παράγουν καλύτερο αποτέλεσμα εάν δίδονται συνθήκες εφάπτομενικότητας στα όρια των υπό κατασκευή επιφανειών. Εύκολα γίνεται αντιληπτό ότι οι δύο επιφάνειες δημιουργούνται για να καθορίσουν τις εφάπτομενες στα άκρα-όρια των επιφανειών που θα δημιουργηθούν στη συνέχεια.



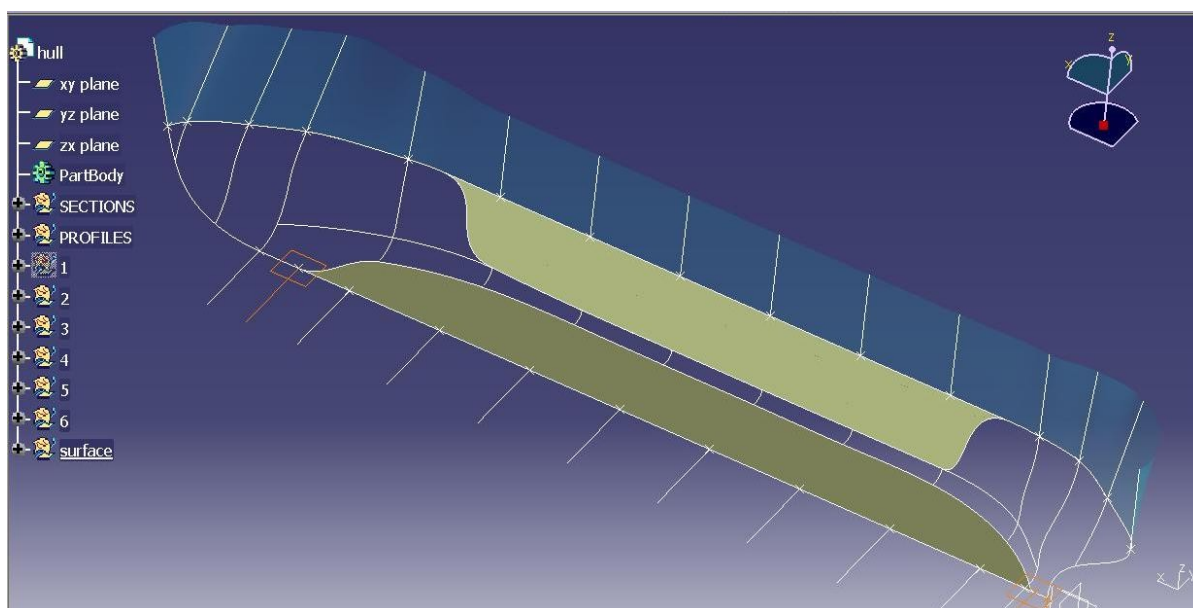
Εικόνα 42: Κατασκευή της DeckLine Support Surface

Με αντίστοιχο τρόπο δημιουργούνται στη συνέχεια τα κατώτατα ακραία σημεία των νομέων ,και συγκεκριμένα των νομέων 9.000 έως 0.500, και του πρυμναίου προφίλ μόνο καθώς και οι αντίστοιχες εφαπτόμενες ευθείες με φορά προς τα κάτω αυτή τη φορά (Εικόνα 43).

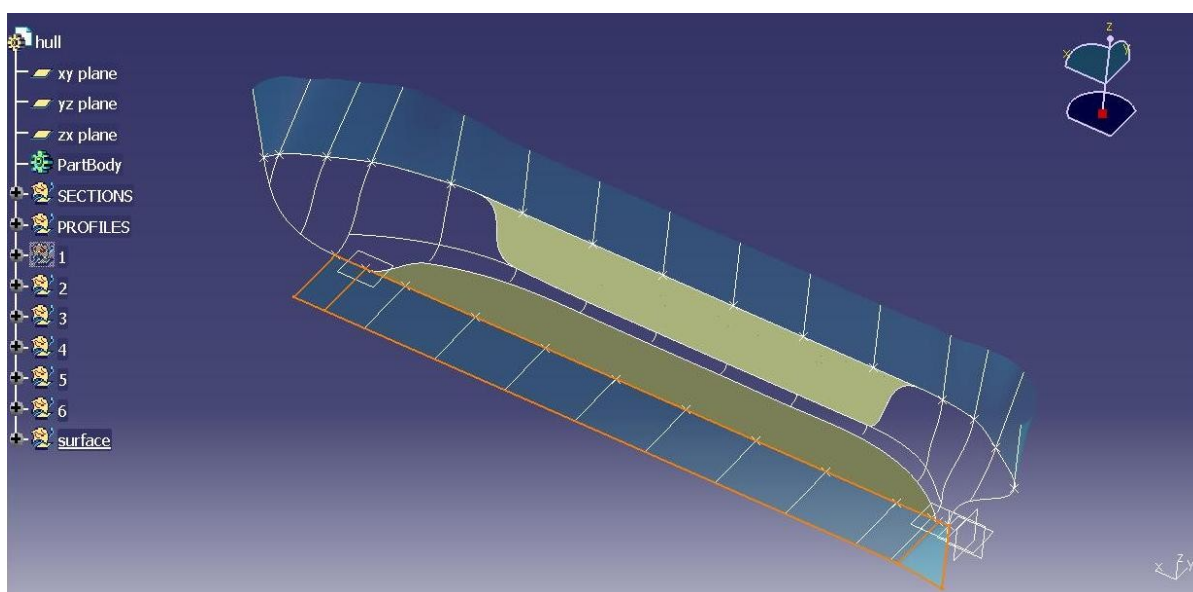


Εικόνα 43: Δημιουργία κατώτατων ακραίων σημείων και ευθειών των νομέων και του προφίλ της πρύμνης

Στην περίπτωση αυτή κρίθηκε απαραίτητο να δημιουργηθούν άλλες δύο ευθείες (Εικόνα 44) παράλληλες στο επίπεδο του Flat of Bottom στα σημεία τομής της Center Line με την καμπύλη του Flat of Bottom. Η επιφάνεια που θα δημιουργηθεί στη συνέχεια πρέπει να είναι εφικτό να συνενωθεί με αυτή του Flat of Bottom, συνεπώς πρέπει να εξασφαλίσουμε ότι η νέα επιφάνεια θα έχει την ίδια κλίση με το Flat of Bottom σε όλο το μήκος του κομματιού της Center Line που βρίσκεται στο άκρο της επιφάνειας του επιπέδου πυθμένα. Στις συγκεκριμένες περιοχές (άκρα του επιπέδου πυθμένα) αλλάζει απότομα κλίση η εφαπτόμενη στο κάτω άκρο των νομέων και χωρίς τις ευθείες που προαναφέρθηκαν δεν είναι μπορεί να είναι σίγουρο ότι η επιφάνεια που θα δημιουργηθεί θα έχει την ίδια κλίση με τον επίπεδο πυθμένα στο κομμάτι που συμπίπτουν. Έπειτα χρησιμοποιώντας όλες τις ευθείες, στη σωστή σειρά, ως τομές και την Center Line ως καμπύλη οδηγό, με τη χρήση της εντολής AddNewLoft δημιουργείται η επιφάνεια CL Support Surface (Εικόνα 45).

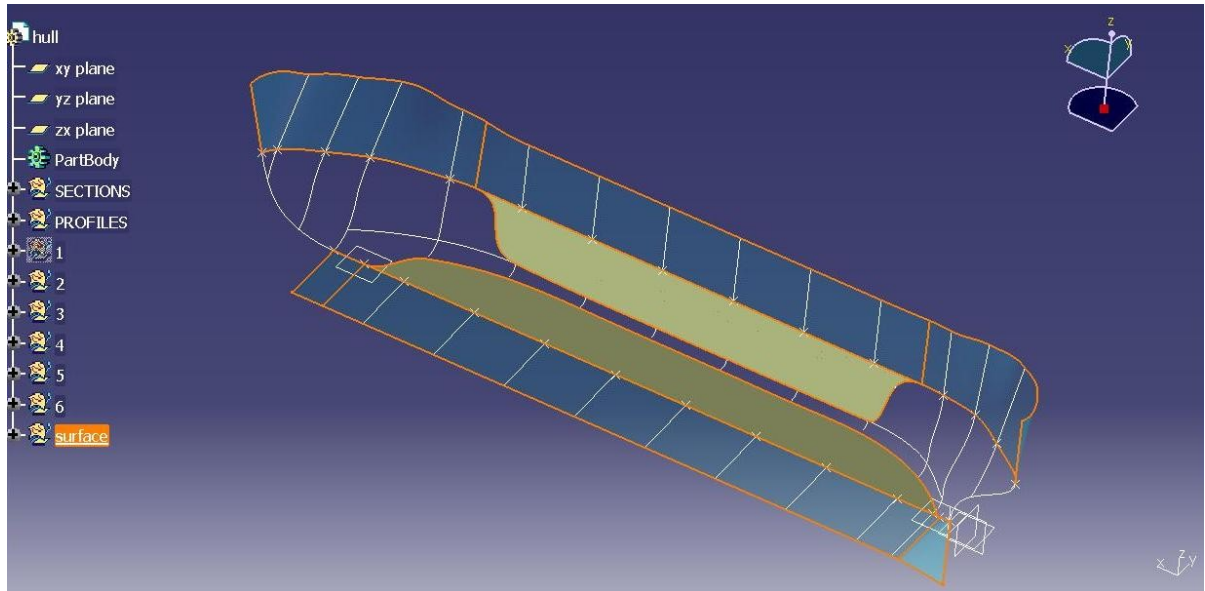


Εικόνα 44: Δημιουργία ευθειών στα άκρα του επιπέδου πυθμένα



Εικόνα 45: Δημιουργία επιφάνειας CL Support Surface

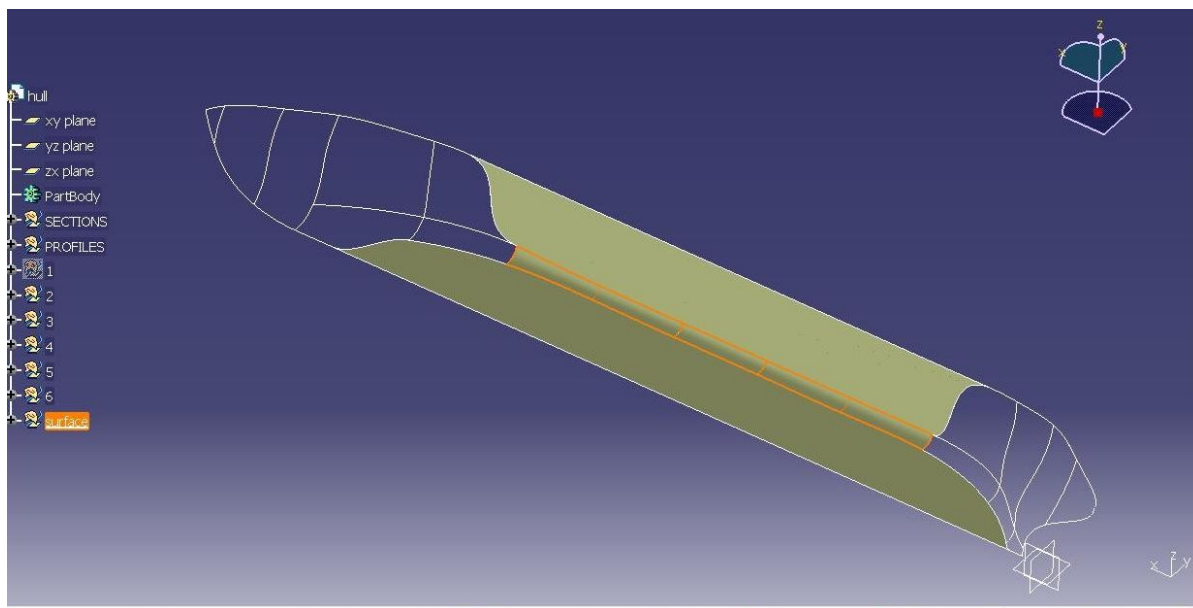
Τέλος ενώνονται (Εικόνα 46) η επιφάνεια του Flat of Side και η DeckLine Support Surface σε μία (General deckline support surface) και αντίστοιχα η CL Support Surface και η επιφάνεια του Flat of Bottom σε άλλη μία (General CL support surface).



Εικόνα 46: Δημιουργία επιφανειών General DeckLine support surface και General CL support surface

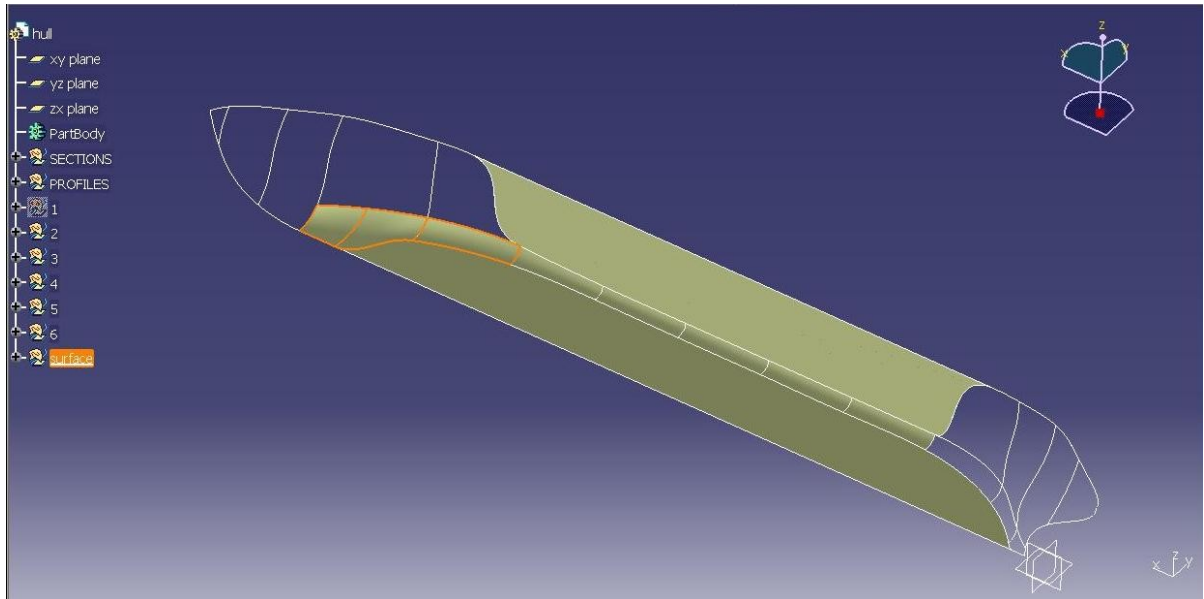
Όλα τα γεωμετρικά στοιχεία που δημιουργήθηκαν για την κατασκευή των παραπάνω επιφανειών στήριξης, καθώς και οι ίδιες οι επιφάνειες, τοποθετήθηκαν στη θέση **hull**⇒**surface**⇒**support**, και κατόπιν «κρύφτηκαν» αφού ο ρόλος τους θα είναι βοηθητικός μόνο.

Η κατασκευή της επιφάνειας της γάστρας, του υπό σχεδίαση πλοίου συνεχίζεται με την κατασκευή της επιφάνειας ανάμεσα στο flat of side και το flat of bottom (*block: SURFACE BETWEEN FLAT OF SIDE AND FLAT OF BOTTOM στις γραμμές: 493-520 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*). Όπως αναφέρθηκε προηγουμένως έχει δημιουργηθεί μία λίστα (fbsandfss) η οποία περιέχει τους νομείς οι οποίοι διαθέτουν και flat of side και flat of bottom. Χρησιμοποιώντας τους νομείς που περιέχει η λίστα fbsandfss ως τομές και τις καμπύλες του flat of side και flat of bottom ως καμπύλες οδηγούς, με τη βοήθεια της εντολής Loft κατασκευάζεται η εν λόγω επιφάνεια (Εικόνα 47). Αξίζει να σημειωθεί ότι οι επιφάνειες του flat of side και flat of bottom χρησιμοποιήθηκαν ως επιφάνειες στήριξης. Η επιφάνεια αυτή ονοματίστηκε με βάση τους ακραίους νομείς τους οποίους περιελάμβανε, στο παράδειγμα που φαίνεται στις εικόνες η εν λόγω επιφάνεια ονομάστηκε : “surface between sections 7.000 and 2.000”.



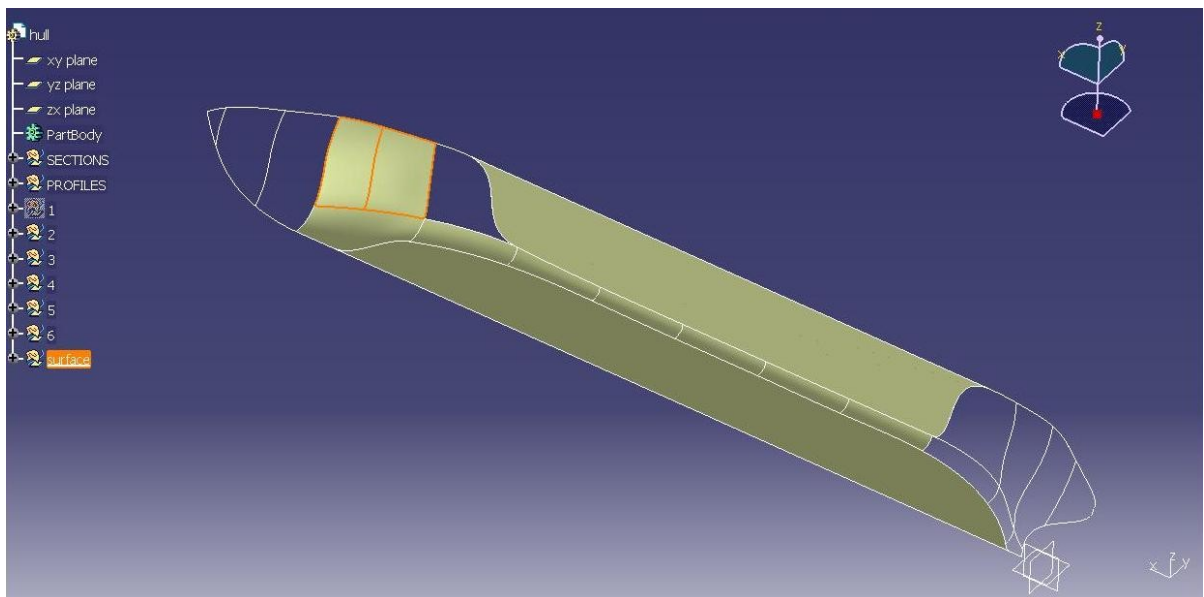
Εικόνα 47: Κατασκευή επιφάνειας ανάμεσα στο Flat of Side και το Flat of Bottom

Για την κατασκευή της επιφάνειας πρόραθεν αυτής ανάμεσα στο Flat of Side και το Flat of Bottom και κάτω από την προραία FoS ίσαλο είναι απαραίτητο να διχοτομηθούν οι νομείς (*block: Split sections forward of fss στις γραμμές: 521-576 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*) που βρίσκονται μπροστά από τον τελευταίο νομέα του flat of side. Έτσι χρησιμοποιώντας την προραία FoS ίσαλο ως καμπύλη τομής διχοτομούνται όλοι οι νομείς που βρίσκονται στις λίστες forefbsnotfss και forenotfbsnotfss. Στη συνέχεια χρησιμοποιώντας την προραία FoS ίσαλο και την General CL ως καμπύλες οδηγούς και τα κάτω μέρη των διχοτομημένων πλέον νομέων από τις λίστες forefbsnotfss και forenotfbsnotfss καθώς και τον διχοτομημένο ακραίο προραίο νομέα που ανήκει στο flat of side, ως τομές κατασκευάζεται μία επιφάνεια (*block: SURFACE UNDER FOREFOSWL στις γραμμές: 577-617 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py*-Εικόνα 48) με τη βοήθεια της εντολής Loft (Παράρτημα 2). Από τη μεριά της General CL χρησιμοποιήθηκε ως επιφάνεια υποστήριξης η General CL support Surface, και από τη μεριά του ακραίου προραίου νομέα του flat of side (στο συγκεκριμένο παράδειγμα ο νομέας 7.000) η επιφάνεια ανάμεσα στο flat of side και το flat of bottom. Με αντίστοιχη λογική με αυτή που περιγράφηκε για την προηγούμενη επιφάνεια, ονοματίστηκε και αυτή: “Lower surface between sections 9.000-7.000”.



Εικόνα 48: Δημιουργία της επιφάνειας κάτω από την προραία FoS ισάλου

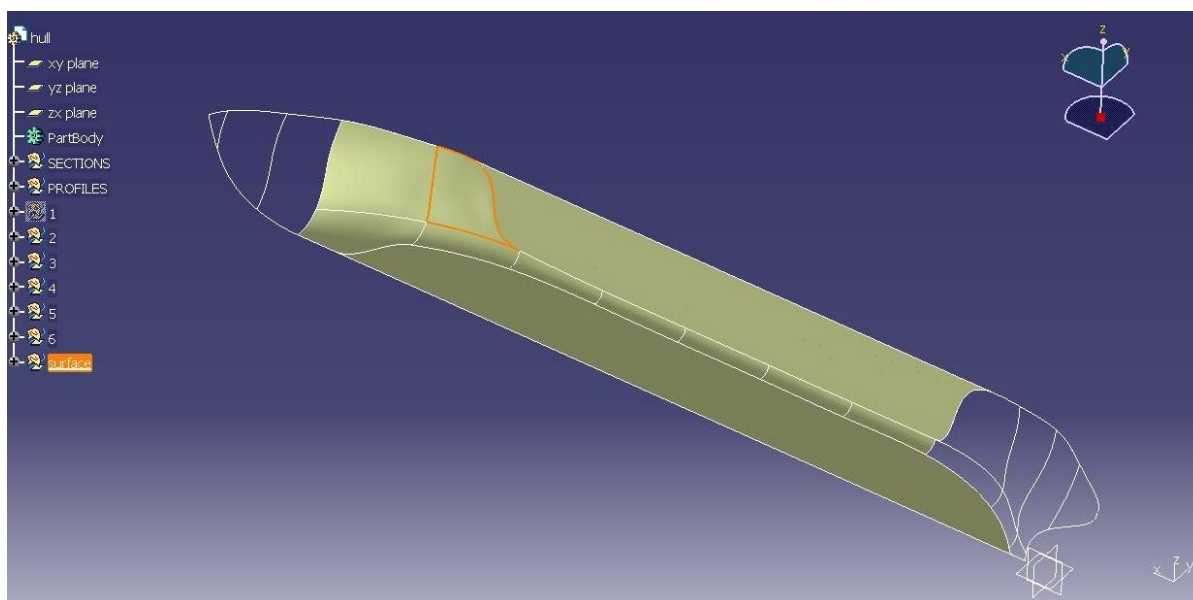
Στη συνέχεια με αντίστοιχο τρόπο θα σχεδιαστεί η επιφάνεια άνωθεν της προραίας FoS ισάλου (*block: SURFACE OVER FOREFOSWL στις γραμμές:618-649 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py-Εικόνα 49*). Αυτή τη φορά ως τομές χρησιμοποιήθηκαν τα επάνω μέρη των διχοτομημένων νομέων και αντί για την καμπύλη General CL με επιφάνεια υποστήριξης την General CL support surface χρησιμοποιήθηκε η καμπύλη General DeckLine με επιφάνεια υποστήριξης την General DeckLine support surface. Κατά αντιστοιχία με τις προηγούμενες επιφάνειες η συγκεκριμένη ονομάστηκε: “Upper surface between sections 9.000-7.000”.



Εικόνα 49: Δημιουργία της επιφάνειας πάνω από την προραία FoS ισάλου

Παρατηρείται ότι ανάμεσα στο flat of side και τον αμέσως επόμενο νομέα προς την πλώρη έχει δημιουργηθεί ένα κενό. Λόγω της έντονης μεταβολής της γεωμετρίας της γάστρας στη συγκεκριμένη περιοχή προτιμήθηκε, για την επίτευξη πιο λείας επιφάνειας συνολικά, να δημιουργηθεί τελευταίο αυτό το κομμάτι της επιφάνειας έτσι ώστε να υπάρχουν επιφάνειες στήριξης και για τις τέσσερις καμπύλες-όρια της.

Με χρήση της εντολής Fill (Παράρτημα 2) και με όρια τις καμπύλες FoSForeSpline, DeckLineFore, το επάνω μέρος του διχοτομημένου νομέα ακριβώς μετά το flat of side (8.000 στο παράδειγμα) και την πρωραία FoS ίσαλο και αντίστοιχες επιφάνειες στήριξης την επιφάνεια του flat of side, την Deckline support surface, την επιφάνεια επάνω από την πρωραία FoS ίσαλο και την επιφάνεια κάτω από την πρωραία FoS ίσαλο, δημιουργήθηκε μία επιφάνεια (*block: FILL THE GAP στις γραμμές:650-686 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py-Εικόνα 50*) η οποία ονομάστηκε: “GAP”.

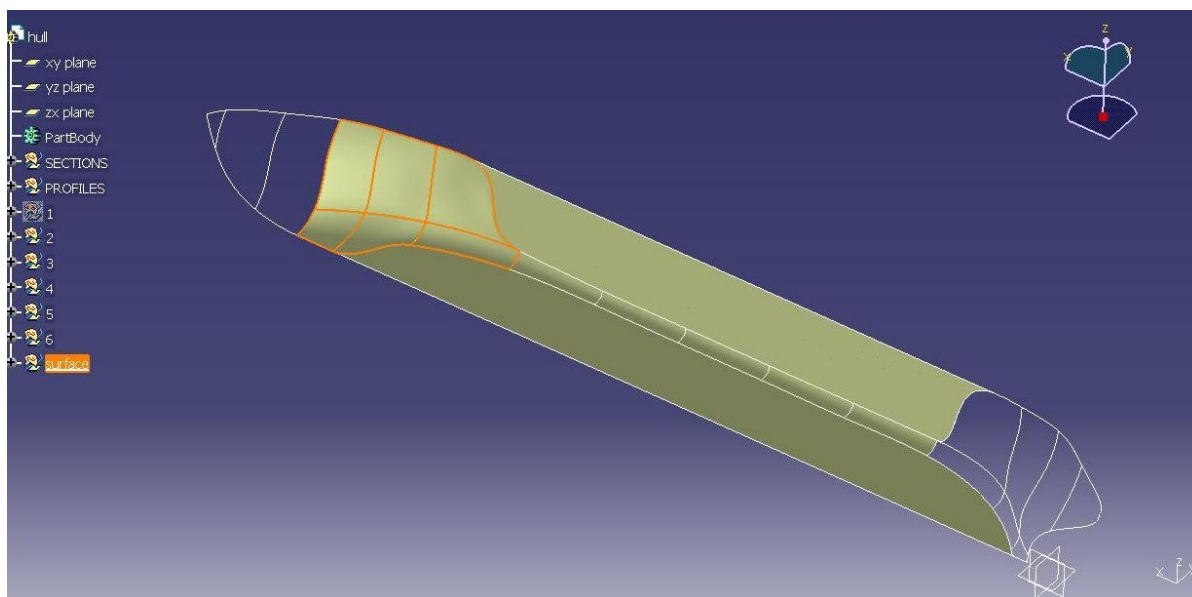


Εικόνα 50: Δημιουργία επιφάνειας “GAP”

Στη συνέχεια ενώνονται όλες οι επιφάνειες πλώραθεν του flat of side μέχρι και τον νομέα 9.000 σε μία, η οποία ονομάστηκε “ForeSurface” (*block: JOIN FORE SURFACE στις γραμμές:687-704 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py-Εικόνα 51*).

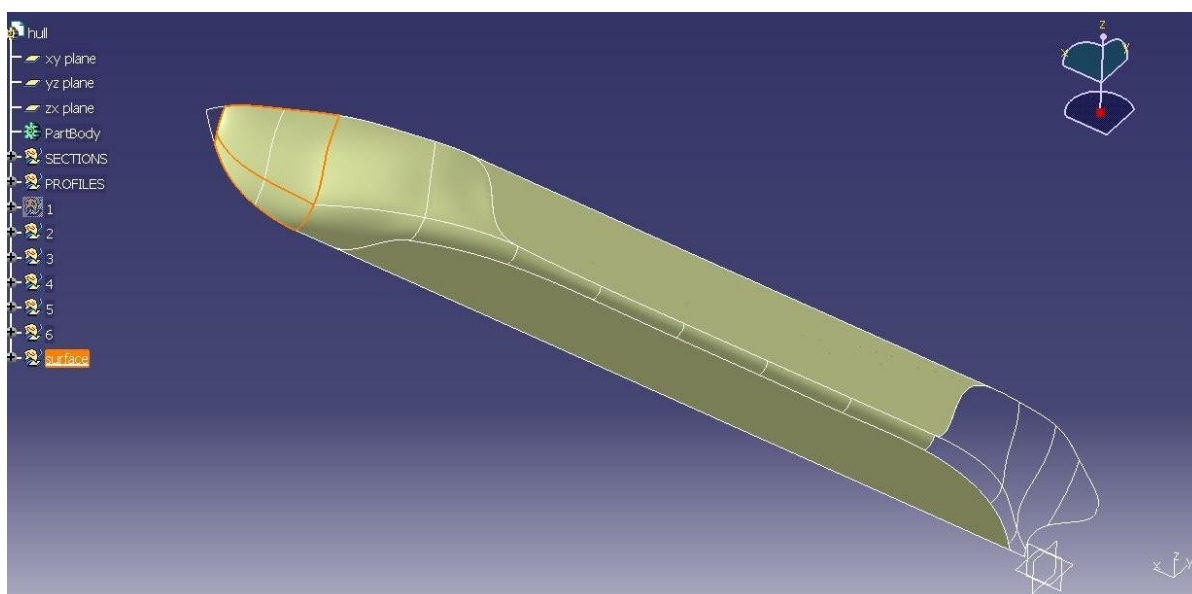
Στην περίπτωση όπου η FoS κατηγορία είναι fore ή both (δηλαδή η επιφάνεια του flat of side τελειώνει πάνω σε νομέα και όχι ανάμεσα σε δύο) όλη η επιφάνεια πλώραθεν του flat of side κατασκευάζεται με ένα loft με καμπύλες οδηγούς τις General CL και DeckLine Fore και αντίστοιχες επιφάνειες υποστήριξης τις General CL support surface και Deckline support surface και τομές όλους τους πλώραθεν του flat of side. Έτσι δημιουργείται η επιφάνεια ForeSurface (*block: Fore surface creation στις γραμμές:706-747 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py-Εικόνα 51*).

Όλες οι επιφάνειες που δημιουργήθηκαν τοποθετήθηκαν στη θέση: **hull⇒surface**, και όλες οι καμπύλες που διχοτομήθηκαν στη θέση: **hull⇒surface⇒splits**

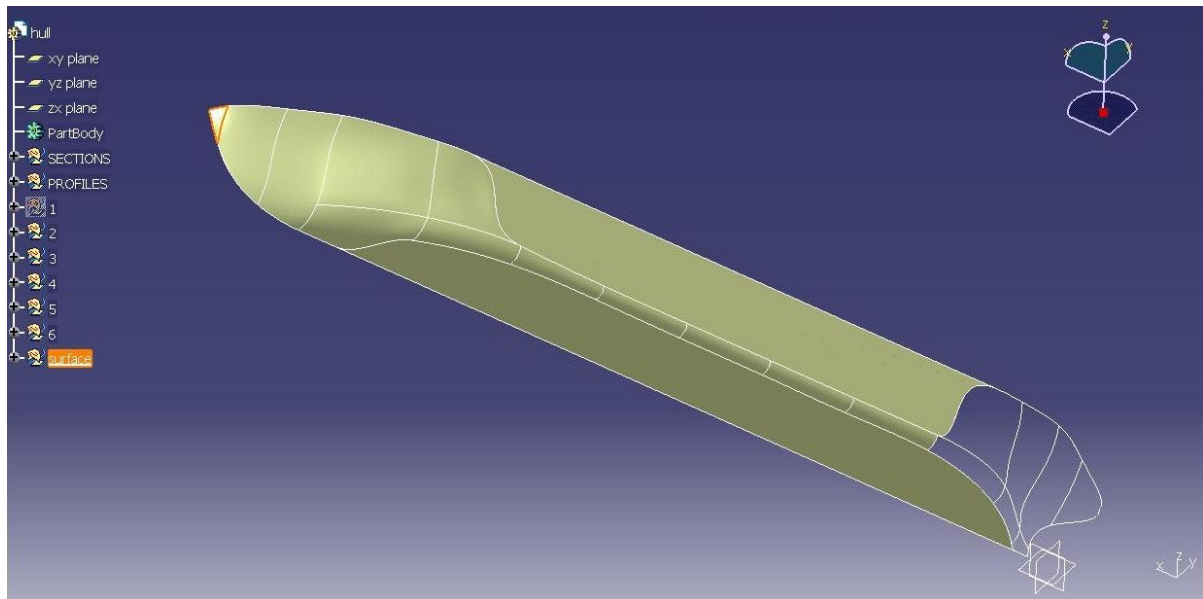


Εικόνα 51: Συνένωση προωραίων επιφανειών μέχρι το νομέα 9.000

Σε αυτό το σημείο εάν το υπό σχεδίαση πλοίο δεν διαθέτει βολβοειδή πλώρη, κατασκευάζεται και η επιφάνεια της πλώρης (πρώραθεν του νομέα 9.000). Αρχικά κατασκευάζεται το κομμάτι από το νομέα 9.000 μέχρι και το νομέα 10.00 με χρήση της εντολής loft (Παράρτημα 2), με καμπύλες οδηγούς το προφίλ της πλώρης και την DeckLineFore και επιφάνεια υποστήριξης της δεύτερης την DeckLine Support Surface, και τομές τους νομείς 9.000, 9.500 και 10.00 (*block: CREATE BowSurface στις γραμμές: 748-779 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py-Εικόνα 52*). Τέλος με χρήση της εντολής Fill (Παράρτημα 2) δημιουργείται μία επιφάνεια για να καλύψει το κενό στην άκρη του προωραίου τμήματος και συγκεκριμένα ανάμεσα στο προφίλ της πλώρης, το νομέα 10.00 και το περίγραμμα του κυρίου καταστρώματος. Ως επιφάνεια υποστήριξης χρησιμοποιείται μόνο για την DeckLineFore, η DeckLine Support Surface (*block: FILL THE TIP στις γραμμές: 780-809 του function: creation του module:surface που καλείται εντός του προγράμματος: final.py-Εικόνα 53*)

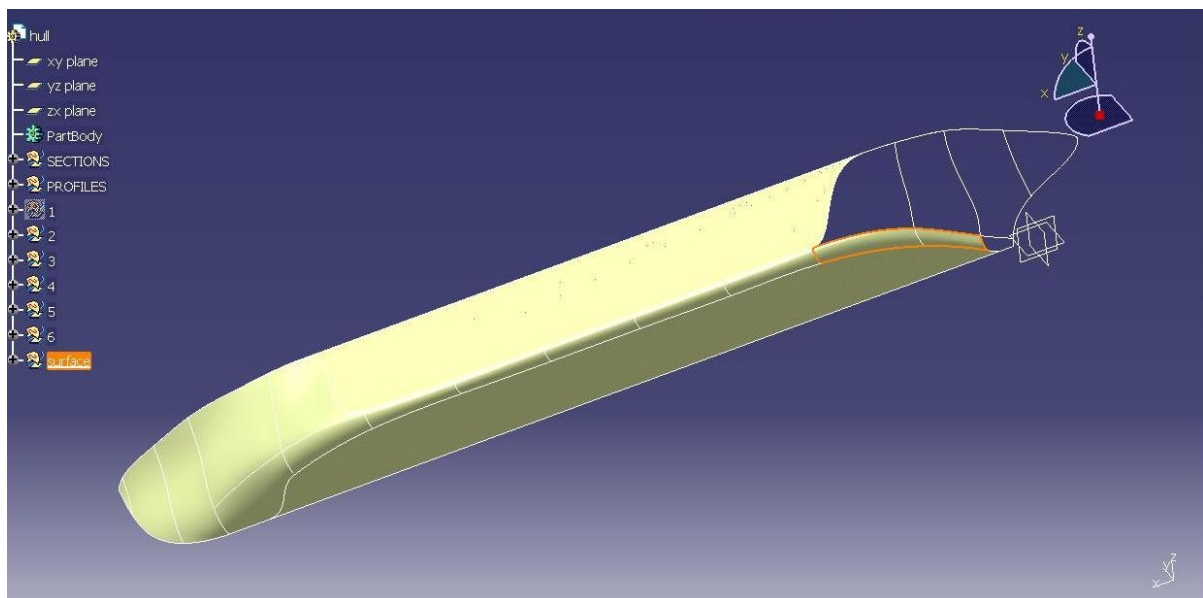


Εικόνα 52: Δημιουργία επιφάνειας μη βολβοειδούς πλώρης (νομείς 9.000-10.00)

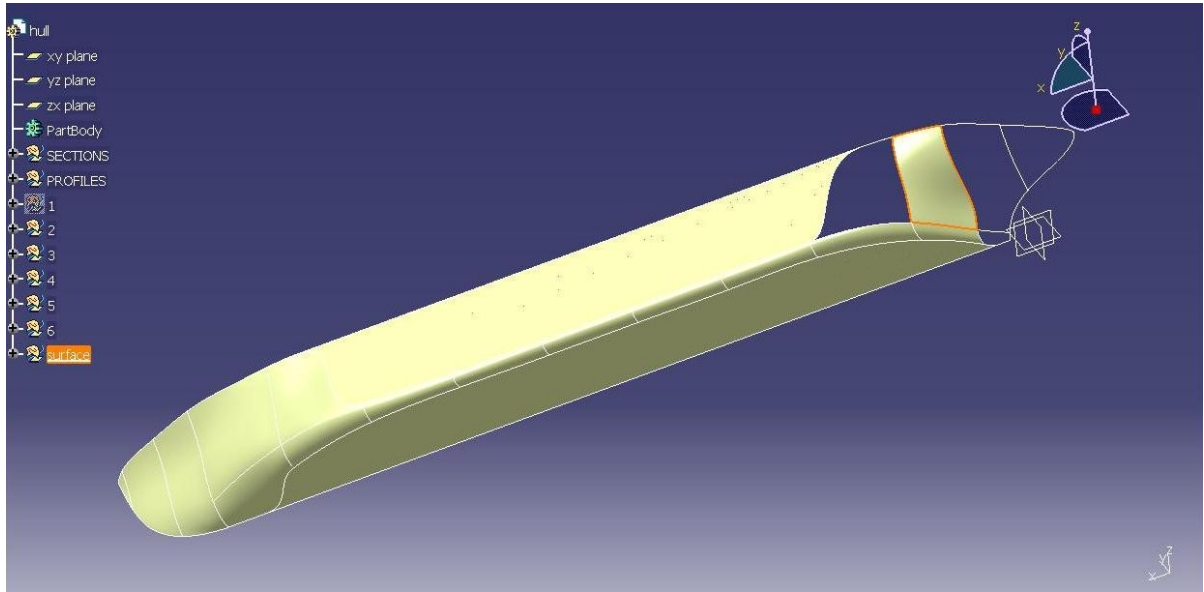


Εικόνα 53: Δημιουργία ακραίου τμήματος επιφάνειας μη βολβοειδούς πλώρης

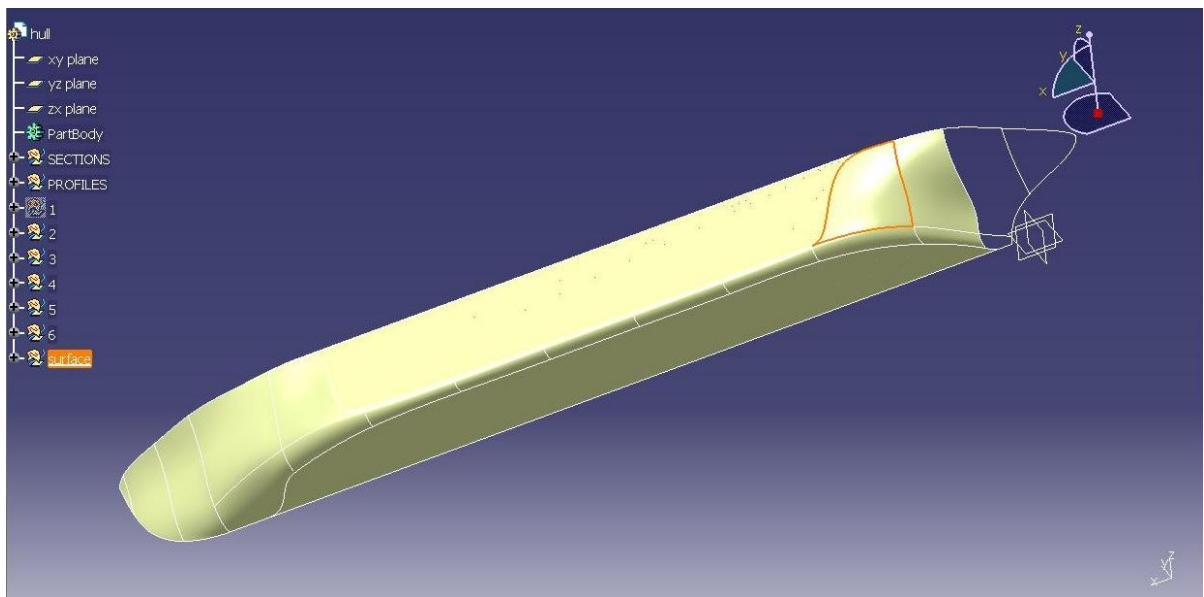
Κατά τον ίδιο τρόπο κατασκευάστηκε και η πρυμναία επιφάνεια από το τέλος του flat of side μέχρι το νομέα 0.500. Πρώτα διχοτομήθηκαν οι πρυμναίοι νομείς με καμπύλη τομής της πρυμναία FoS ίσαλο. Σε αυτήν την περίπτωση χρειάστηκε να διχοτομηθεί και η πρυμναία FoS ίσαλος καθώς εκτείνεται πρὸ μνηθεν του νομέα 0.500. Στη συνέχεια δημιουργήθηκε το κομμάτι της επιφάνειας κάτω από την πρυμναία FoS ίσαλο (Εικόνα 54), έπειτα το κομμάτι άνωθεν αυτής (Εικόνα 55) και τέλος το κομμάτι ανάμεσα στις δυο προαναφερθείσες επιφάνειες και αυτής του flat of side (Εικόνα 56).



Εικόνα 54: Δημιουργία επιφάνειας κάτω από την πρυμναία FoS ίσαλο



Εικόνα 55: Δημιουργία επιφάνειας πάνω από την προμναία FoS ίσαλο



Εικόνα 56: Δημιουργία επιφάνειας “AFT-GAP”

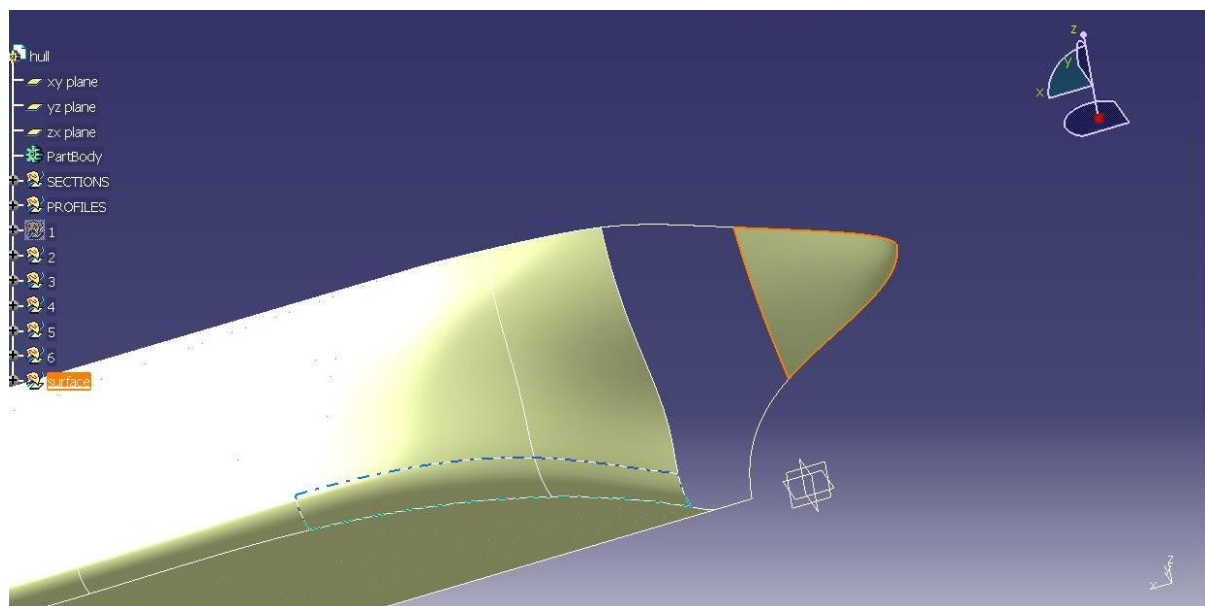
6.5.2. Δημιουργία επιφάνειας πρύμνης

Στη συνέχεια ακολουθεί η κατασκευή της επιφάνειας της πρύμνης, από το νομέα 0.500 και πρύμνηθεν. Για αυτή την κατασκευή καλείται το module: aft το οποίο περιέχει τρία functions, τα : cruiser, transomABC και transomD, τα οποία χρησιμοποιούνται για την κατασκευή των αντίστοιχων τύπων πρύμνης.

6.5.2.1. Δημιουργία Πρύμνης Καταδρομικού

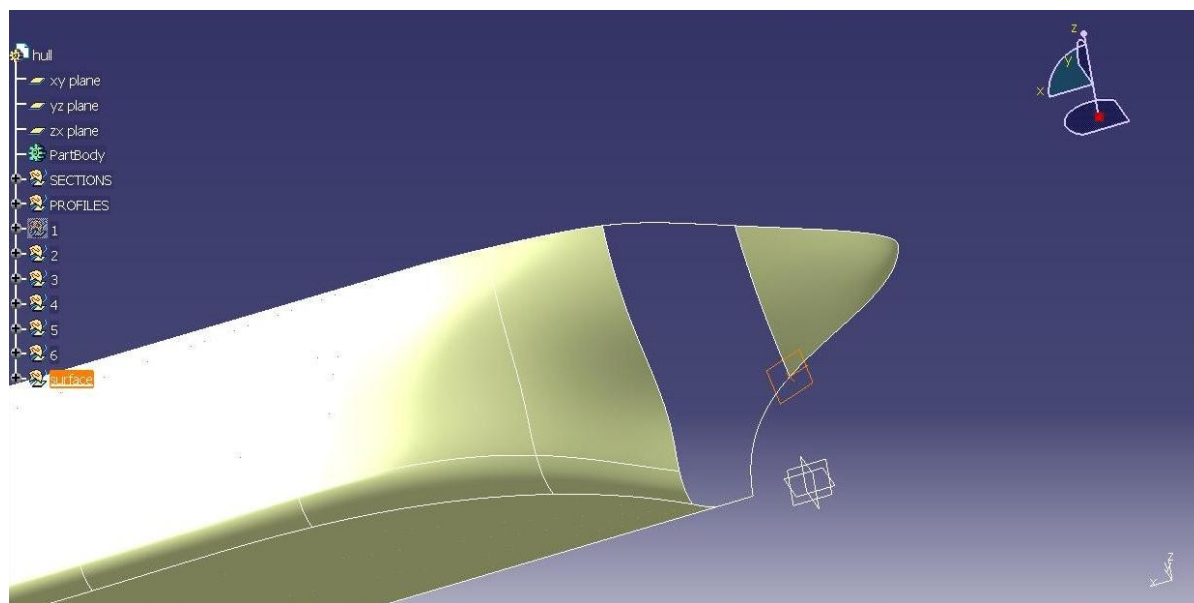
Η κατασκευή της επιφάνειας της πρύμνης καταδρομικού ξεκινάει με την κατασκευή του τμήματος (AftCruiserSurface -Εικόνα 57) ανάμεσα στις καμπύλες: νομέας 0.000 , DEckLineAft και προφίλ της πρύμνης, (*block: στις γραμμές:8-31 του function: cruiser του module:aft που καλείται εντός του προγράμματος: final.py*) χρησιμοποιώντας την εντολή Fill (Παράρτημα 2) με όρια αυτά που φαίνονται στον παρακάτω πίνακα.

AftCruiserSurface	
Καμπύλες-όρια	Επιφάνειες στήριξης
Νομέας 0.000	
Προφίλ πρύμνης	
DEckLineAft	DeckLine Support Surface



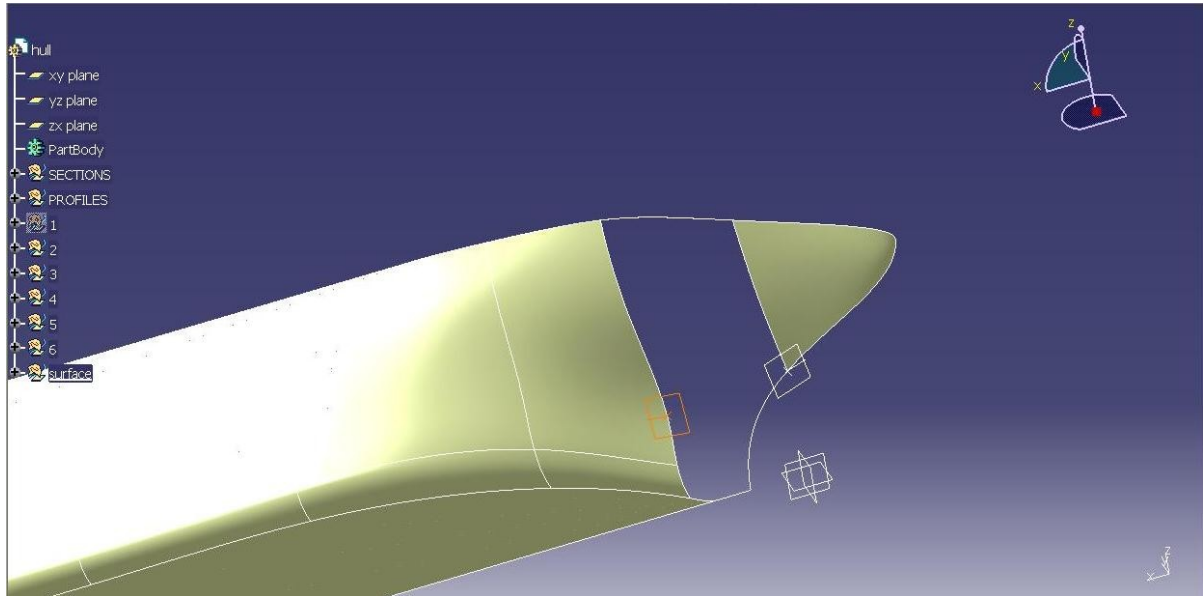
Εικόνα 57: Κατασκευή AftCruiserSurface

Έπειτα περιγράφεται η κατασκευή μιας σειράς βοηθητικών γεωμετριών (στις γραμμές: 32-104 του function: cruiser του module: aft που καλείται εντός του προγράμματος: final.py) και συγκεκριμένα ενός σημείου, ενός επιπέδου και μίας ευθείας, κατ' ακολουθία, οι οποίες θα χρησιμοποιηθούν από εδώ και στο εξής τόσο για την κατασκευή των επιφανειών της πρύμνης όσο και της βολβοειδούς πλώρης αργότερα. Πιο συγκεκριμένα για την περίπτωση της πρύμνης καταδρομικού έχουμε (Εικόνα 58): Αρχικά δημιουργείται το κατώτατο ακραίο σημείο του νομέα 0.000 (LowerPoint_0.000) και σε αυτό το σημείο ακριβώς κατασκευάζεται ένα επίπεδο (WL3AftPlane) εφαπτόμενο στην επιφάνεια που κατασκευάστηκε αμέσως πριν (AftCruiserSurface) με χρήση της εντολής AddNewPlaneTangent (Παράρτημα 2). Στη συνέχεια κατασκευάζεται μία ευθεία (TangentLine3Aft) με αρχή το παραπάνω σημείο, η οποία θα είναι μία προβολή του άξονα X πάνω στο επίπεδο WL3AftPlane και θα έχει μήκος 2m με φορά προς την πρύμνη, με χρήση της εντολής AddNewLinePtDirOnSupport (Παράρτημα 2).



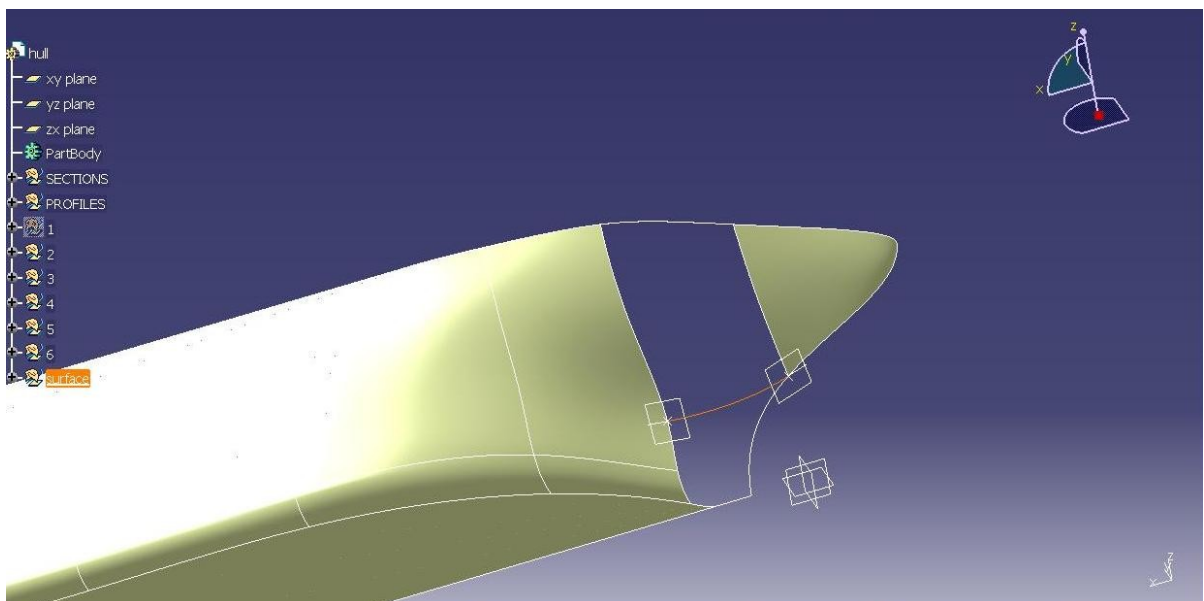
Εικόνα 58: Βοηθητικές γεωμετρικές επί του νομέα 0.000

Στη συνέχεια θα κατασκευαστούν οι ίδιες βοηθητικές γεωμετρικές που περιγράφηκαν νωρίτερα, επί του νομέα 0.500 αυτή τη φορά. Το σημείο (PointOnSection0.500) που θα χρησιμοποιηθεί θα είναι ένα σημείο στο 20% του μήκους του νομέα 0.500, εάν η Z συντεταγμένη του fos σημείου του πρυμναίου νομέα του flat of side είναι μεγαλύτερη από αυτή του κατώτατου σημείου του νομέα 0.000. Σε αντίθετη περίπτωση το σημείο λαμβάνεται στο 30% του μήκους του διχοτομημένου νομέα άνωθεν της πρυμναίας FoS ισάλου. Το επίπεδο (WL3forePlane) σχεδιάζεται εφαπτόμενο στην επιφάνεια AftSurface και η ευθεία (TangentLine3Fore) σχεδιάζεται με φορά προς την πλώρη (Εικόνα 59).



Εικόνα 59: Βοηθητικές γεωμετρίες επί του νομέα 0.500

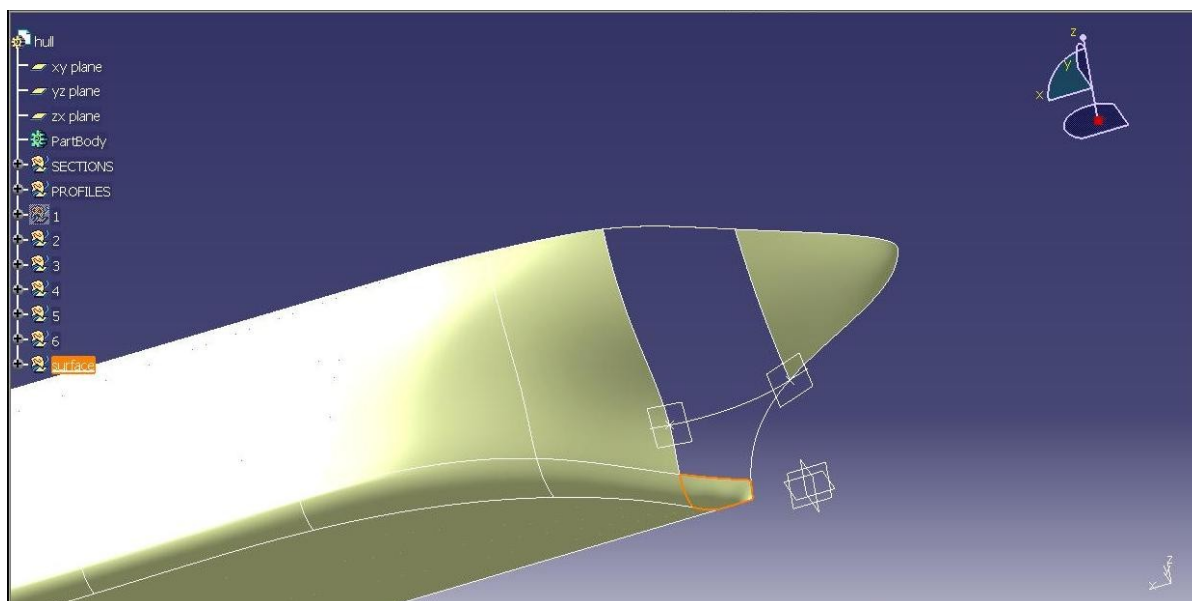
Τέλος κατασκευάζεται ένα spline (*block: SPLINE 3 CREATION στις γραμμές: 105-123 του function: cruiser του module:aft που καλείται εντός του προγράμματος: final.py*) ανάμεσα στα δύο παραπάνω σημεία, το οποίο θα είναι εφαπτόμενο στις δύο ευθείες που δημιουργήθηκαν προηγουμένα (Εικόνα 60).



Εικόνα 60: Δημιουργία πρυμναίου βοηθητικού spline

Στη συγκεκριμένη περίπτωση ,που παρουσιάζεται, η πρυμναία FoS ίσαλος βρίσκεται κάτω από το κατώτατο ακραίο σημείο του νομέα 0.000, συνεπώς η επιφάνεια κάτω από το πρυμναίο βοηθητικό spline (spline3) και ανάμεσα στους νομείς 0.000 και 0.500 κατασκευάζεται σε δύο τμήματα. Το πρώτο τμήμα (LOWER_0.5-0_SURFACE-Εικόνα 61) το οποίο βρίσκεται κάτω από την πρυμναία FoS ίσαλο κατασκευάζεται με την εντολή Fill (Παράρτημα 2) με όρια και επιφάνειες στήριξης αυτά που φαίνονται στον παρακάτω πίνακα (*block: FILL LOWER_0.5-0_SURFACE στις γραμμές: 125-155 του function: cruiser του module:aft που καλείται εντός του προγράμματος: final.py*).

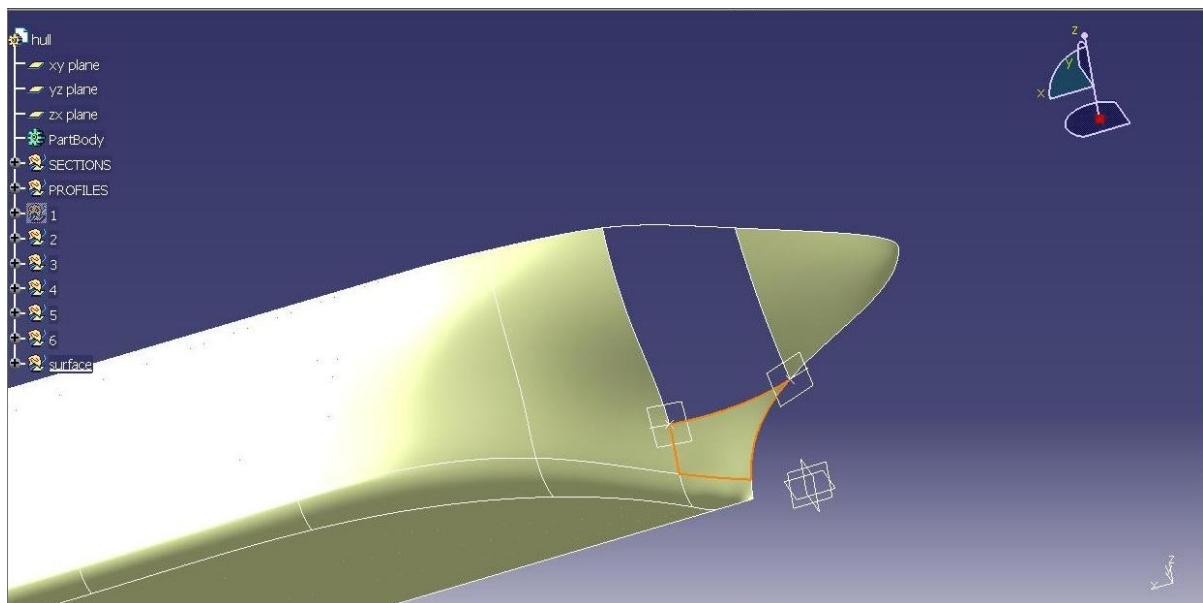
<u>LOWER 0.5-0 SURFACE</u>	
Καμπύλες-όρια	Επιφάνειες Στήριξης
Spline3	
Νομέας 0.500	AftSurface
GeneralCL	General CL Support Surface
Προφίλ πρύμνης	



Εικόνα 61: Κατασκευή της LOWER_0.5-0_SURFACE

Το δεύτερο τμήμα (MIDDLE_0.5-0_SURFACE-Εικόνα 62) το οποίο βρίσκεται πάνω από την πρυμναία FoS ίσαλο κατασκευάζεται με την εντολή Fill (Παράρτημα 2) με όρια και επιφάνειες στήριξης αυτά που φαίνονται στον παρακάτω πίνακα (*block: FILL MIDDLE_0.5-0_SURFACE στις γραμμές: 156-186 του function: cruiser του module:aft που καλείται εντός του προγράμματος: final.py*). Αυτές οι δύο επιφάνειες ενοποιούνται σε μία την LowerCruiserSurface (*block: JOIN SURFACES BENEATH SPLINE3 στις γραμμές: 187-197 του function: cruiser του module:aft που καλείται εντός του προγράμματος: final.py*)

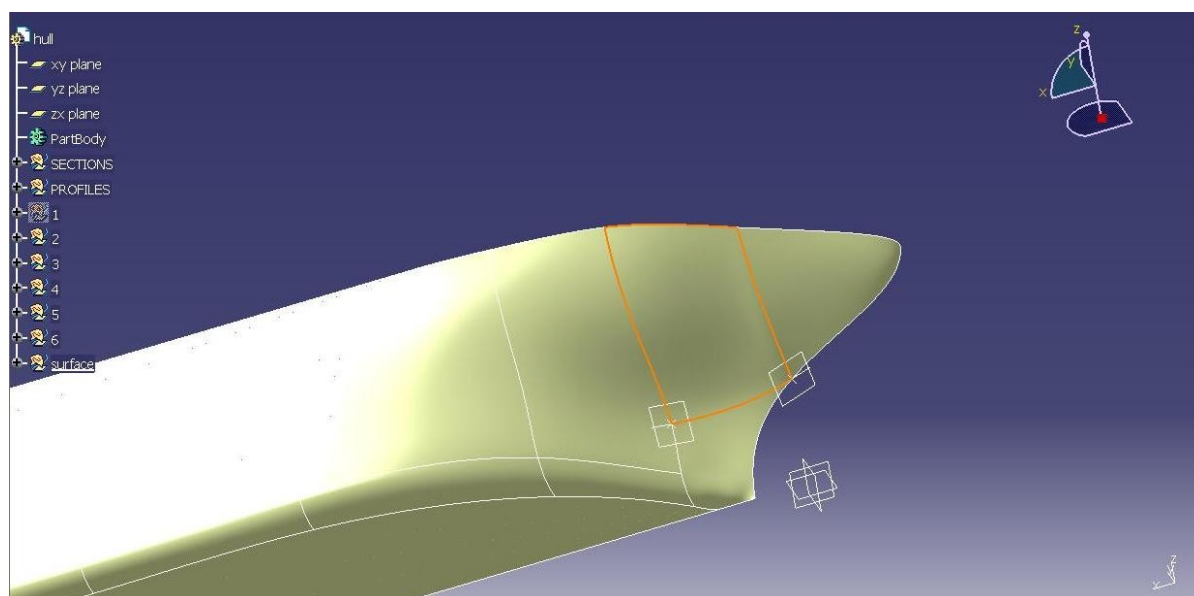
<u>MIDDLE 0.5-0 SURFACE</u>	
Καμπύλες-όρια	Επιφάνειες Στήριξης
Spline3	
Νομέας 0.500	AftSurface
AftFoSWL	LOWER_0.5-0_SURFACE
Προφίλ πρύμνης	



Εικόνα 62: Κατασκευή της MIDDLE_0.5-0_SURFACE

Τέλος κατασκευάζεται η επιφάνεια άνωθεν της πρυμναίας FoS ισάλου (TopCruiserSurface -Εικόνα 63), με χρήση της εντολής Fill (Παράρτημα 2) και καμπύλες-όρια αυτές που φαίνονται στον παρακάτω πίνακα.

<u>TOP_0.5-0 SURFACE</u>	
Καμπύλες-όρια	Επιφάνειες Στήριξης
Spline3	LowerCruiserSurface
Νομέας 0.500	AftSurface
GeneralDeckLine	General DeckLine Support Surface
Νομέας 0.000	AftCruiserSurface



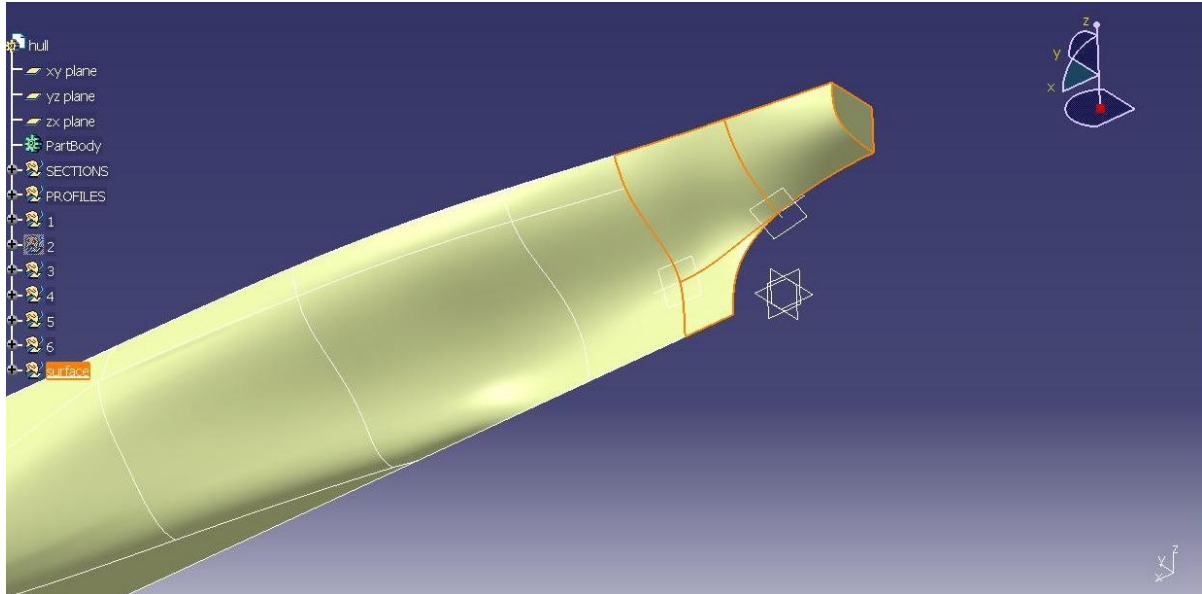
Εικόνα 63: Κατασκευή TOP_0.5-0 SURFACE

Στην περίπτωση όπου η πρυμναία FoS ισάλος βρίσκεται πιο ψηλά από το κατώτατο ακραίο σημείο του νομέα η κατασκευή των επιφανειών της πρύμνης παραμένει η ίδια, μόνο που το κομμάτι κάτω από το πρυμναίο βοηθητικό spline (LowerCruiserSurface) κατασκευάζεται ολόκληρο και όχι με δύο τμήματα όπως πριν.

Όλες οι επιφάνειες που κατασκευάστηκαν και απαρτίζουν την επιφάνεια της πρύμνης τοποθετήθηκαν στη θέση: **hull**⇒**surface** και οποιεσδήποτε βοηθητικές γεωμετρίες σχεδιάστηκαν τοποθετήθηκαν στη θέση: **hull**⇒**surface**⇒**cruiser**.

6.5.2.2. Δημιουργία Πρύμνης Άβακα τύπου A,B,C

Η κατασκευή αυτού του τύπου πρύμνης διαφέρει από την πρύμνη καταδρομικού ελάχιστα. Βασική διαφορά είναι η κατασκευή της επιφάνειας του καθρέφτη. Κατά τα άλλα, πάλι κατασκευάστηκε πρώτα το κομμάτι πρύμνηθεν του νομέα 0.000, στη συνέχεια σχεδιάστηκε η βοηθητική γεωμετρία, κατά τον ίδιο τρόπο με πριν, έπειτα σχεδιάστηκε η επιφάνεια κάτω από το πρυμναίο βοηθητικό spline και τέλος η επιφάνεια άνωθεν αυτού (Εικόνα 64).

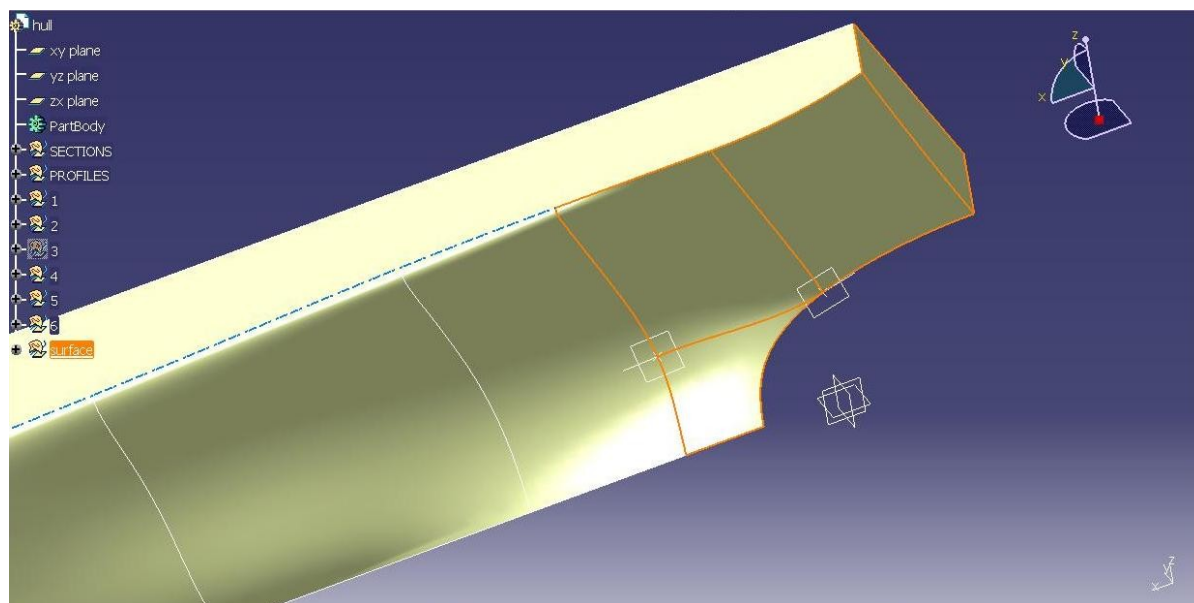


Εικόνα 64: Κατασκευή πρύμνης άβακα τύπου ABC

Σε αυτή την περίπτωση η βοηθητική γεωμετρία τοποθετήθηκε στη θέση **hull⇒surface⇒transom** και όλες οι επιφάνειες στη θέση **hull⇒surface**.

6.5.2.3. Δημιουργία Πρύμνης Άβακα τύπου D

Και σε αυτόν τον τύπο πρύμνης οι διαφορές στη σχεδίαση παραμένουν ελάχιστες. Όπως και πριν κατασκευάζεται η επιφάνεια του καθρέφτη, μετά η επιφάνεια πρύμνηθεν του νομέα 0.000, στη συνέχεια η βοηθητική γεωμετρία και έπειτα η επιφάνεια ανάμεσα στους νομείς 0.500 και 0.000 σε δύο τμήματα, πάνω και κάτω από το πρυμναίο βοηθητικό spline (Εικόνα 65)



Εικόνα 65: Κατασκευή πρύμνης άβακα τύπου D

Σε αυτή την περίπτωση η βοηθητική γεωμετρία τοποθετήθηκε στη θέση **hull⇒surface⇒transom** και όλες οι επιφάνειες στη θέση **hull⇒surface**.

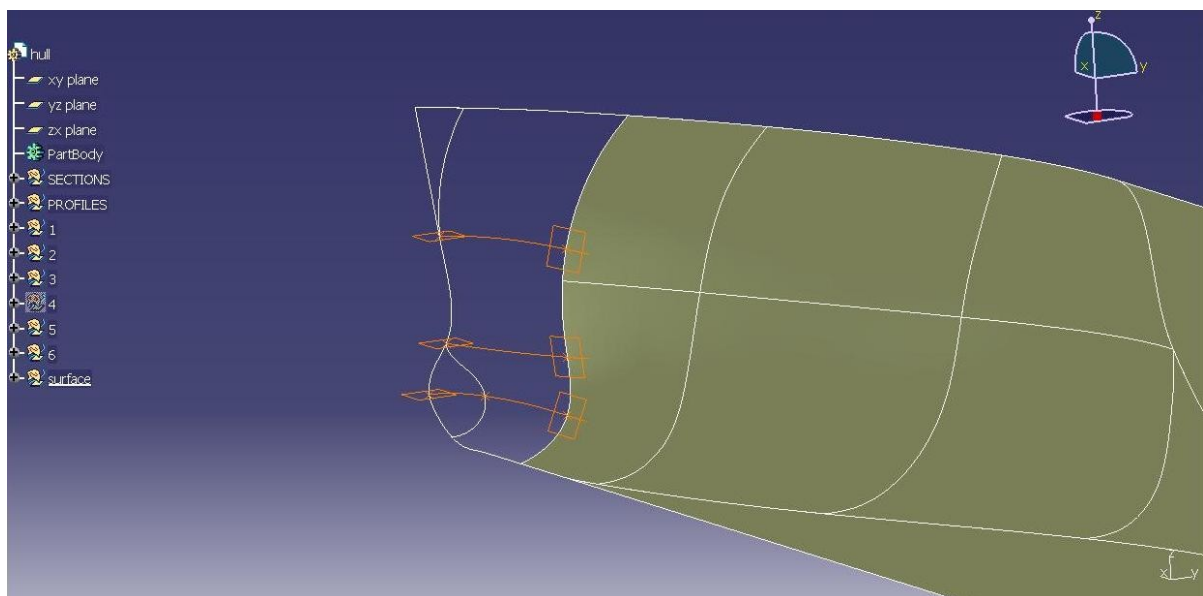
6.5.3.Κατασκευή επιφάνειας βολβοειδούς πλώρης

Στην περίπτωση όπου το μοντέλο διαθέτει βολβοειδή πλώρα, για την σχεδίαση της επιφάνειας της πλώρης (πλώραθεν του νομέα 9.500) είναι απαραίτητη η δημιουργία μιας σειράς από βοηθητικά γεωμετρικά στοιχεία, παρόμοια με αυτά που σχεδιάστηκαν για να διευκολύνουν την κατασκευή των επιφανειών της πρύμνης (στις γραμμές: 7-227 του function: bulb του module:surface που καλείται εντός του προγράμματος: final.py). Πιο συγκεκριμένα θα πραγματοποιηθεί η εξής διαδικασία τρεις φορές με διαφορετικό σημείο εκκίνησης κάθε φορά.

Αρχικά σχεδιάζεται ένα επίπεδο (WL Plane) παράλληλο στο XY το οποίο διέρχεται από το σημείο εκκίνησης. Στη συνέχεια σχεδιάζεται το σημείο τομής (WL Intersection_9.500) του νομέα 9.500 με αυτό το επίπεδο. Έπειτα σχεδιάζεται ένα επίπεδο (Tangent Plane) το οποίο διέρχεται από αυτό το σημείο και είναι εφαπτόμενο στην ForeSurface. Στη συνέχεια σχεδιάζεται μια ευθεία (Tangent Line) με αρχή το σημείο τομής του WL Plane με το νομέα 9.500, φορά προς την πρύμνη και μήκους 2m, η οποία αποτελεί προβολή του άξονα X στο επίπεδο Tangent Plane. Τέλος σχεδιάζεται ένα spline με αρχή το σημείο εκκίνησης και τέλος το σημείο τομής του WL Plane με το νομέα 9.500 στο οποίο επιβάλλεται ως εφαπτομενικό διάνυσμα η Tangent Line.

Η παραπάνω διαδικασία επαναλήφθηκε τρεις φορές με σημεία εκκίνησης τα εξής: **1)** το κατώτατο ακραίο σημείο του επάνω μέρους του νομέα 10.00 (spline2), **2)** το ανώτατο ακραίο σημείο του κάτω μέρους του νομέα 10.00 (spline1), και **3)** το ακραίο σημείο του προφίλ της πλώρης κατά τον άξονα των X (BulbWL). Τα παραπάνω σημεία χρειάστηκε να σχεδιαστούν πριν την σχεδίαση των βοηθητικών γεωμετρικών στοιχείων.

Στην περίπτωση 3 εκτός από το σημείο τομής του WL Plane με το νομέα 9.500 σχεδιάστηκε και αυτό με το κάτω μέρος του νομέα 10.00, και συμπεριλήφθηκε και αυτό στη σχεδίαση του spline.

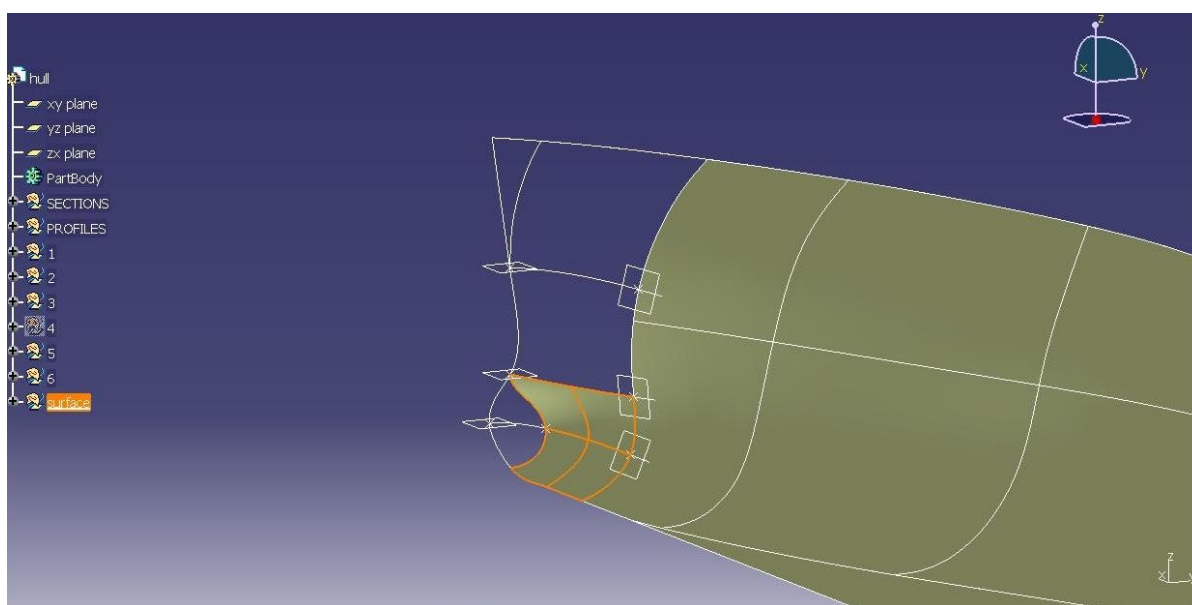


Εικόνα 66: Κατασκευή βοηθητικών γεωμετρικών στοιχείων για τη σχεδίαση της επιφάνειας βολβοειδούς πλώρης.

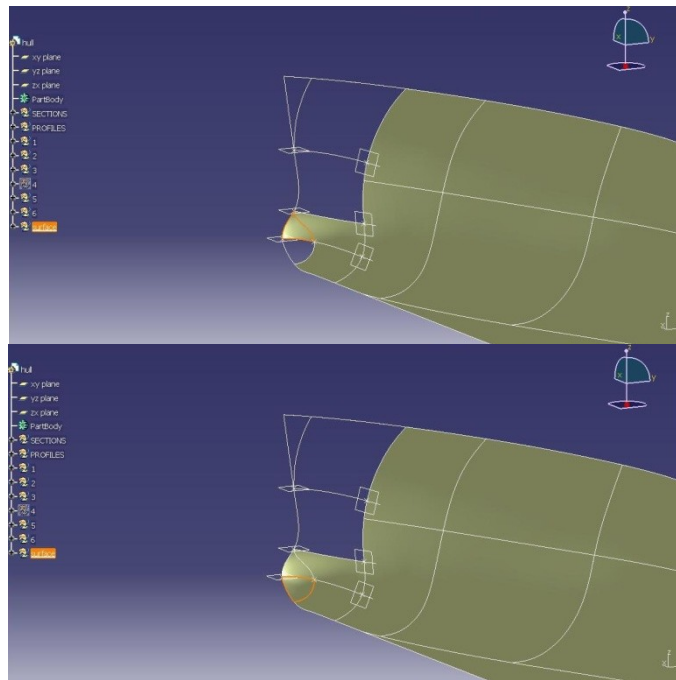
Στη συνέχεια, αφού κατασκευάστηκαν τα τρία προραία βοηθητικά splines, διχοτομούνται, το προφίλ της πλώρης και ο νομέας 9.500 σε τρία (Bow1, Bow2, Bow3) και δύο τμήματα αντίστοιχα για την ευκολότερη κατασκευή των επιφανειών της βολβοειδούς πλώρης (στις γραμμές: 228-267 του function: bulb του module:surface που καλείται εντός του προγράμματος: final.py).

Η επιφάνεια αυτού του τύπου πλώρης κατασκευάζεται σε τρία στάδια. Αρχικά κατασκευάζεται το τμήμα κάτω από το spline2, στη συνέχεια αυτό ανάμεσα στο spline1 και το spline2 και τέλος αυτό πάνω από το spline1. Το κατώτερο τμήμα της επιφάνειας της βολβοειδούς πλώρης (BulboousBowLowerSurface-Εικόνα 67) κατασκευάζεται με τη βοήθεια της εντολής Loft (στις γραμμές: 268-352 του function: bulb του module:surface που καλείται εντός του προγράμματος: final.py). Ως καμπύλες οδηγί χρησιμοποιήθηκαν, το spline2, η BulbWL και το κάτω μέρος του προραίου προφίλ, και ως τομές το κάτω τμήμα του νομέα 10.00 και το κατώτερο τμήμα του νομέα 9.500 (αυτό κάτω από το spline2). Ακολούθως κατασκευάζονται δύο επιφάνειες στην κορυφή του βολβού με τη βοήθεια της εντολής Fill(Εικόνα 68).

Surface 1		Surface 2	
Καμπύλες-όρια	Επιφάνειες στήριξης	Καμπύλες-όρια	Επιφάνειες στήριξης
BulbWL		BulbWL	Surface 1
LOWER	BulboousBowLowerSurface	LOWER	BulboousBowLowerSurface
Bow2		Bow3	



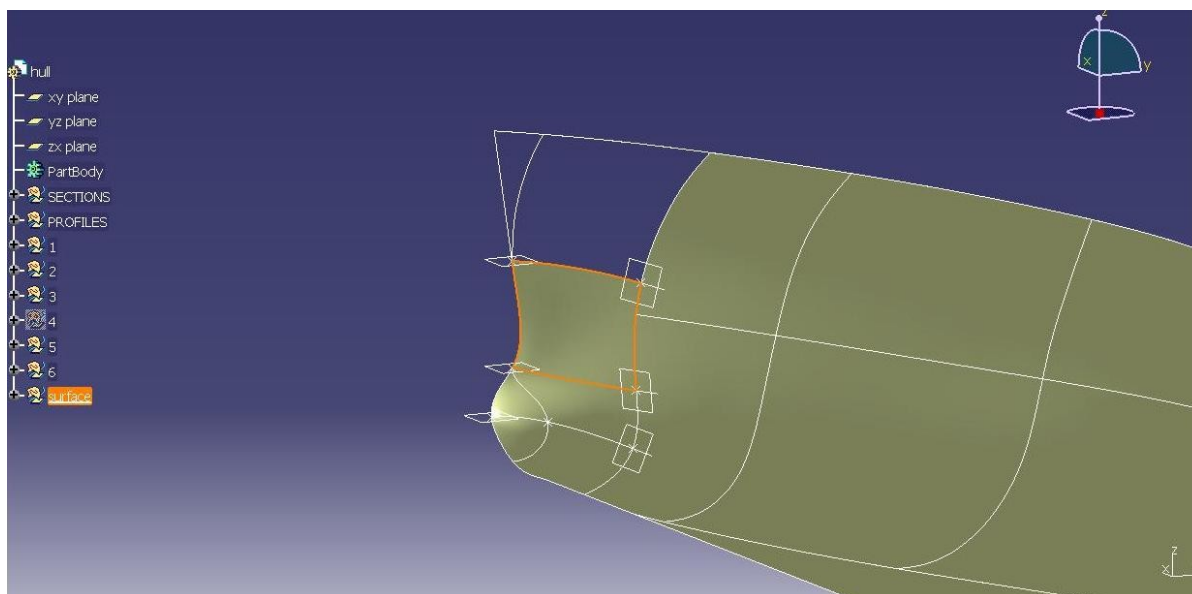
Εικόνα 67: Κατασκευή του κάτω τμήματος της επιφάνειας της βολβοειδούς πλώρης.



Εικόνα 68: Κατασκευή της κορυφής του βολβού

Στη συνέχεια, πάλι με τη βοήθεια της εντολής Fill (στις γραμμές: 353-383 του *function: bulb* του *module:surface* που καλείται εντός του προγράμματος: *final.py*) κατασκευάζεται η ενδιάμεση επιφάνεια της βολβοειδούς πλώρης (BulboousBowMiddleSurface-Εικόνα 69).

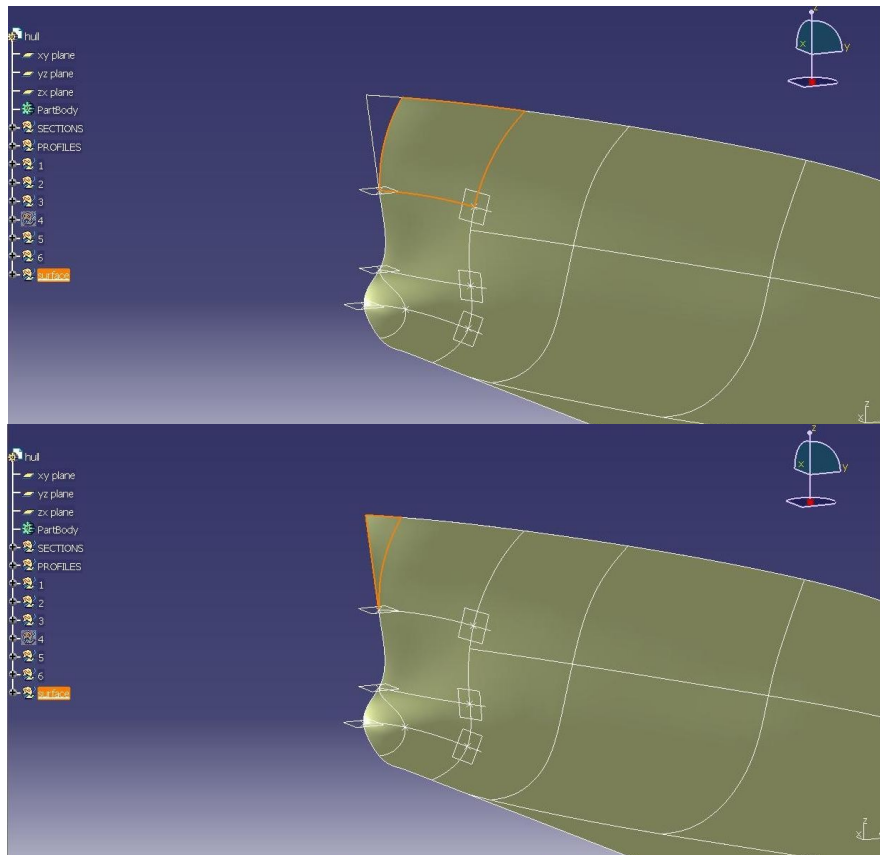
BulboousBowMiddleSurface	
Καμπύλες-όρια	Επιφάνειες στήριξης
Spline2	BulboousBowLowerSurface
Bow1	
Spline1	
9.500	ForeSurface



Εικόνα 69: Κατασκευή ενδιάμεσης επιφάνειας βολβοειδούς πλώρης.

Τέλος κατασκευάζεται το επάνω μέρος της επιφάνειας της βολβοειδούς πλώρης , επίσης με χρήση της εντολής Fill(στις γραμμές: 384-443 του function: bulb του module:surface που καλείται εντός του προγράμματος: final.py), και πιο συγκεκριμένα σε δύο κομμάτια (BulboousBowTopSurface -Εικόνα 70).

BulboousBowTopSurface		BulboousBowTip	
Καμπύλες-όρια	Επιφάνειες στήριξης	Καμπύλες-όρια	Επιφάνειες στήριξης
Spline1	BulboousBowMiddleSurface	Bow1	
9.500	ForeSurface	UPPER	BulboousBowTopSurface
GeneralDeckLine	General DeckLine Support Surface	GeneralDeckLine	General DeckLine Support Surface
UPPER			



Εικόνα 70: Κατασκευή του επάνω τμήματος της επιφάνειας της βολβοειδούς πλώρης.

Τελικά όλες οι παραπάνω επιφάνειες ενοποιούνται (στις γραμμές: 444-473 του *function: bulb* του *module:surface* που καλείται εντός του προγράμματος: *final.py*) σε μία (BULBOOUS BOW). Σε αυτή την περίπτωση η βοηθητική γεωμετρία τοποθετήθηκε στη θέση **hull**⇒**surface**⇒**Bulb** και όλες οι επιφάνειες στη θέση **hull**⇒**surface**.

7. ΠΑΡΑΔΕΙΓΜΑΤΑ

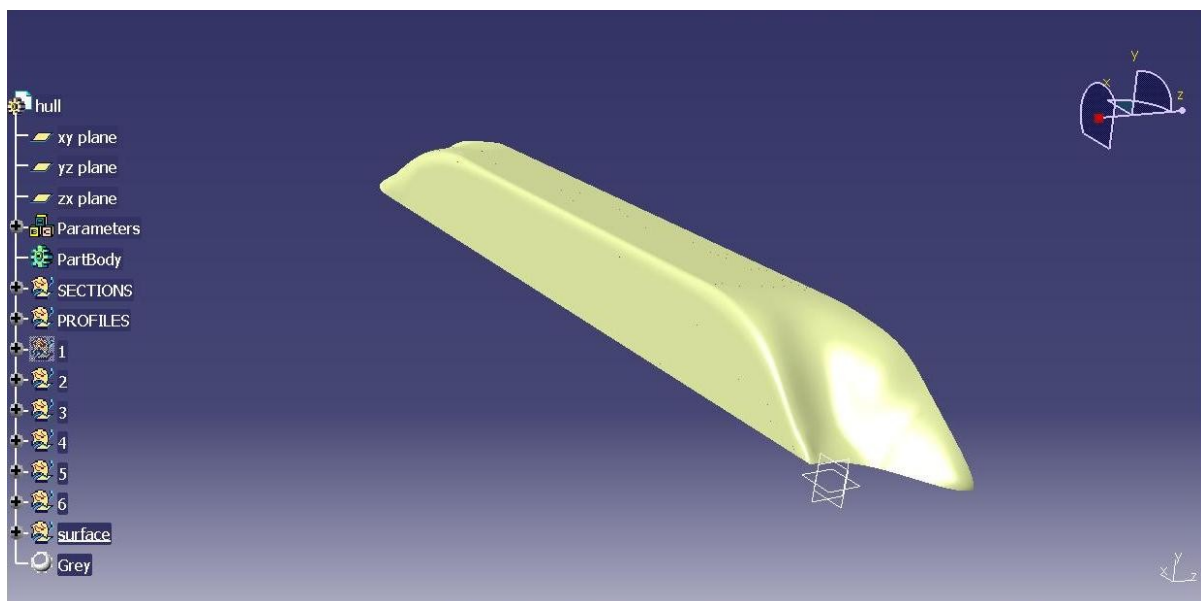
Στη συνέχεια παρουσιάζονται έξι παραδείγματα τρισδιάστατων μοντέλων κατασκευασμένων με το πρόγραμμα final.py το οποίο περιγράφηκε λεπτομερώς στα προηγούμενα κεφάλαια. Κάθε παράδειγμα αντιστοιχεί σε μία από τις έξι κατηγορίες που δημιουργήθηκαν με βάση τους πιθανούς συνδυασμούς πλώρης και πρύμνης. Στον παρακάτω πίνακα φαίνονται τα λεπτομερή χαρακτηριστικά (μήκος μεταξύ καθέτων, πλάτος, κούλο, συντελεστής γάστρας, διαμήκης θέση του κέντρου άντωσης και οι επιλεγμένες οικογένειες πλώρης και πρύμνης) του κάθε μοντέλου.

CATEGORY	WITHOUT BULB			WITH BULB		
	1	2	3	4	5	6
L_{BP}	200	150	180	220	200	250
B	40	20	30	50	40	50
D	20	10	12	23	20	33
LCB	4	2	3	5	4	2
C_B	0.8	0.55	0.6	0.75	0.6	0.68
FORE	U1F75	V2F55	V2F55	B51F70	B42F55	B82F65
AFT	T1A85	CA2A55	CD2A65	U1A80	CB2A65	CD2A70

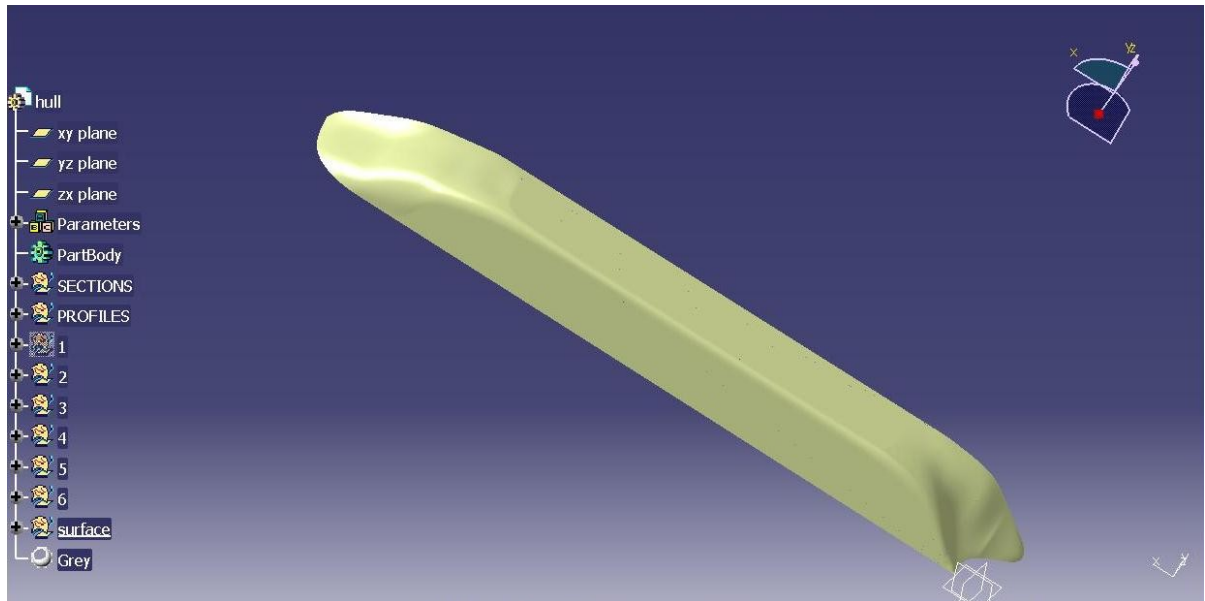
7.1.Κατηγορία 1: (πλώρη χωρίς βολβό , πρύμνη καταδρομικού)



Εικόνα 71Α: Τρισδιάστατο μοντέλο κατηγορίας 1 (πλάγια όψη)

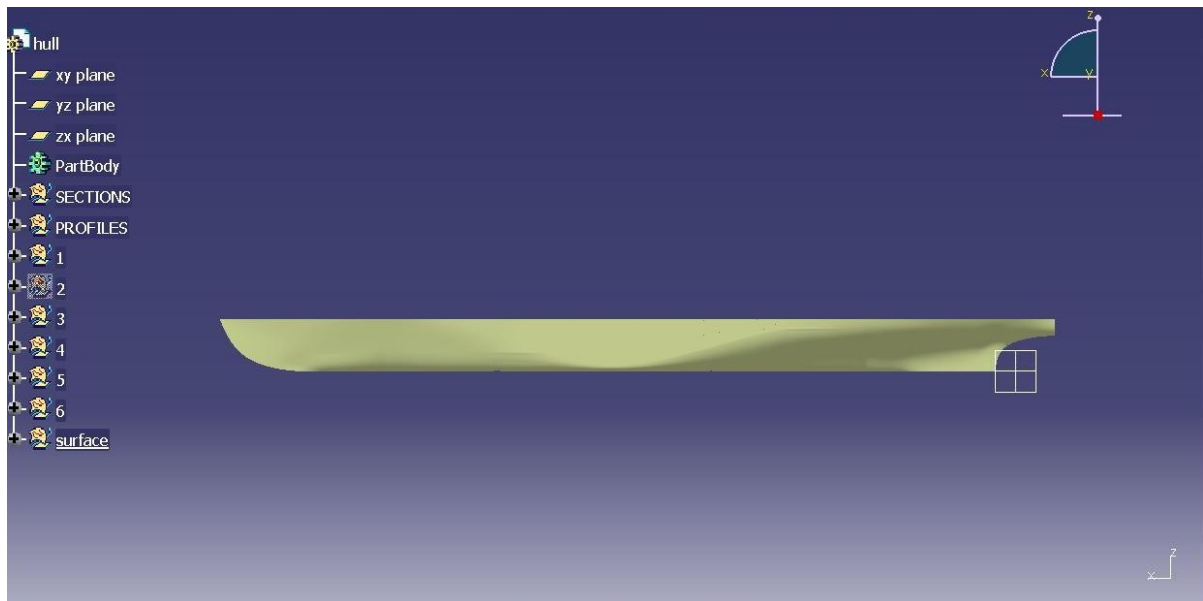


Εικόνα 71Β: Τρισδιάστατο μοντέλο κατηγορίας 1 (τρειςδιάστατη όψη 1)

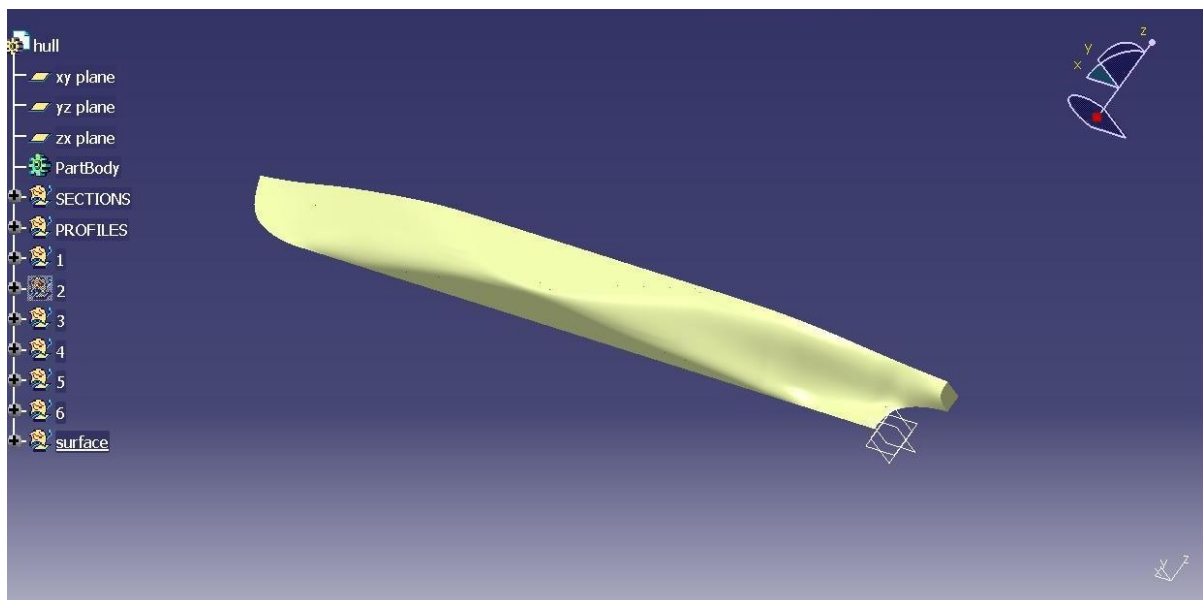


Εικόνα 71Γ: Τρισδιάστατο μοντέλο κατηγορίας 1 (τρισδιάστατη όψη 2)

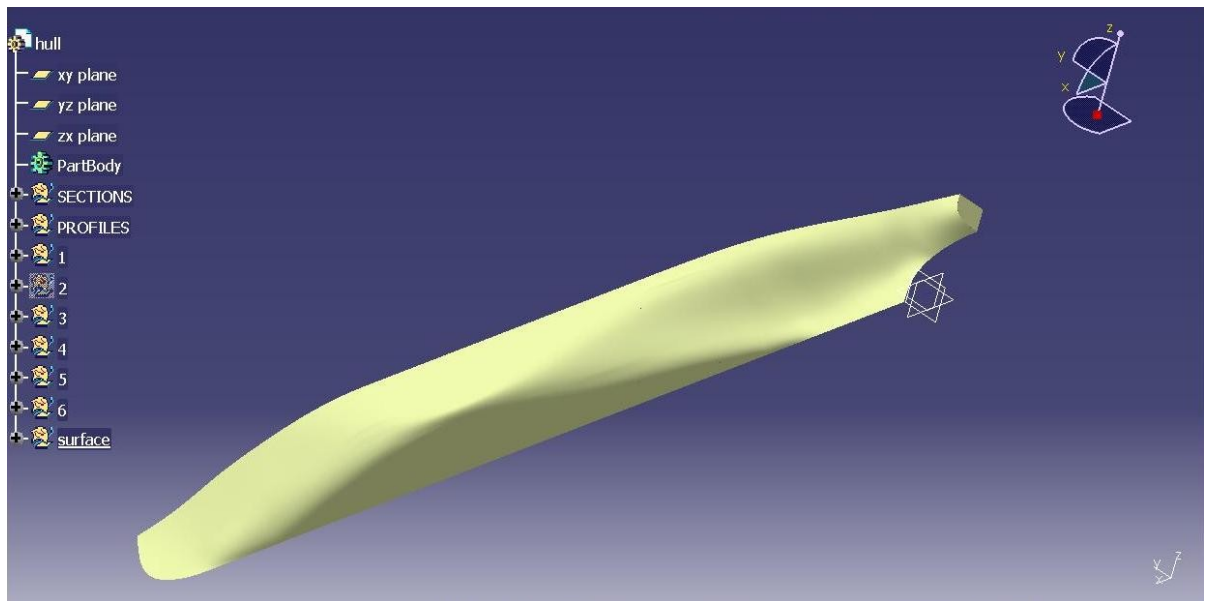
7.2.Κατηγορία 2: (πλώρη χωρίς βολβό , πρύμνη άβακας A-B-C)



Εικόνα 72Α: Τρισδιάστατο μοντέλο κατηγορίας 2 (πλάγια όψη)

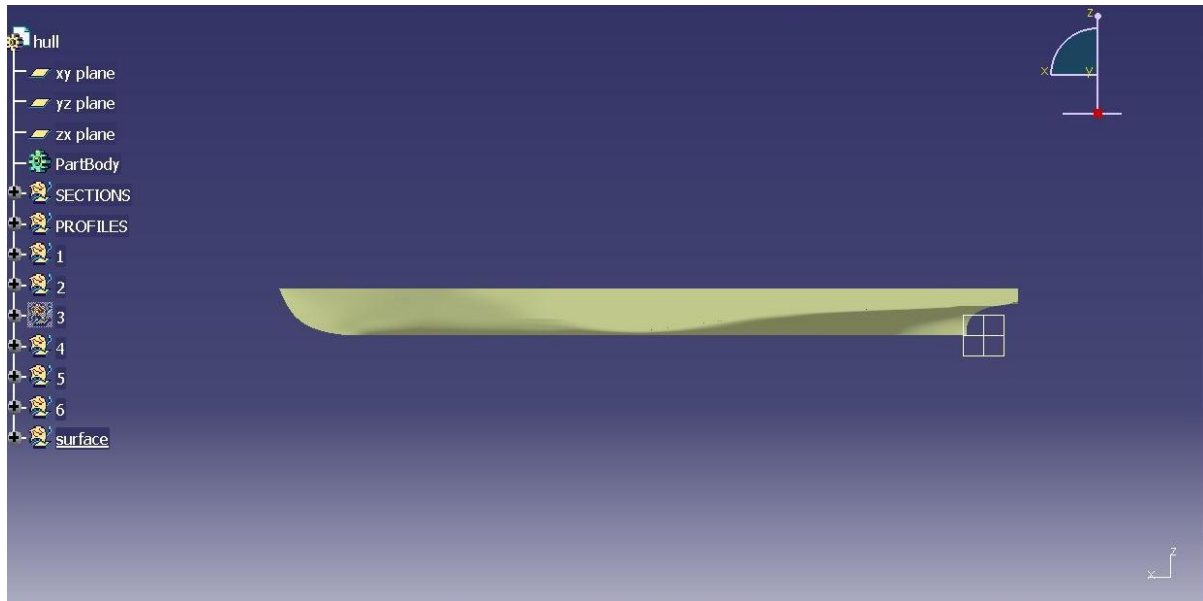


Εικόνα 72Β: Τρισδιάστατο μοντέλο κατηγορίας 2 (τρειςδιάστατη όψη 1)

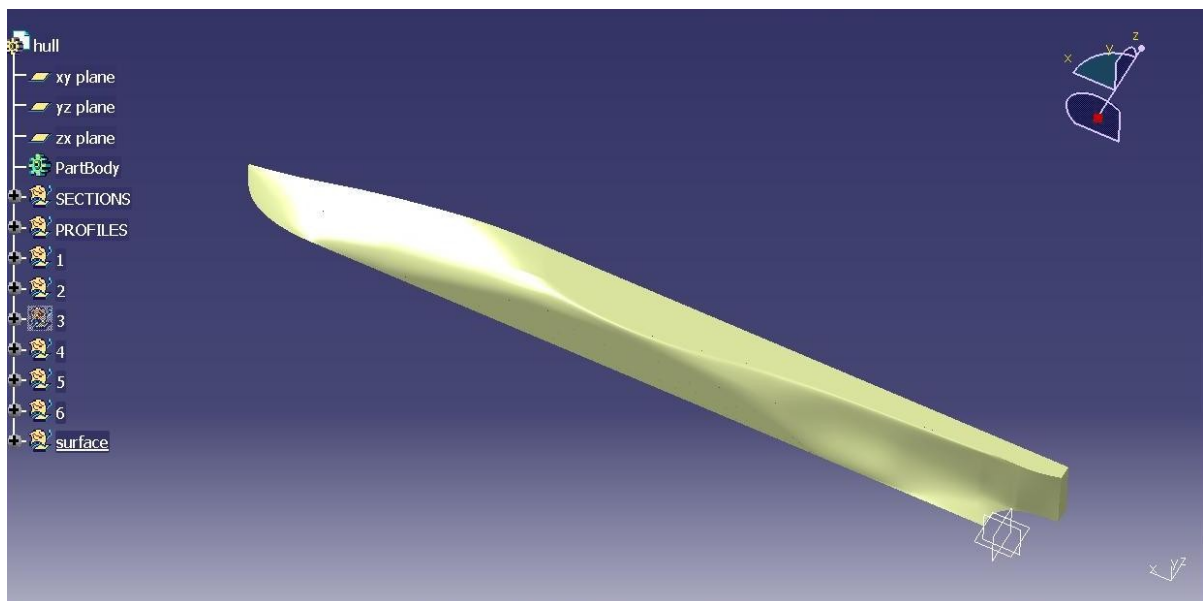


Εικόνα 72Γ: Τρισδιάστατο μοντέλο κατηγορίας 2 (τρισδιάστατη όψη 2)

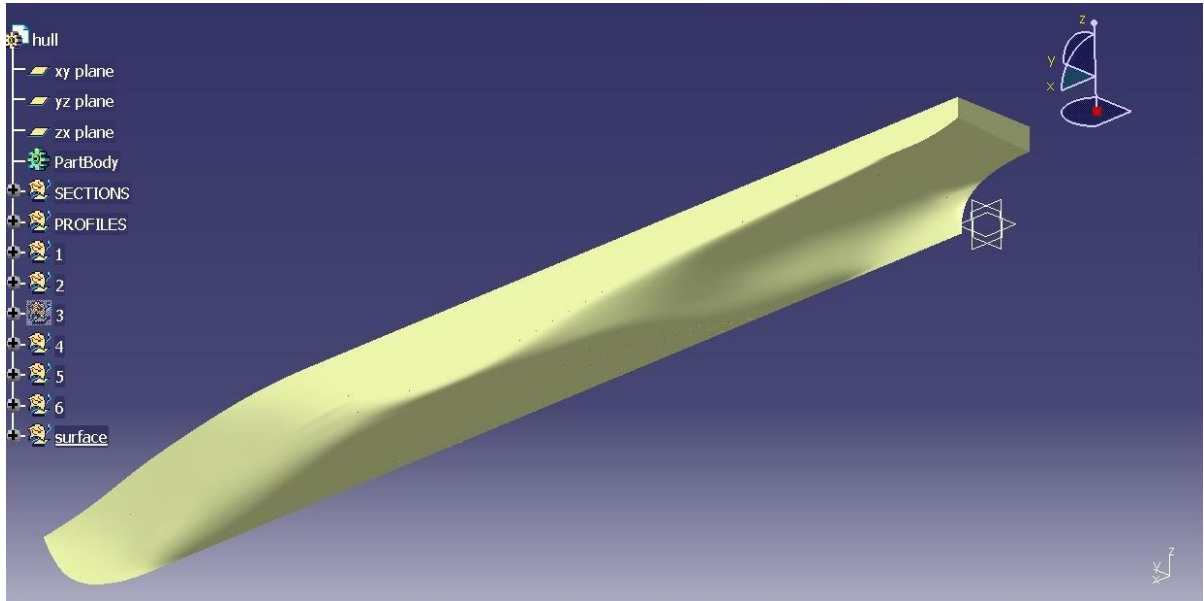
7.3.Κατηγορία 3: (πλώρη χωρίς βολβό , πρύμνη άβακας D)



Εικόνα 73Α: Τρισδιάστατο μοντέλο κατηγορίας 3 (πλάγια όψη)

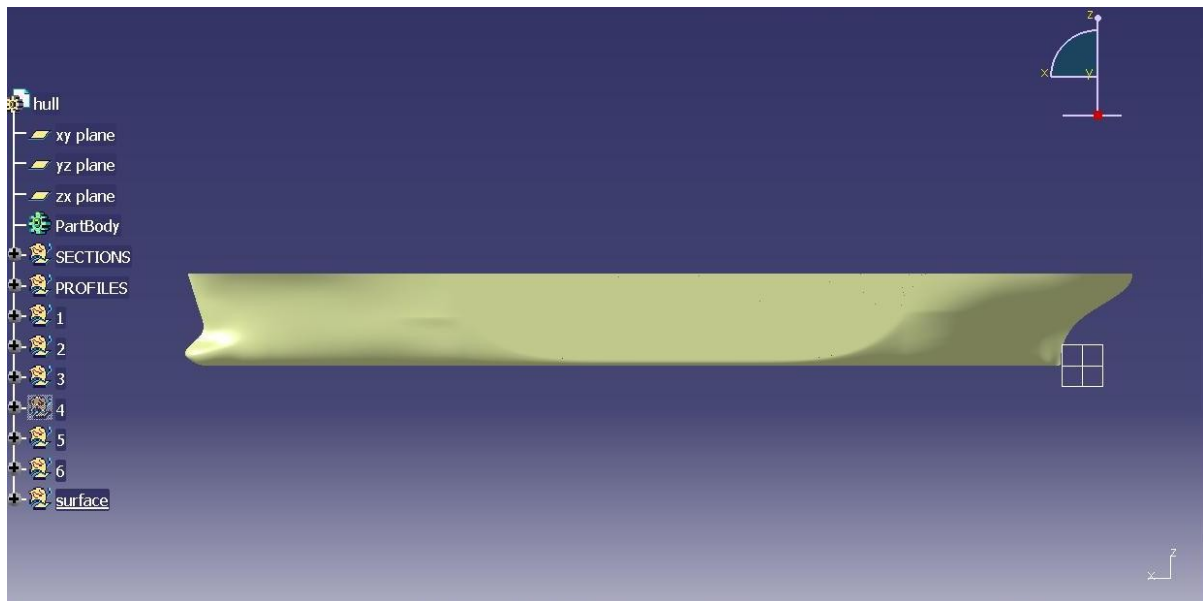


Εικόνα 73Β: Τρισδιάστατο μοντέλο κατηγορίας 3 (τριδιάστατη όψη 1)

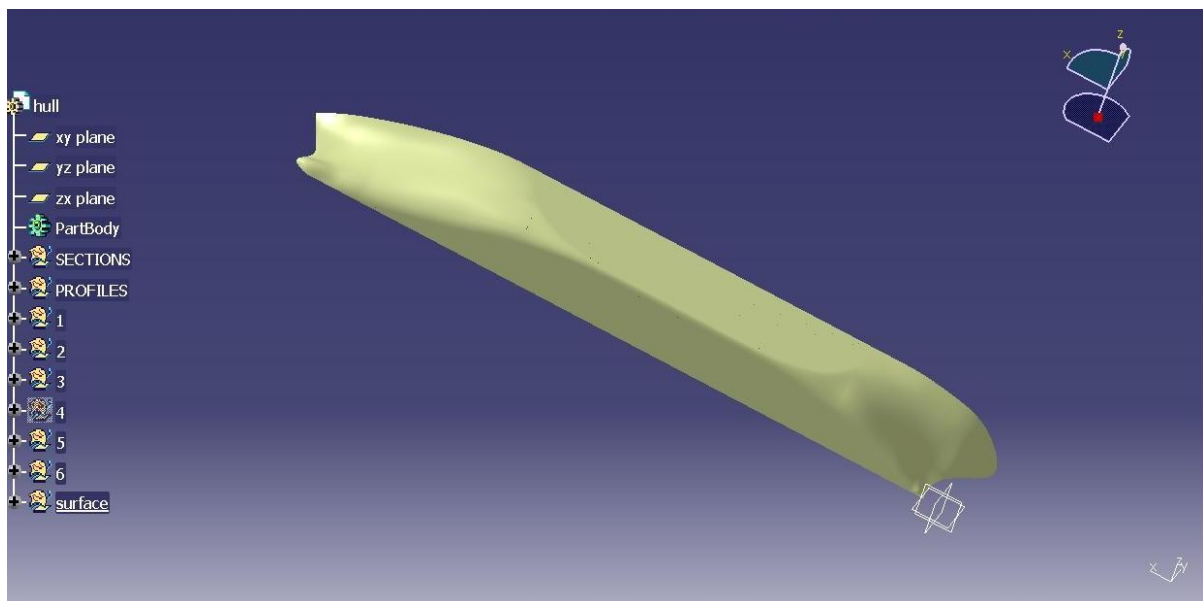


Εικόνα 73Γ: Τρισδιάστατο μοντέλο κατηγορίας 3 (τρειςδιάστατη όψη 2)

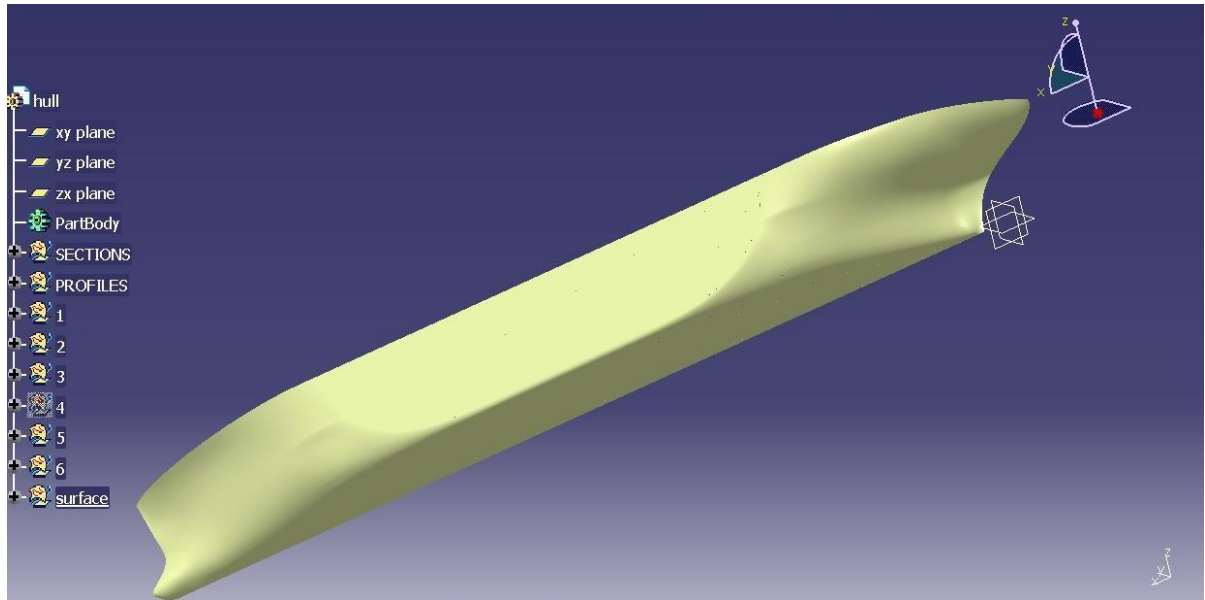
7.4.Κατηγορία 4: (βολβοειδής πλώρη , πρύμνη καταδρομικού)



Εικόνα 74A: Τρισδιάστατο μοντέλο κατηγορίας 4 (πλάγια όψη)

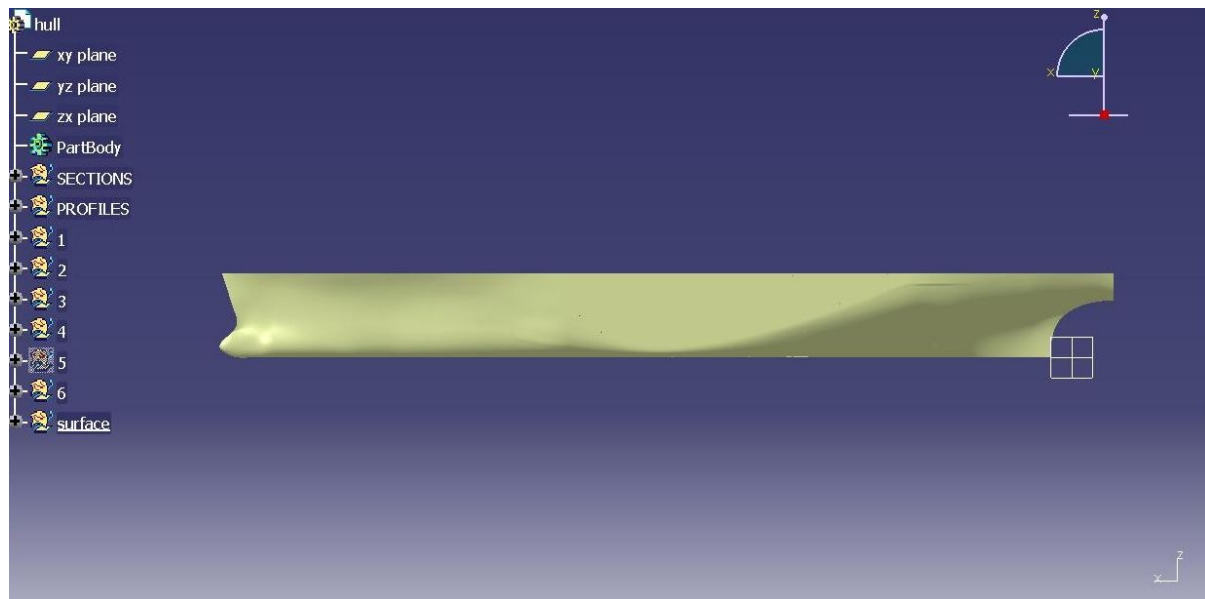


Εικόνα 74B: Τρισδιάστατο μοντέλο κατηγορίας 4 (τρισδιάστατη όψη 1)

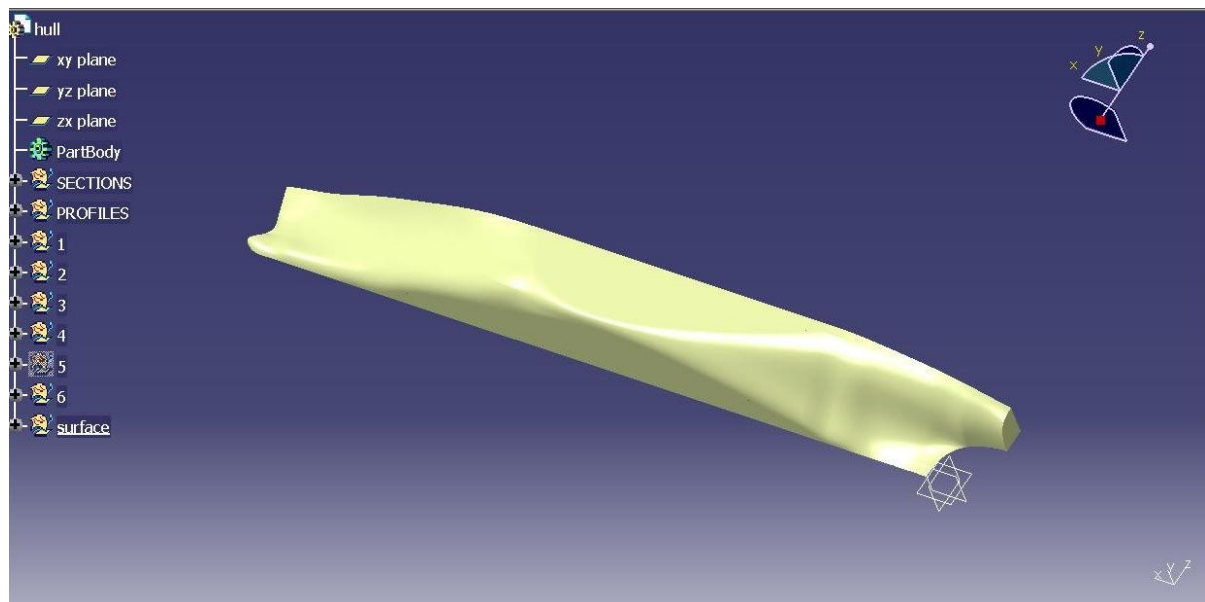


Εικόνα 74Γ: Τρισδιάστατο μοντέλο κατηγορίας 4 (τρισδιάστατη όψη 2)

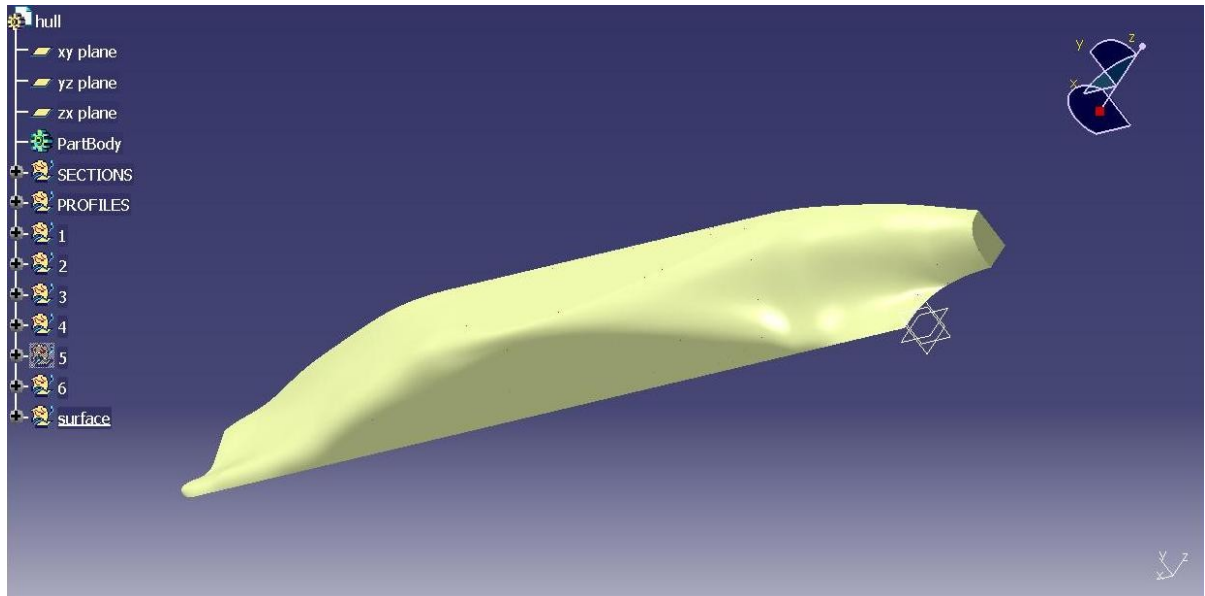
7.5.Κατηγορία 5: (βολβοειδής πλώρη , πρύμνη άβακας A-B-C)



Εικόνα 75A: Τρισδιάστατο μοντέλο κατηγορίας 5 (πλάγια όψη)



Εικόνα 75B: Τρισδιάστατο μοντέλο κατηγορίας 5 (τρειςδιάστατη όψη 1)

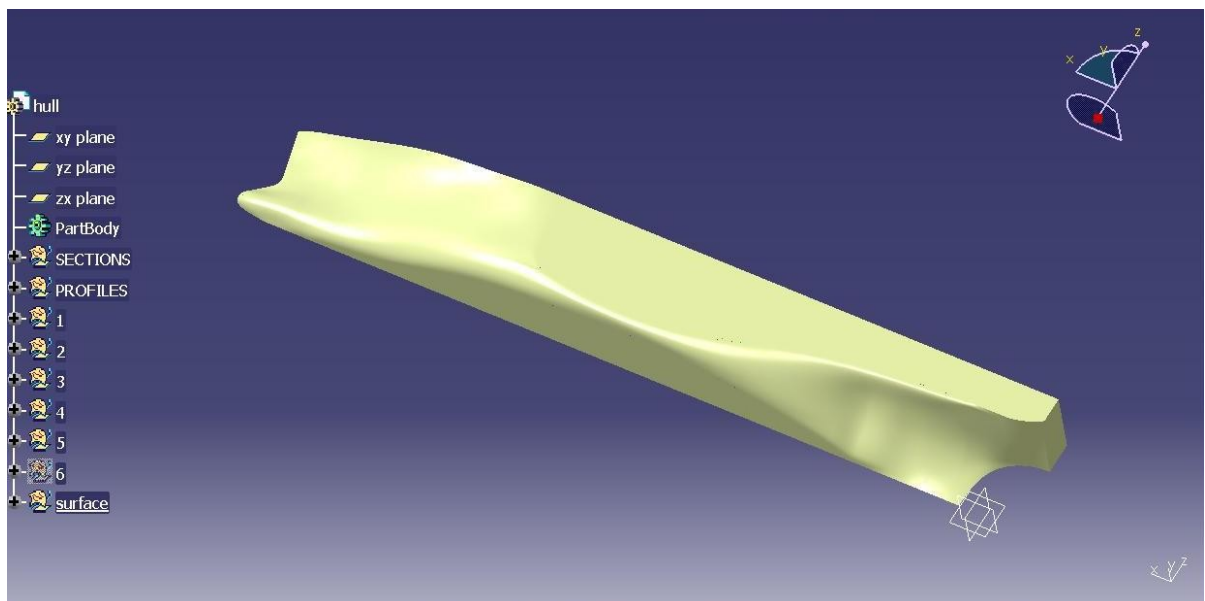


Εικόνα 75Γ: Τρισδιάστατο μοντέλο κατηγορίας 5 (τρειςδιάστατη όψη 2)

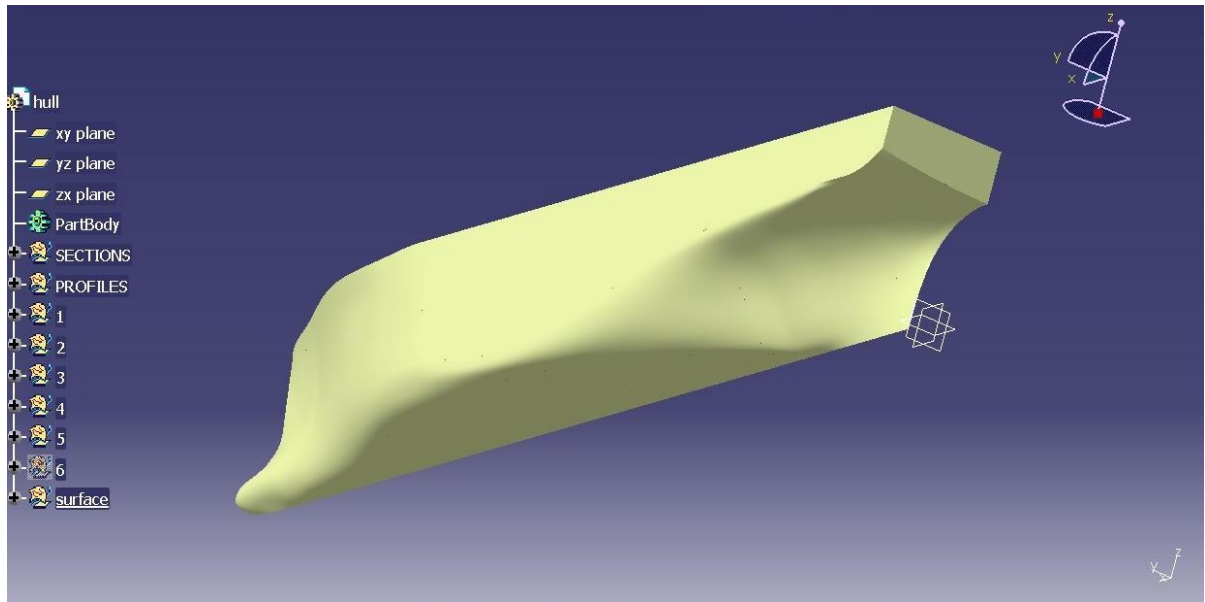
7.6.Κατηγορία 6: (βολβοειδής πλώρη , πρύμνη άβακας D)



Εικόνα 76Α: Τρισδιάστατο μοντέλο κατηγορίας 6 (πλάγια όψη)



Εικόνα 76Β: Τρισδιάστατο μοντέλο κατηγορίας 6 (τρειςδιάστατη όψη 1)

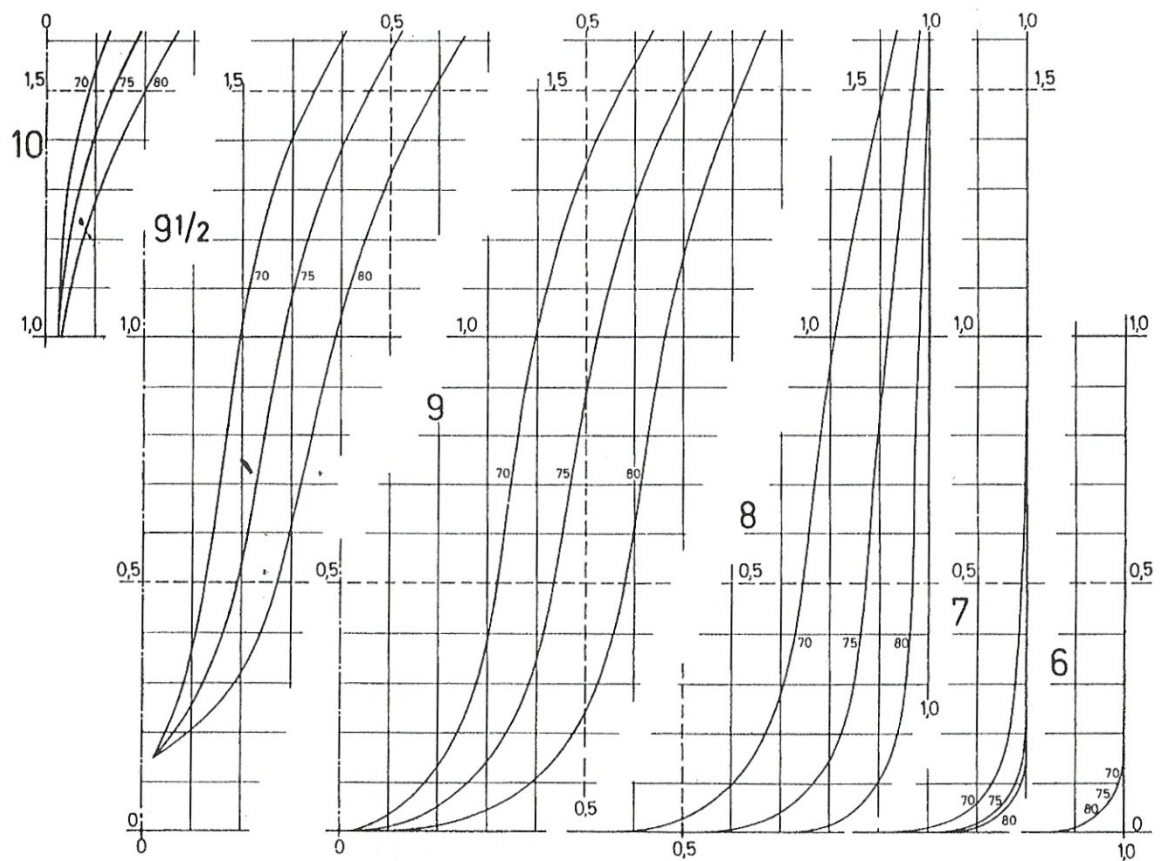


Εικόνα 76Γ: Τρισδιάστατο μοντέλο κατηγορίας 6 (τρειςδιάστατη όψη 2)

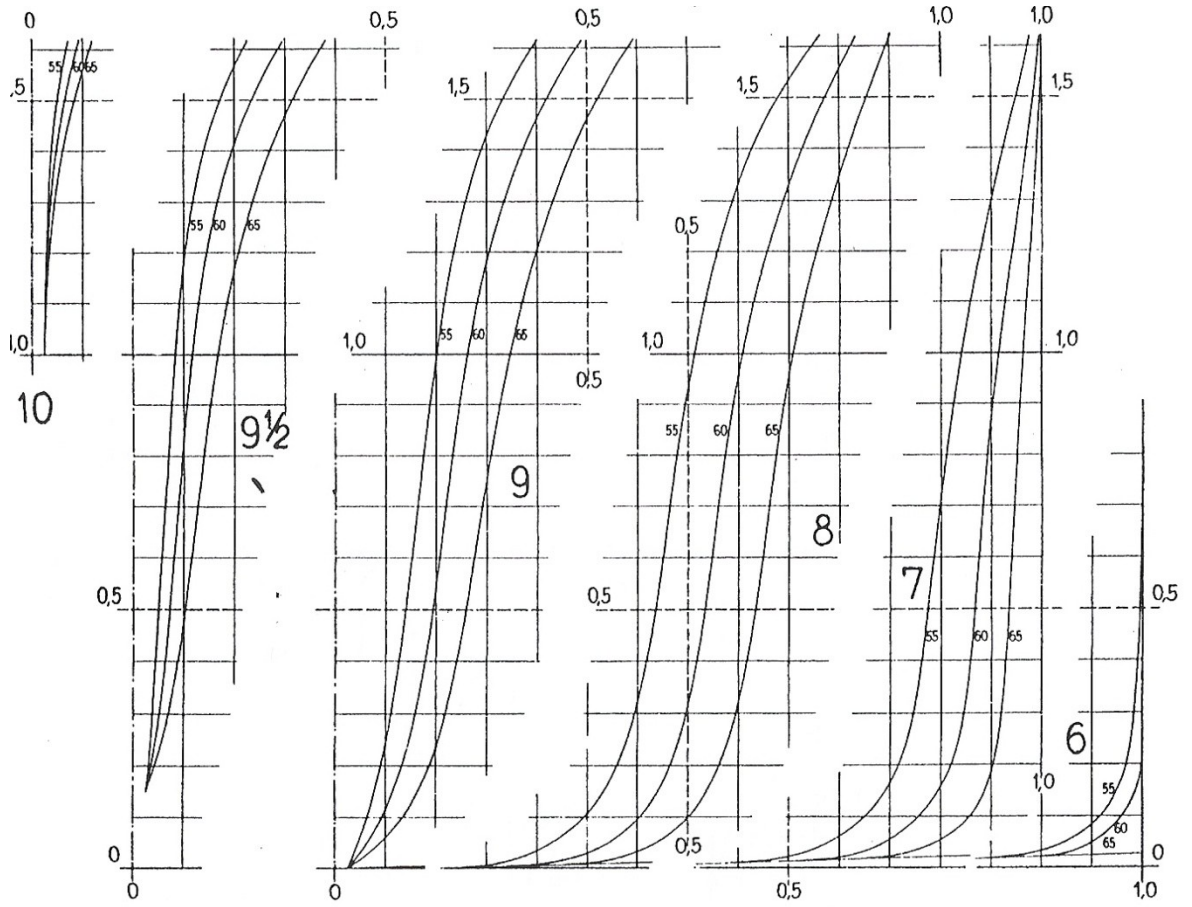
8.ΠΑΡΑΡΤΗΜΑΤΑ

8.1.Παράρτημα 1: «Η Συστηματική Σειρά Formdata»

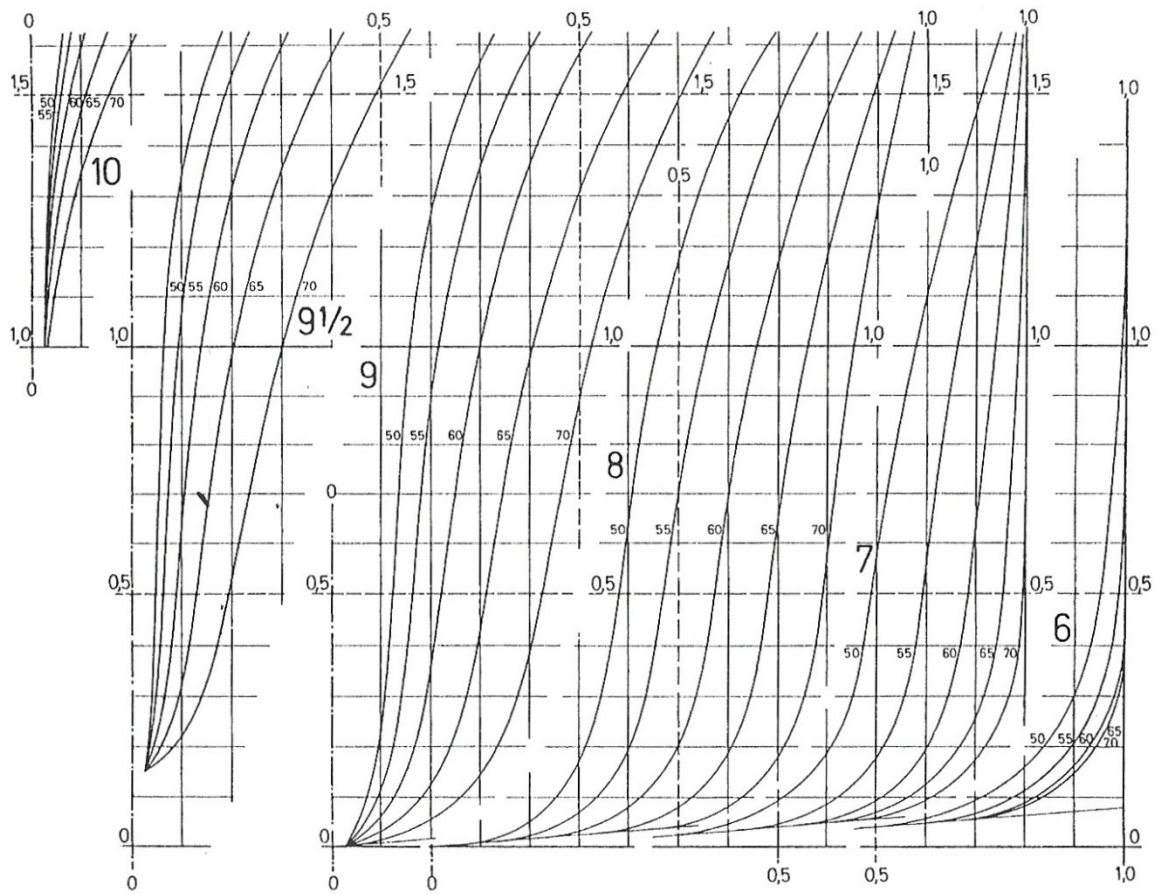
ΟΙΚΟΓΕΝΕΙΑ U1F



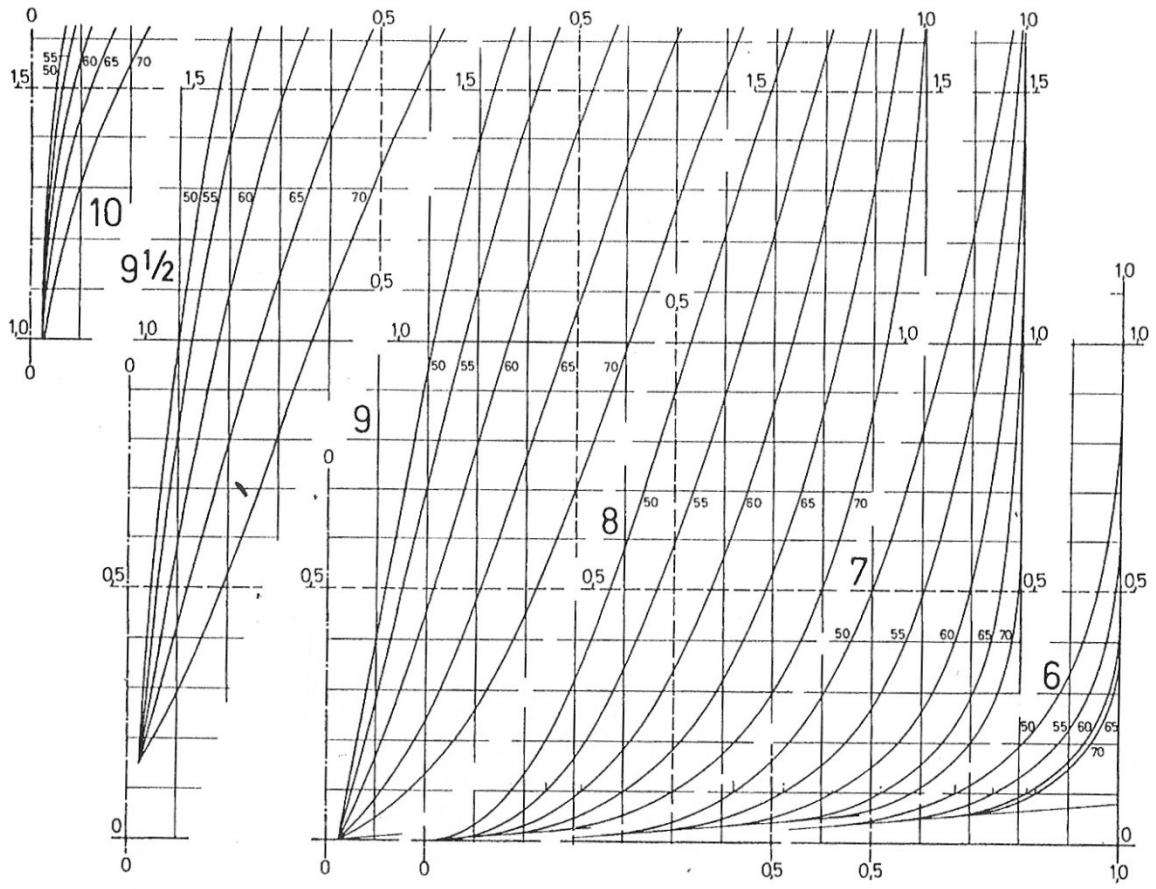
OIKOGENEIA U2F



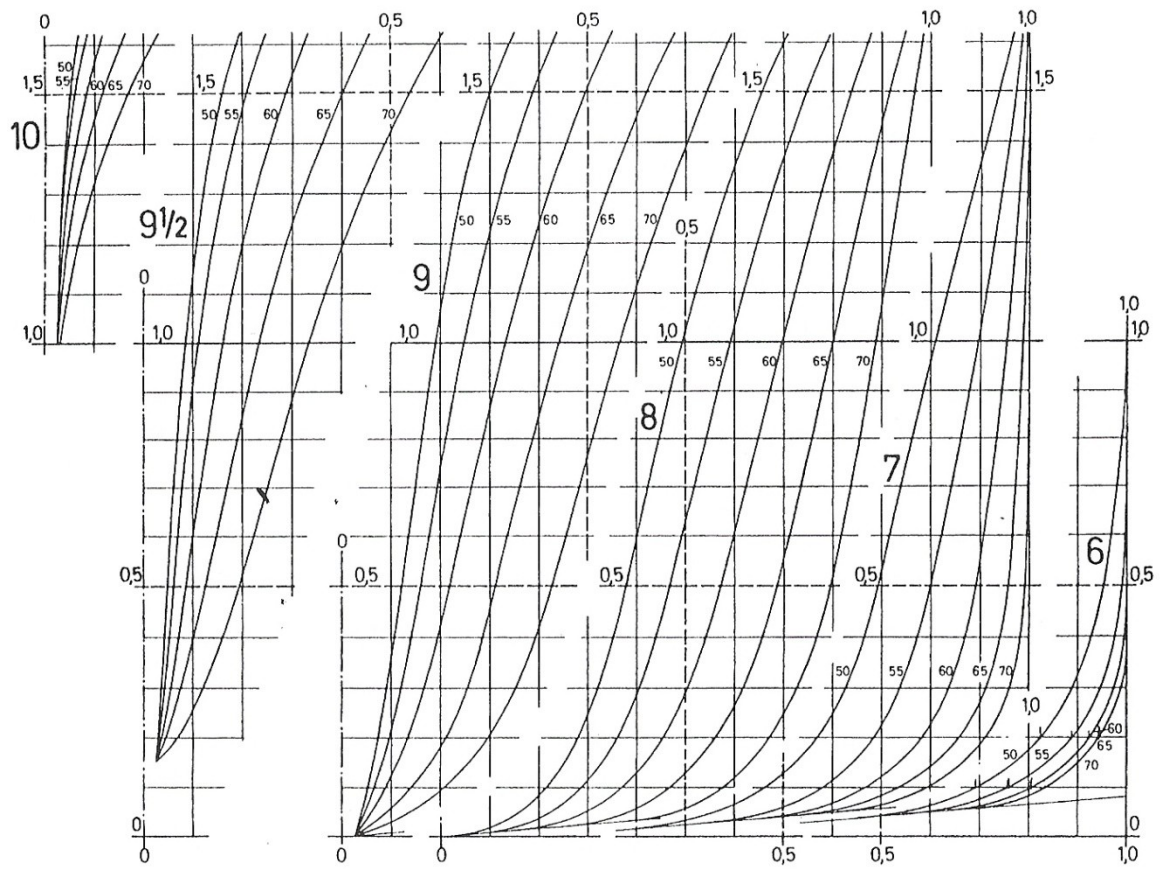
ΟΙΚΟΓΕΝΕΙΑ U3F



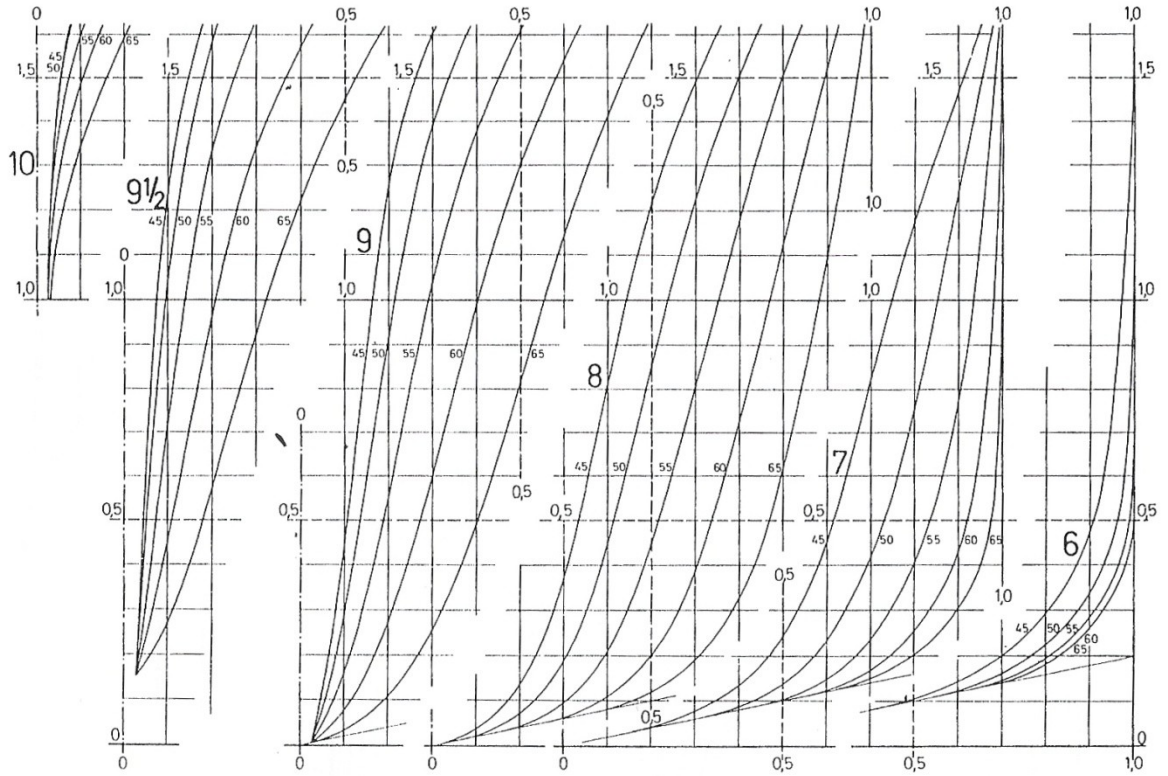
ΟΙΚΟΓΕΝΕΙΑ V3F



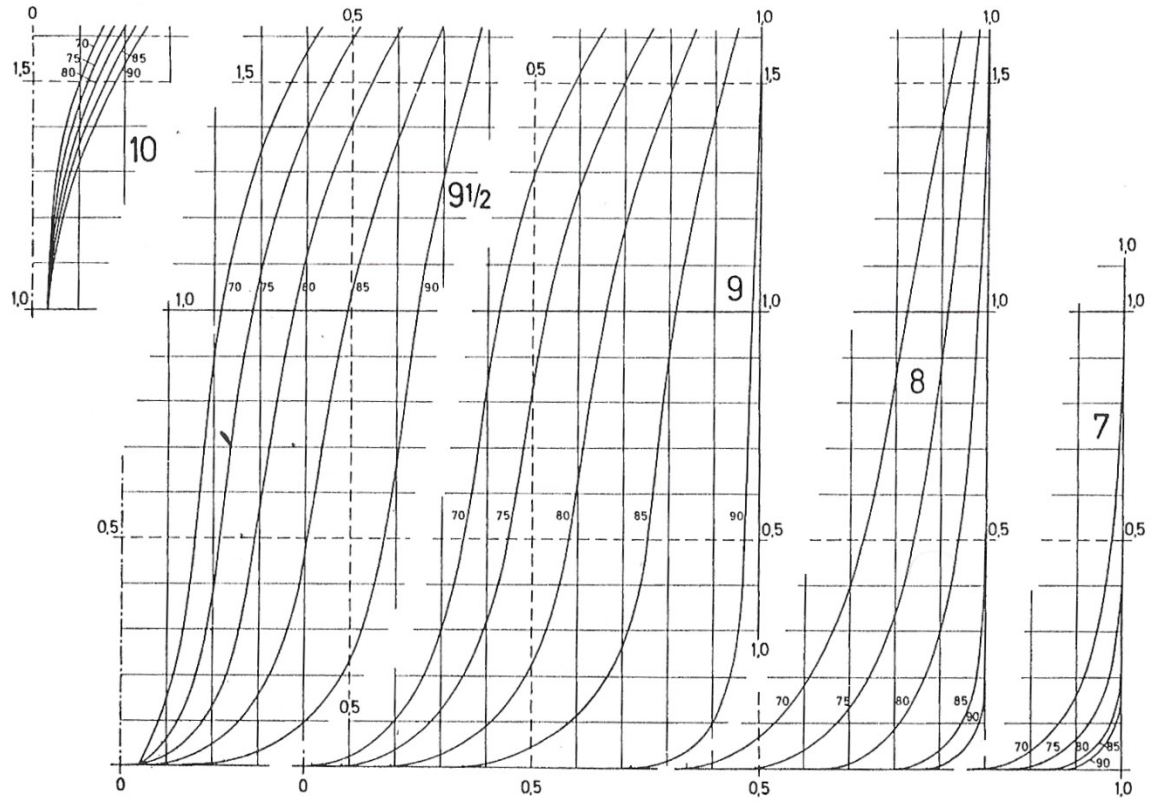
ΟΙΚΟΓΕΝΕΙΑ N3F



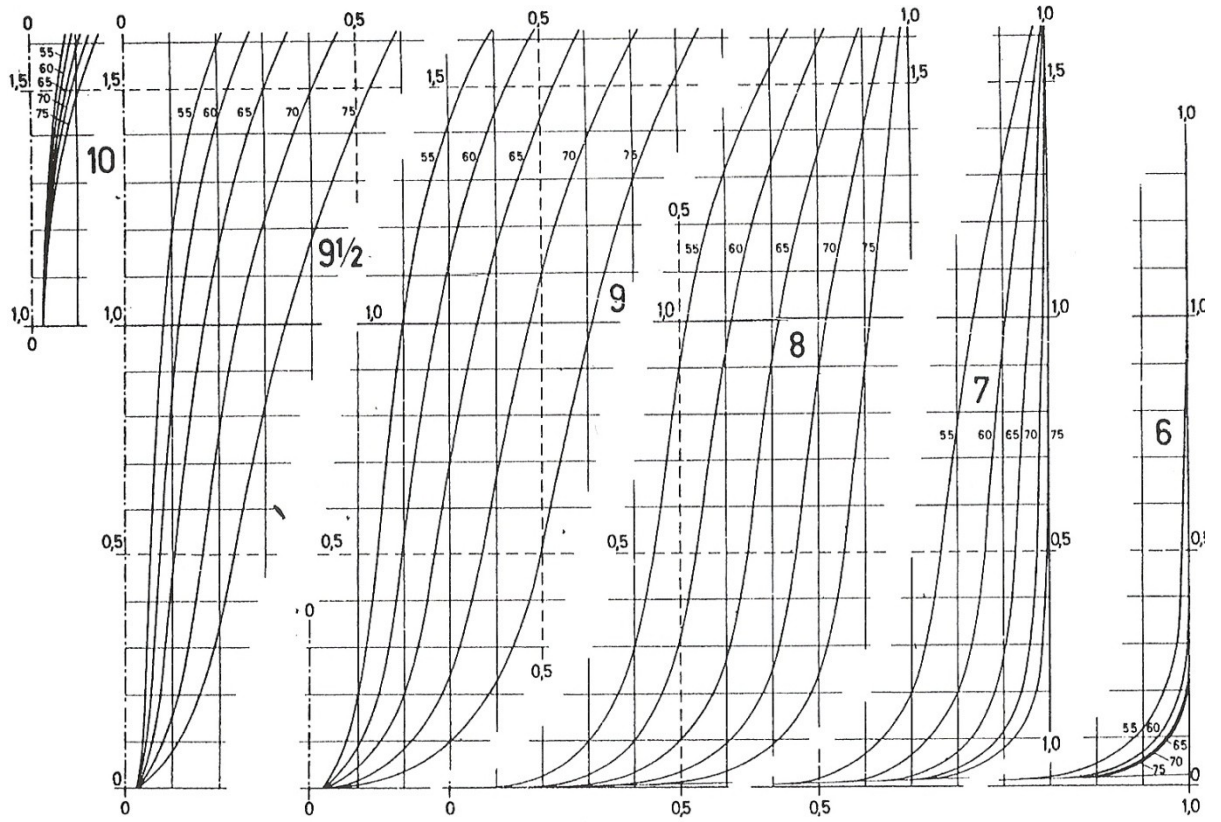
ΟΙΚΟΓΕΝΕΙΑ Ν4F



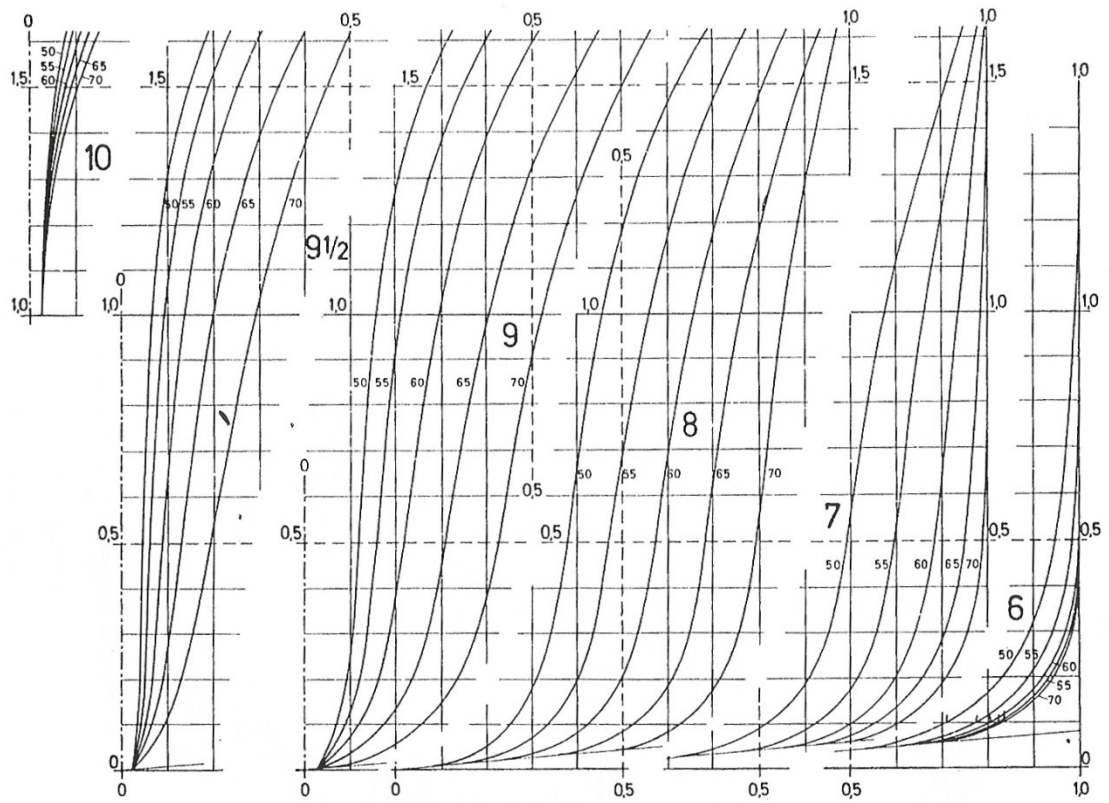
ΟΙΚΟΓΕΝΕΙΑ B₀1F



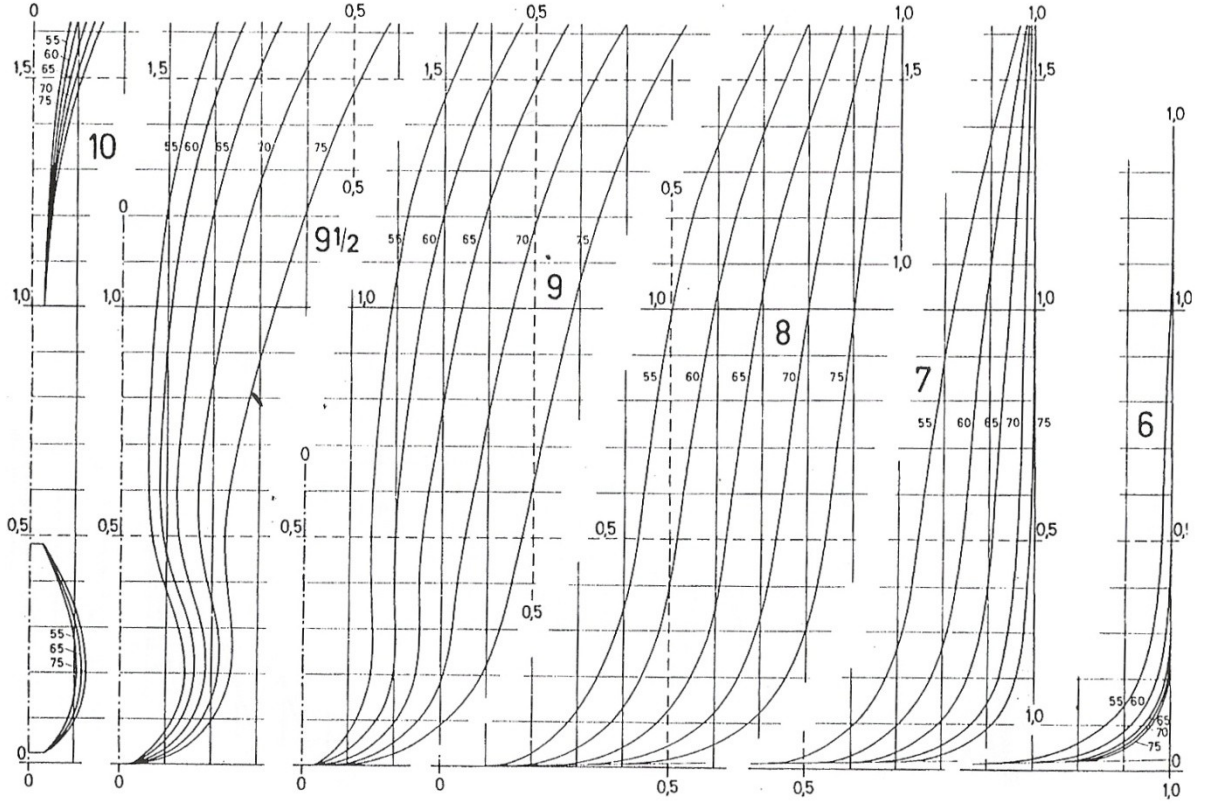
ΟΙΚΟΓΕΝΕΙΑ B₀2F



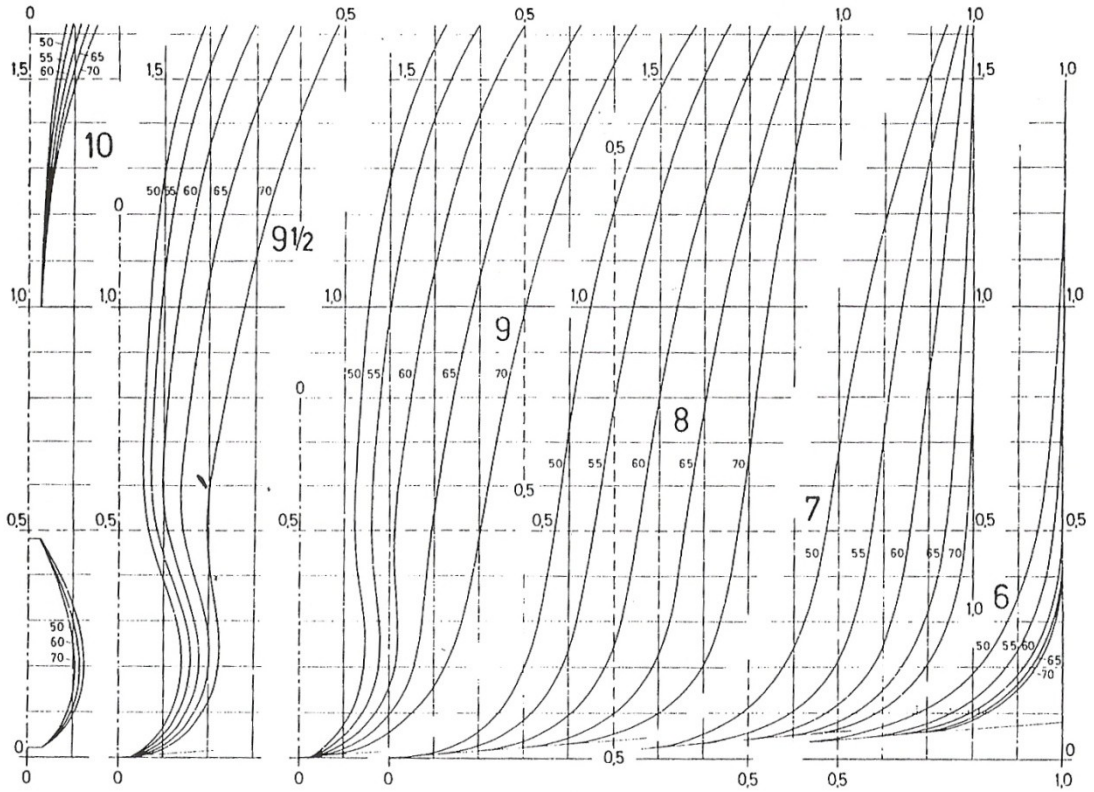
ΟΙΚΟΓΕΝΕΙΑ B₀3F



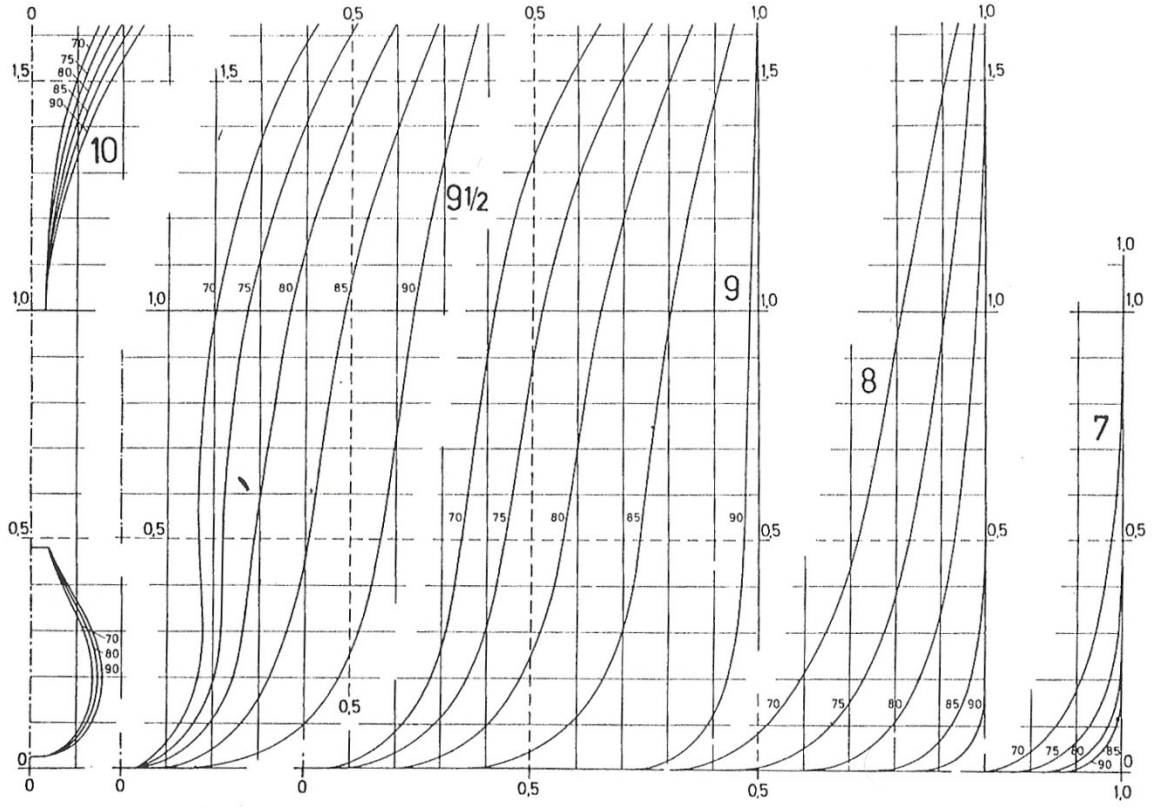
ΟΙΚΟΓΕΝΕΙΑ B₄2F



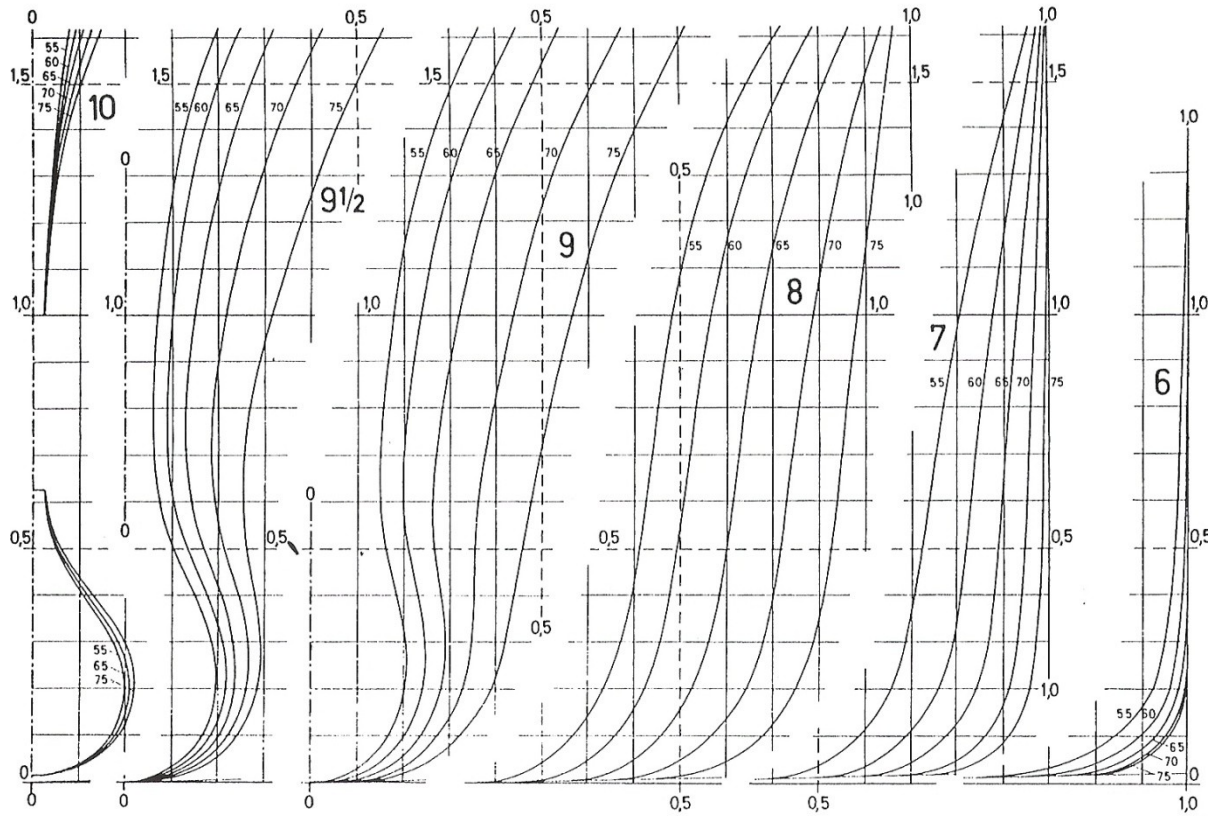
ΟΙΚΟΓΕΝΕΙΑ B₄3F



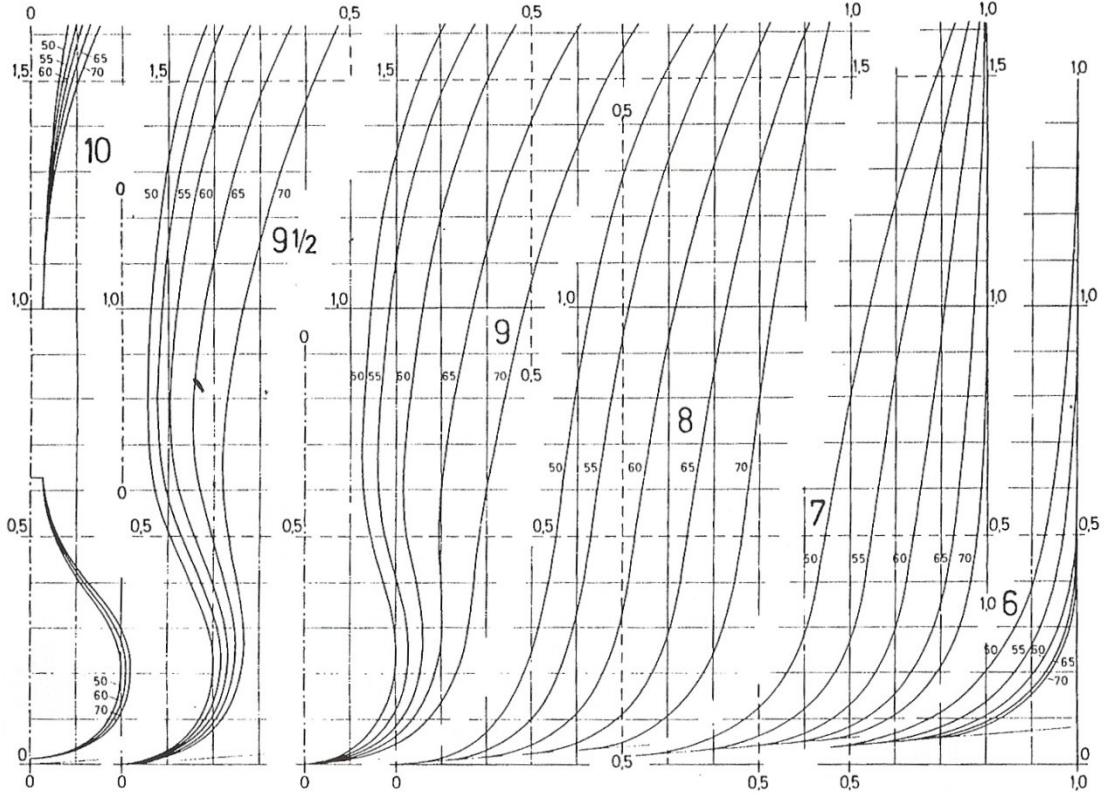
ΟΙΚΟΓΕΝΕΙΑ B₅1F



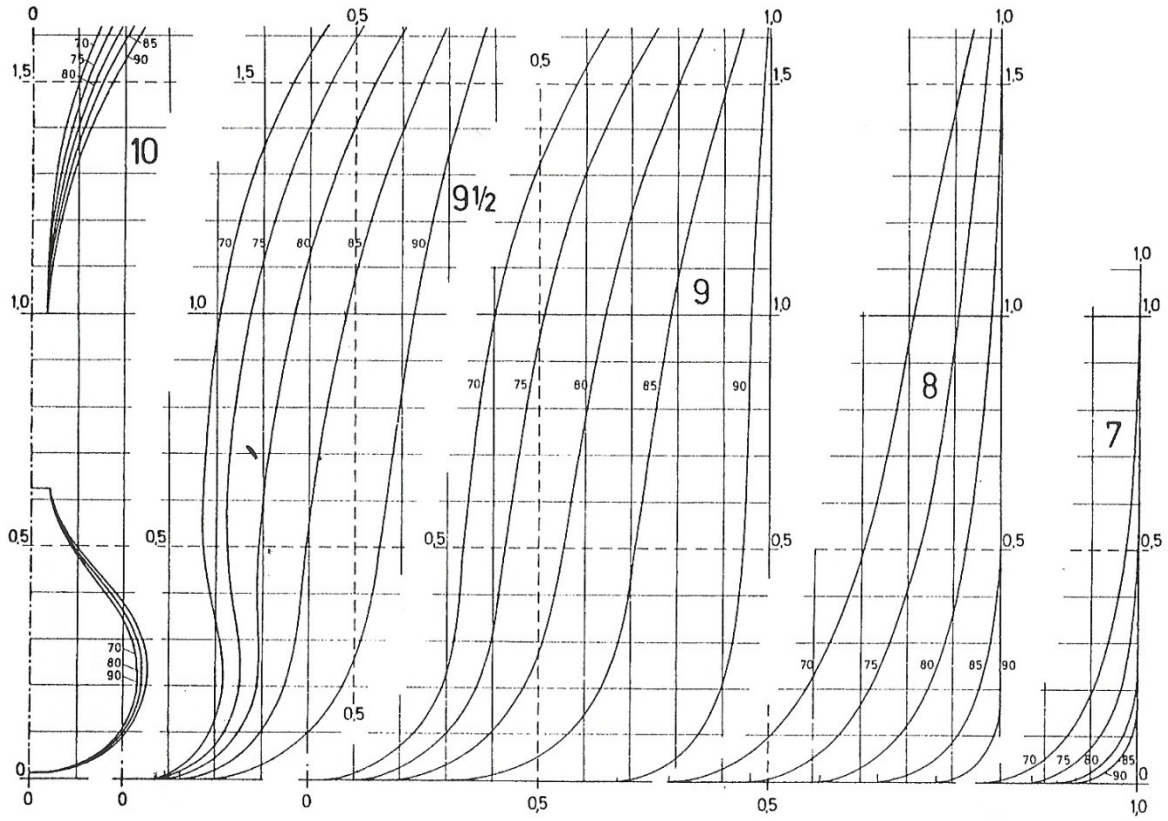
ΟΙΚΟΓΕΝΕΙΑ B₈2F



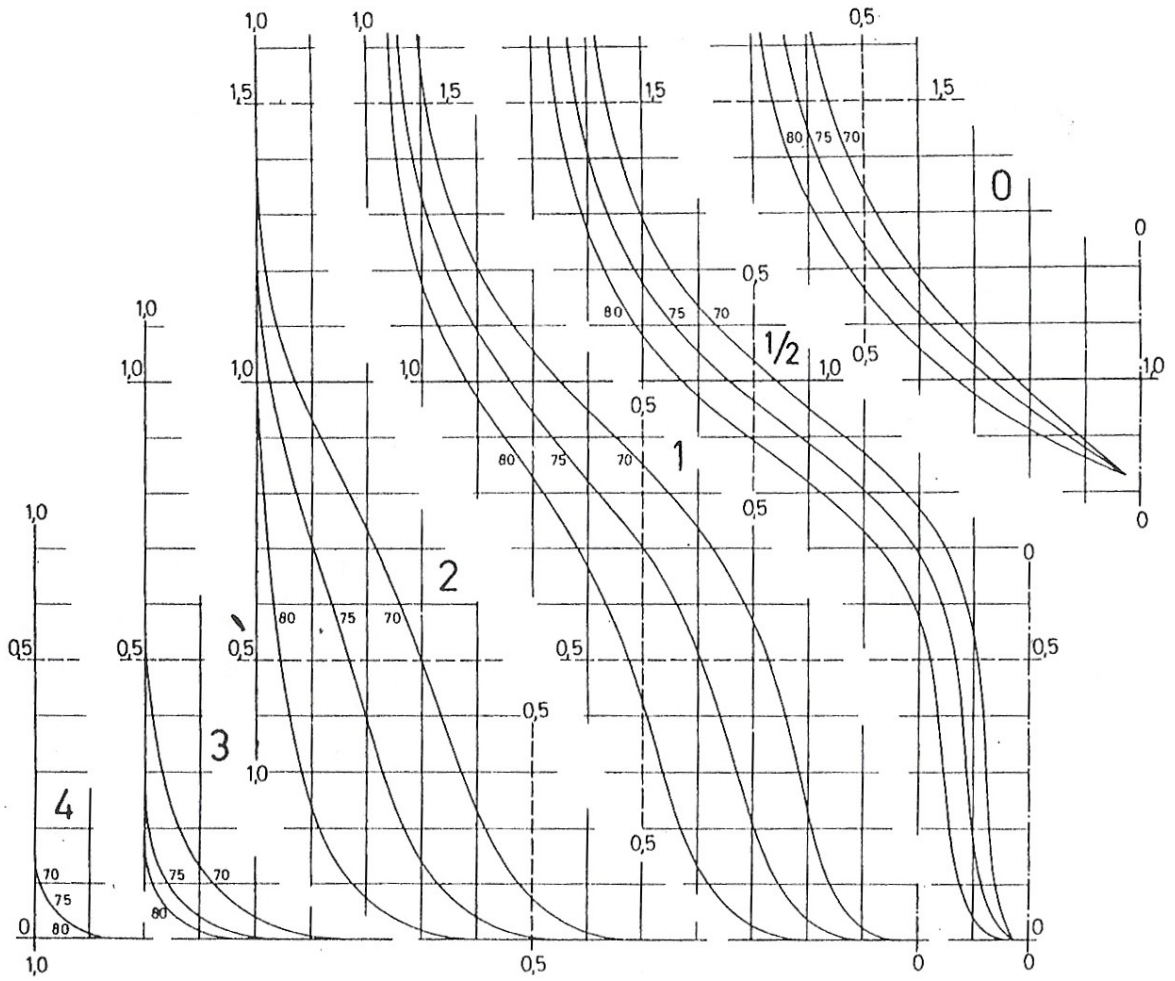
ΟΙΚΟΓΕΝΕΙΑ B₈3F



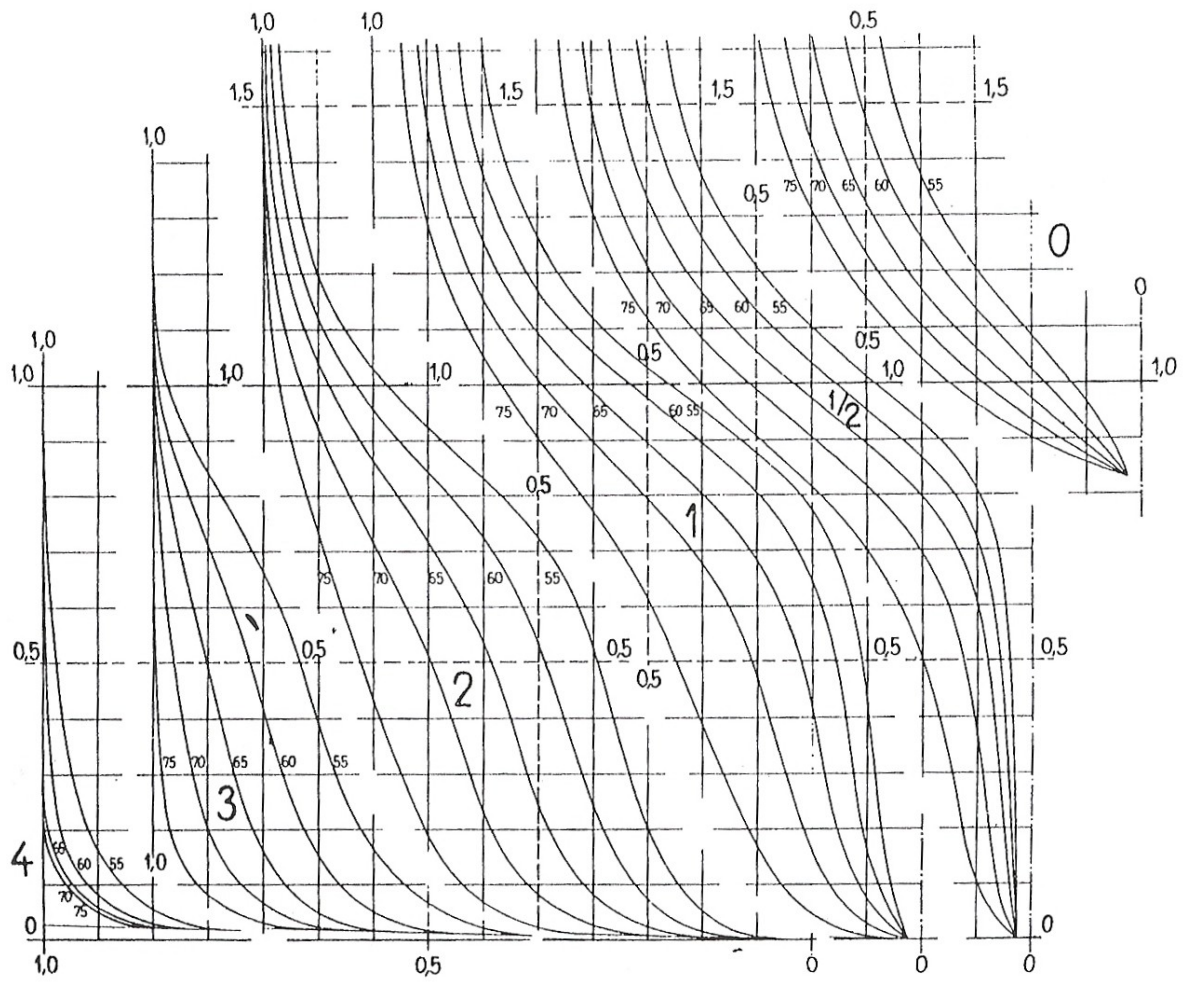
OIKOΓENEIA B₁₀1F



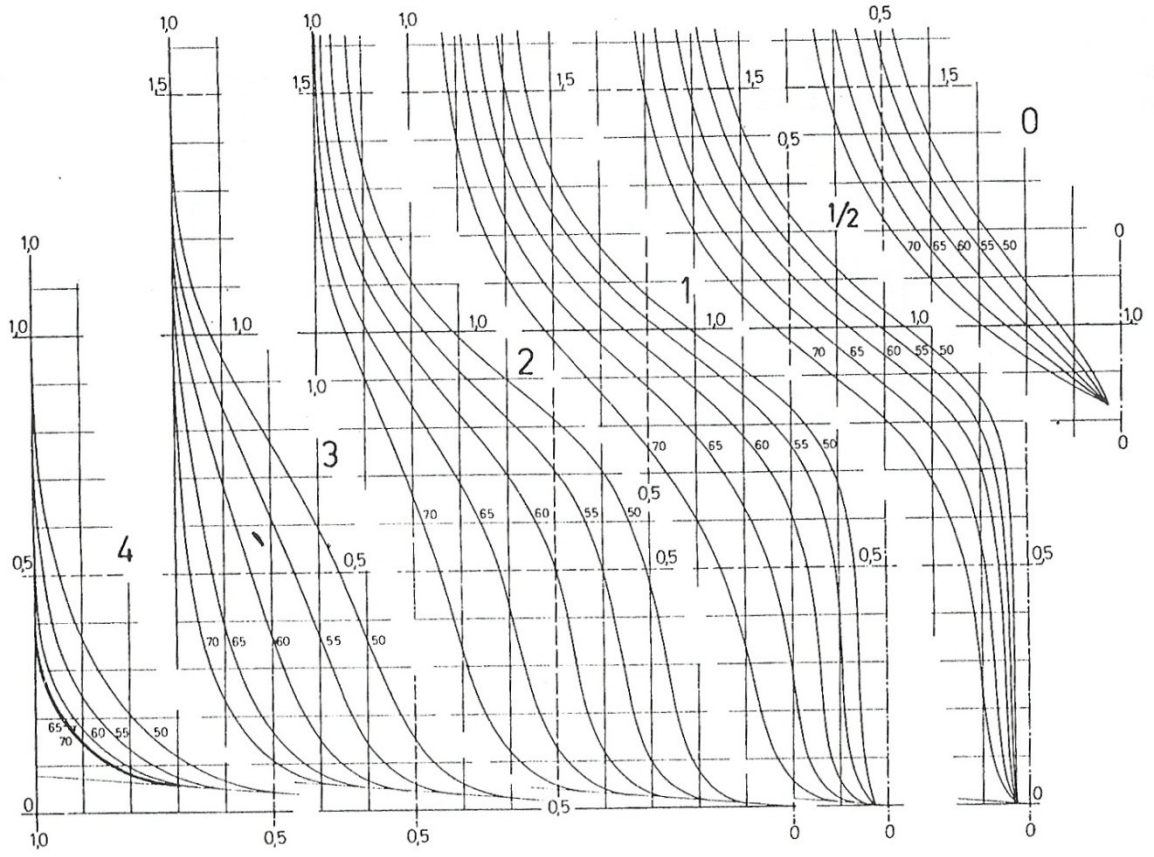
ΟΙΚΟΓΕΝΕΙΑ U1A



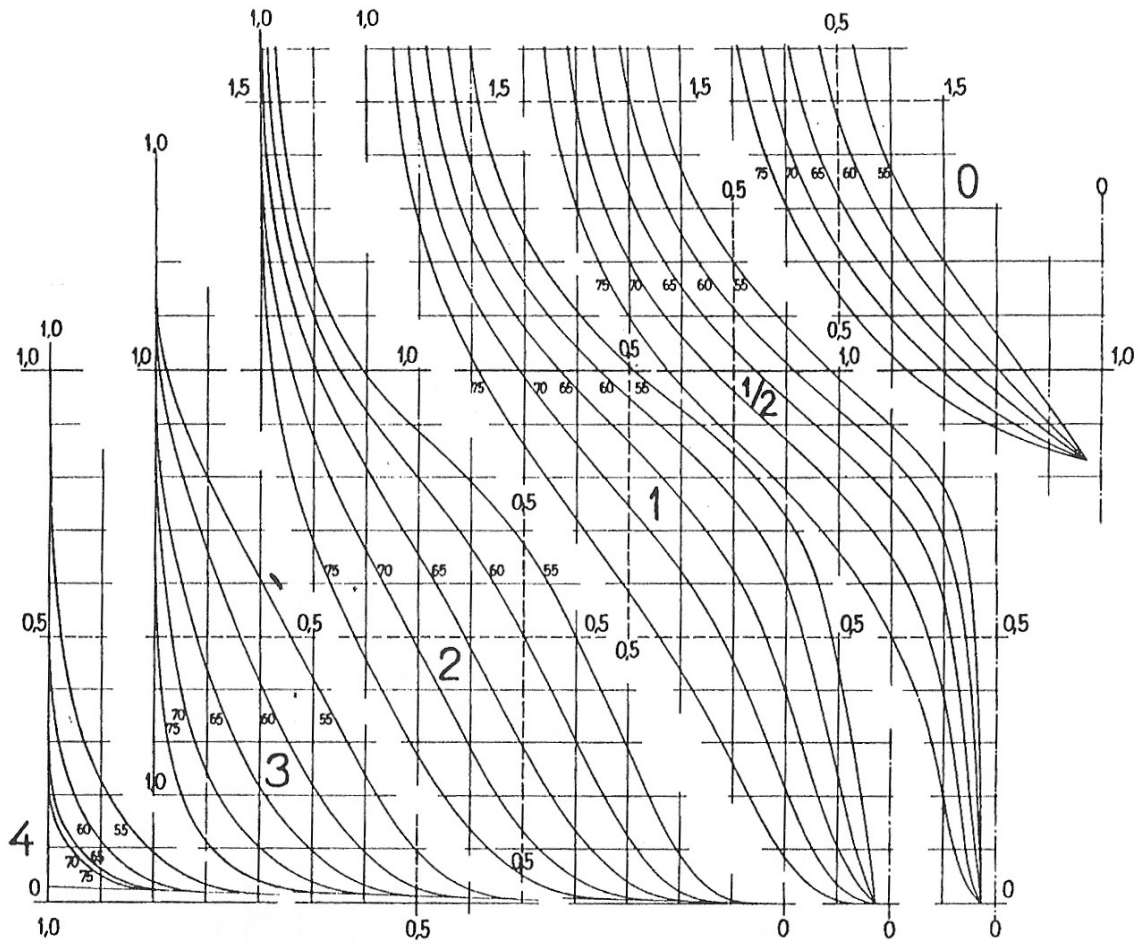
ΟΙΚΟΓΕΝΕΙΑ U2A



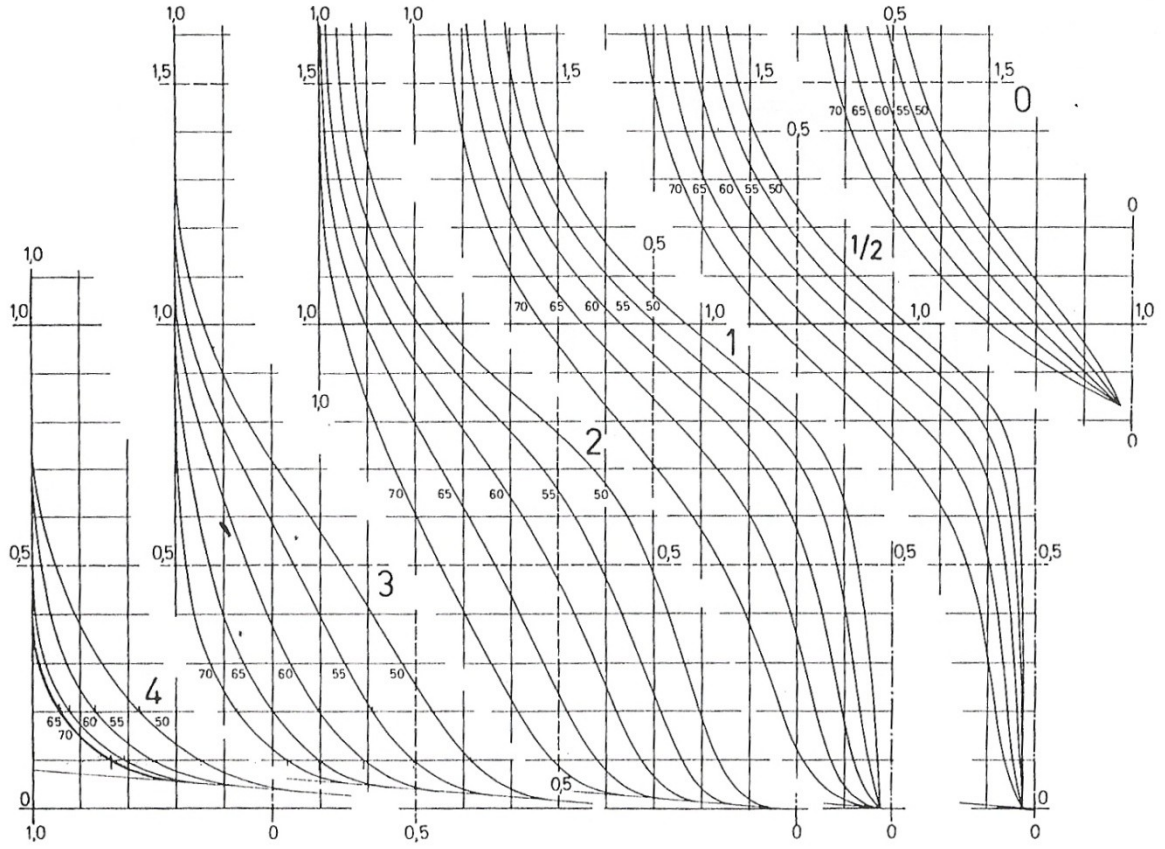
ΟΙΚΟΓΕΝΕΙΑ U3A



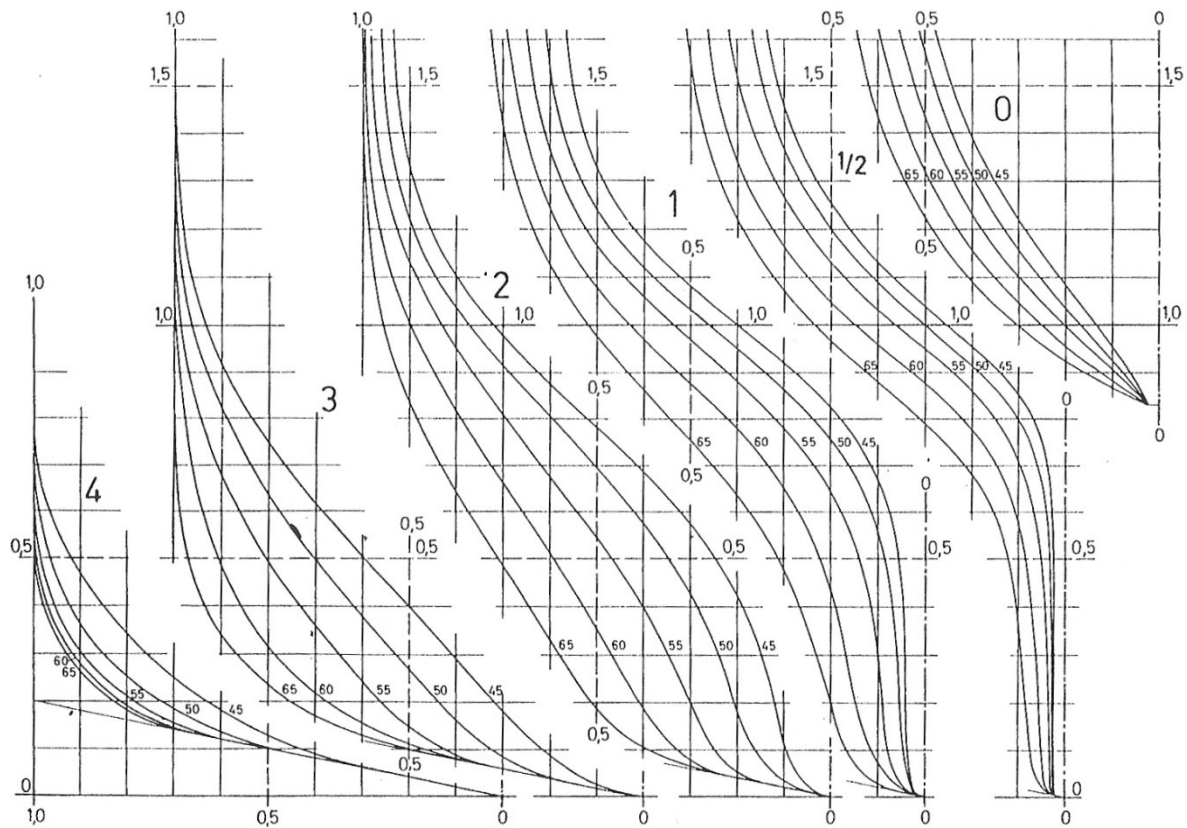
ΟΙΚΟΓΕΝΕΙΑ N2A



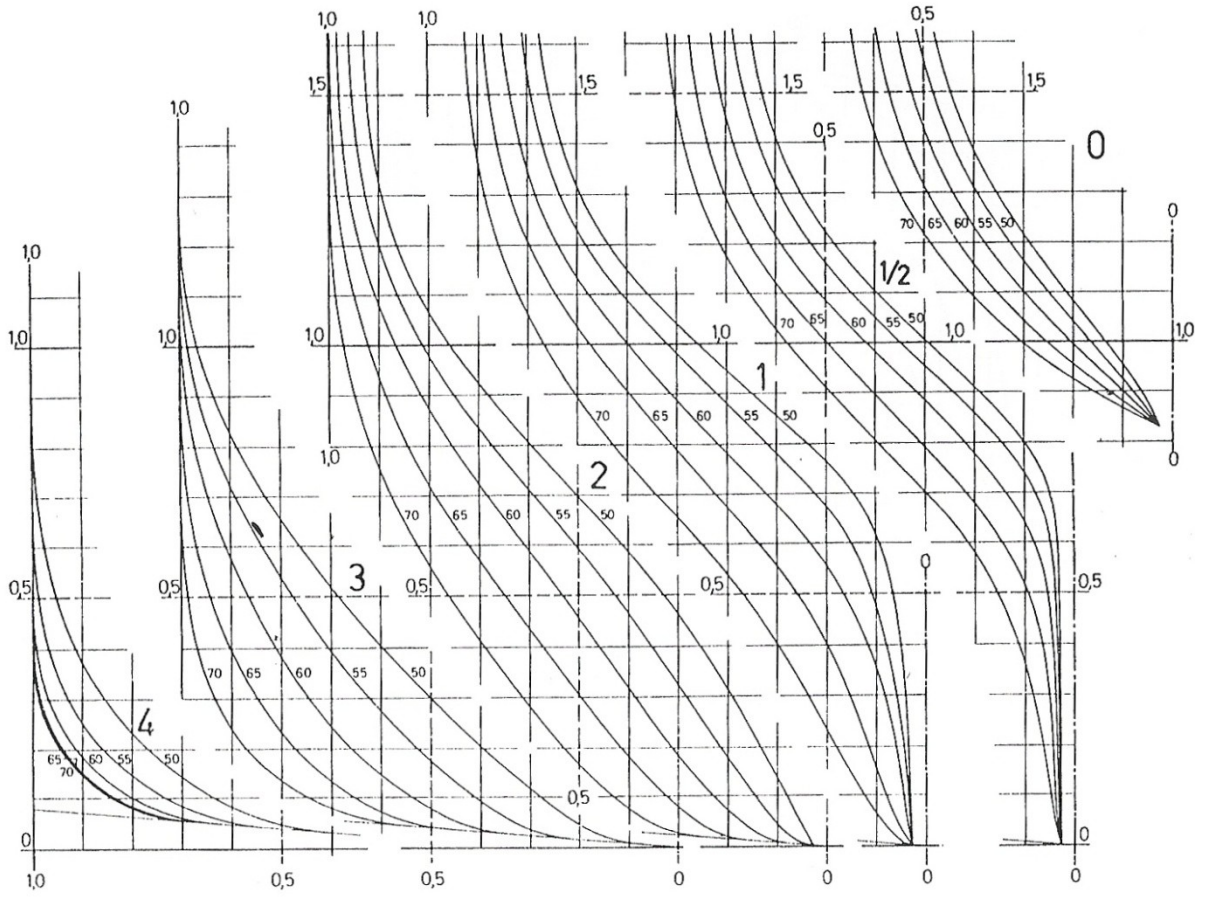
ΟΙΚΟΓΕΝΕΙΑ Ν3Α



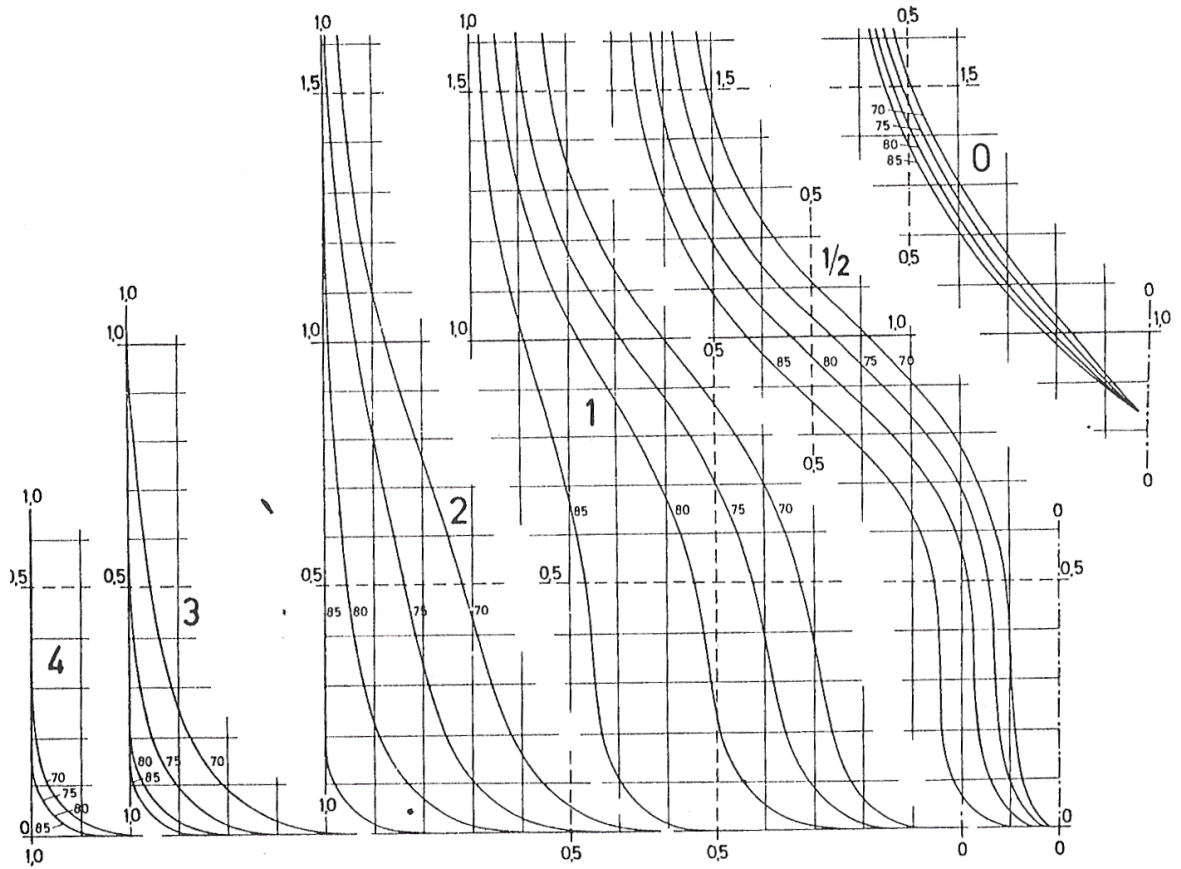
ΟΙΚΟΓΕΝΕΙΑ Ν4Α



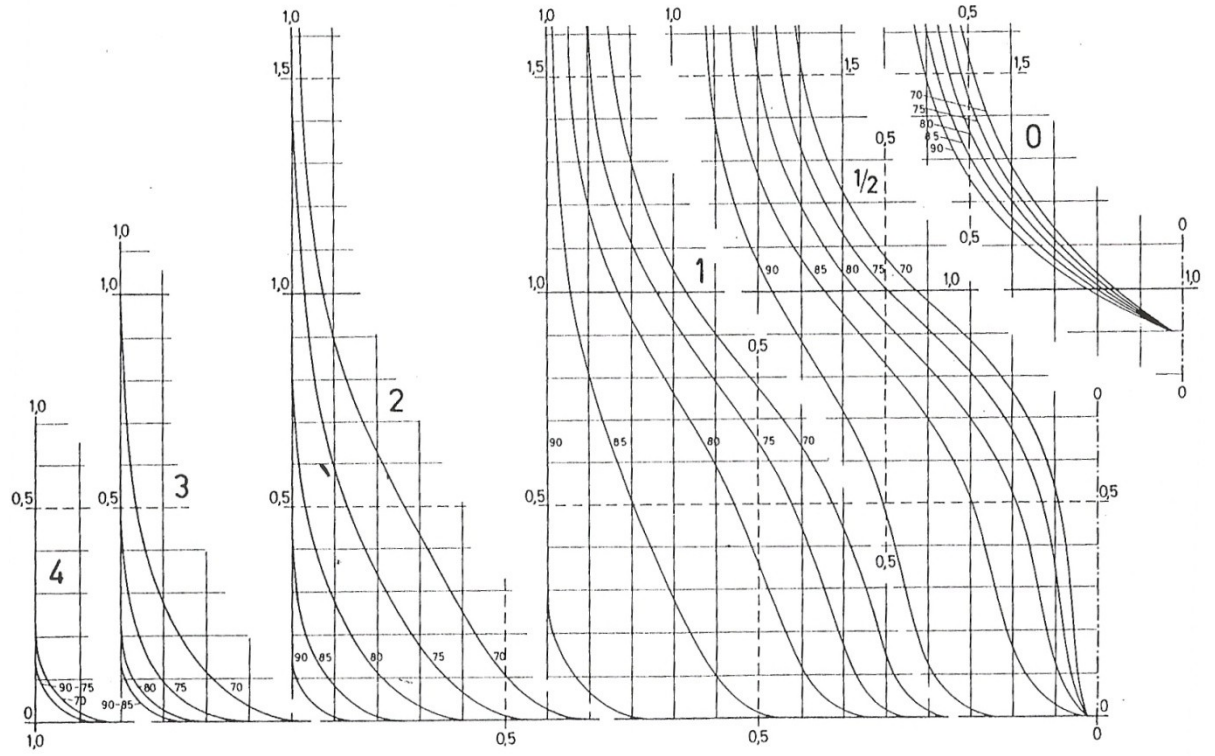
ΟΙΚΟΓΕΝΕΙΑ V3A



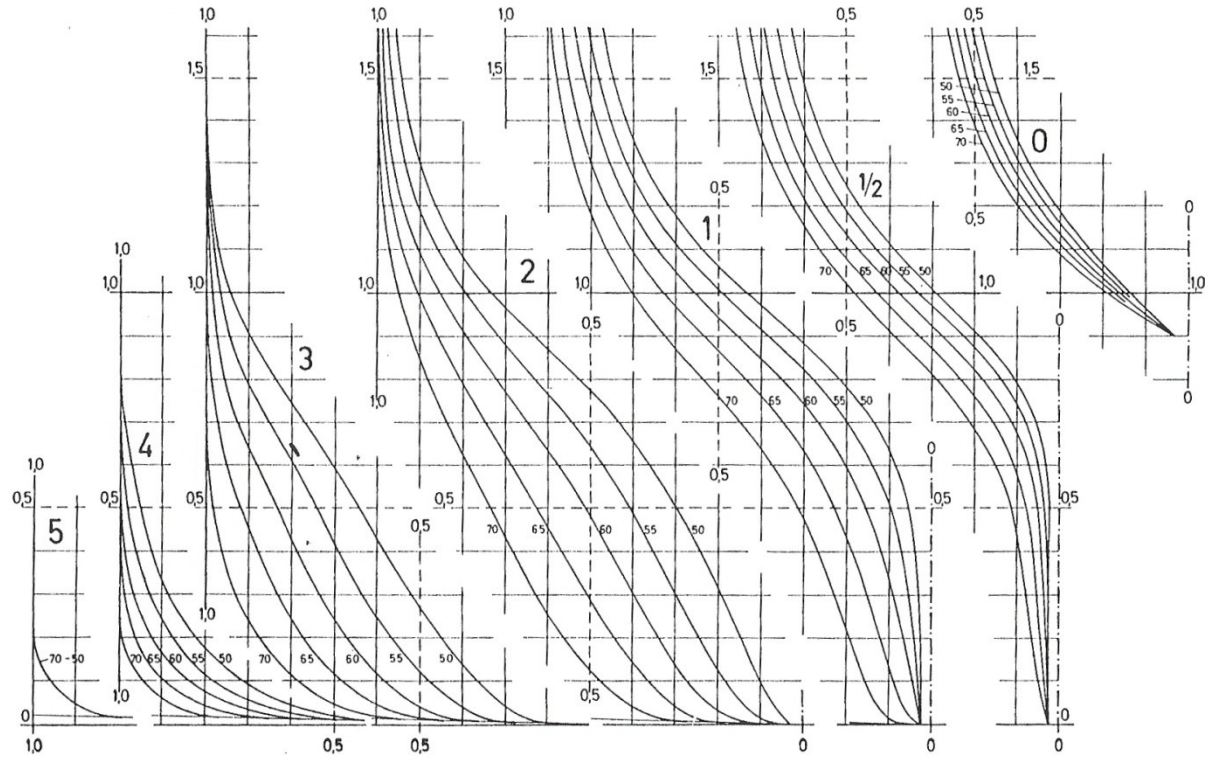
ΟΙΚΟΓΕΝΕΙΑ Τ1Α



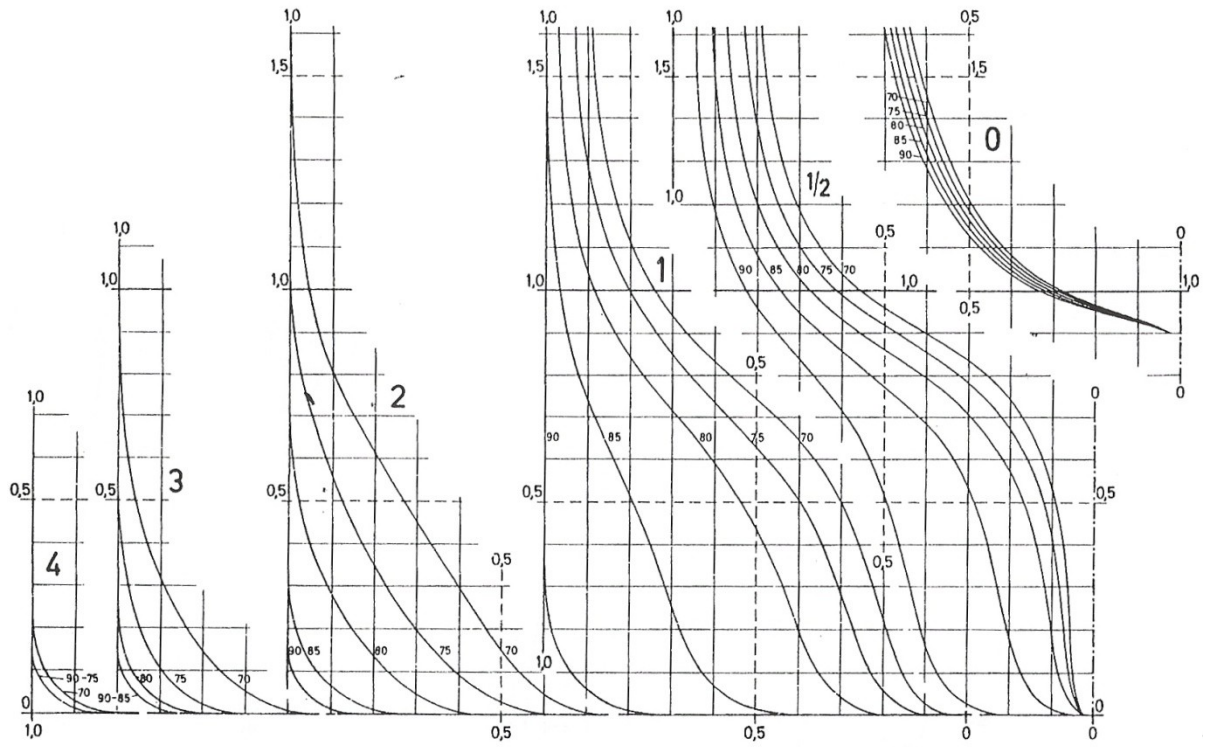
ΟΙΚΟΓΕΝΕΙΑ CA1A



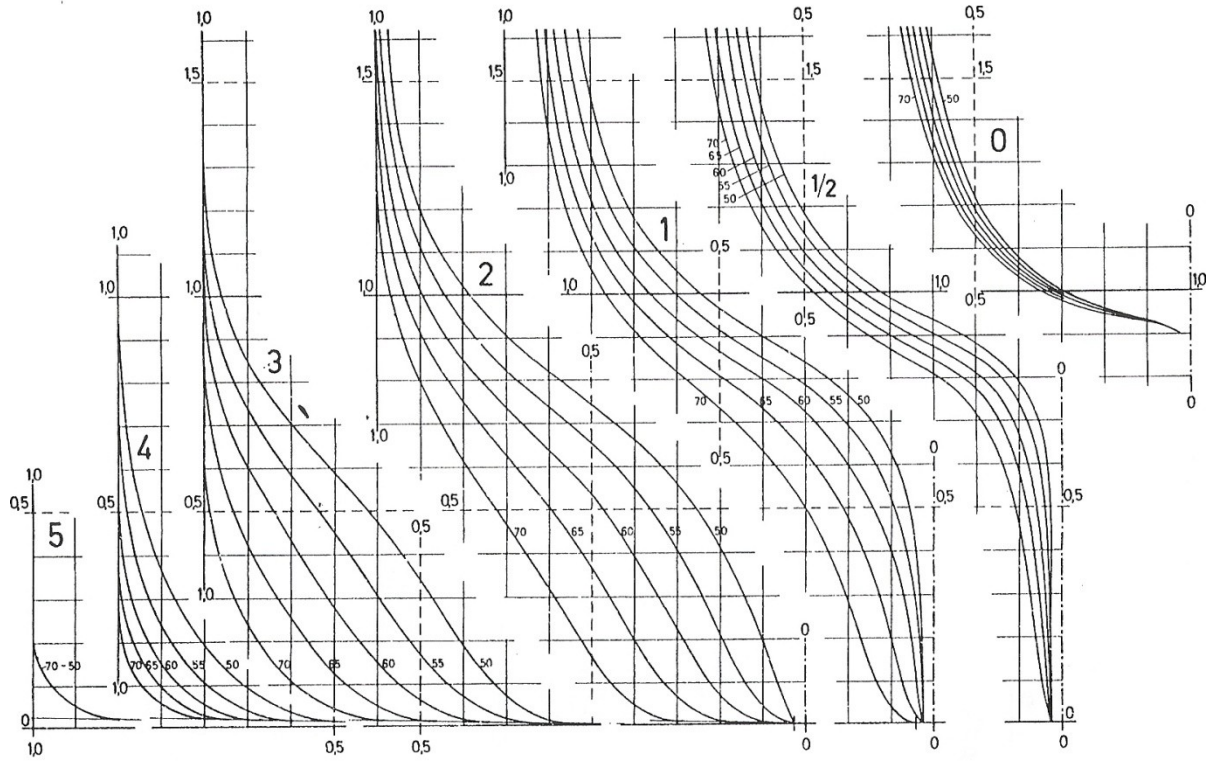
ΟΙΚΟΓΕΝΕΙΑ CA2A



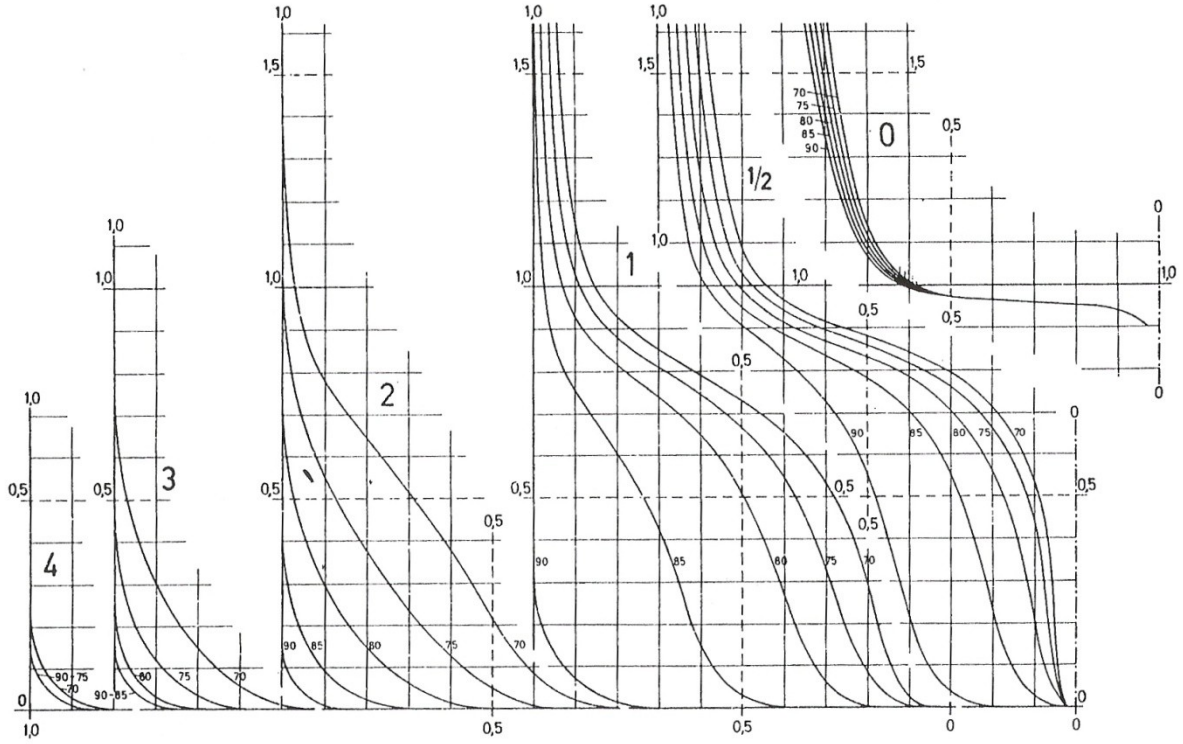
ΟΙΚΟΓΕΝΕΙΑ CB1A



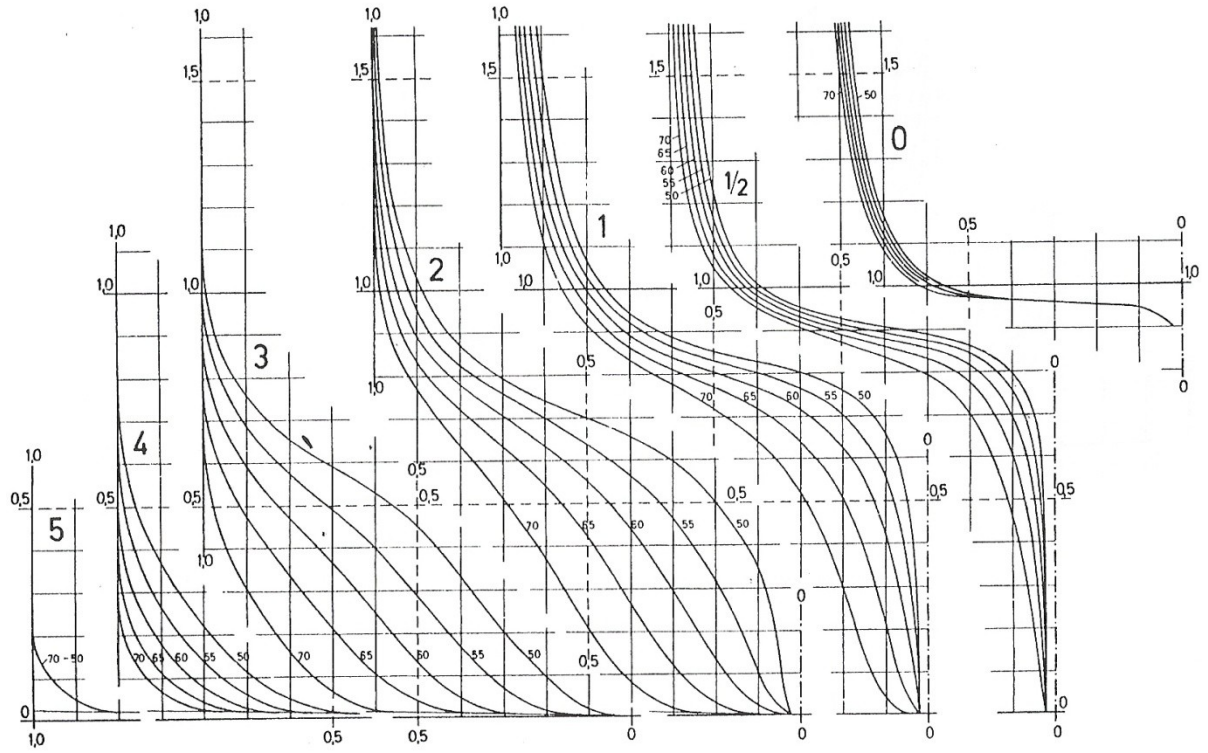
ΟΙΚΟΓΕΝΕΙΑ CB2A



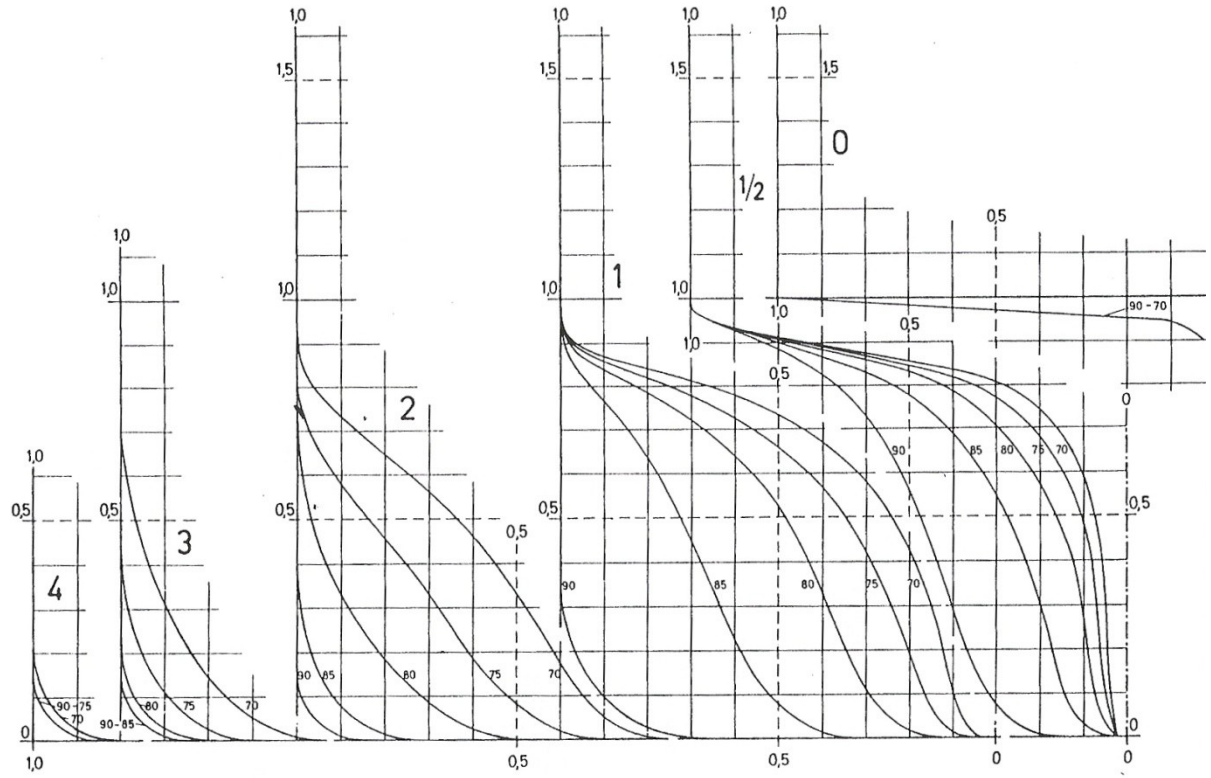
ΟΙΚΟΓΕΝΕΙΑ CC1A



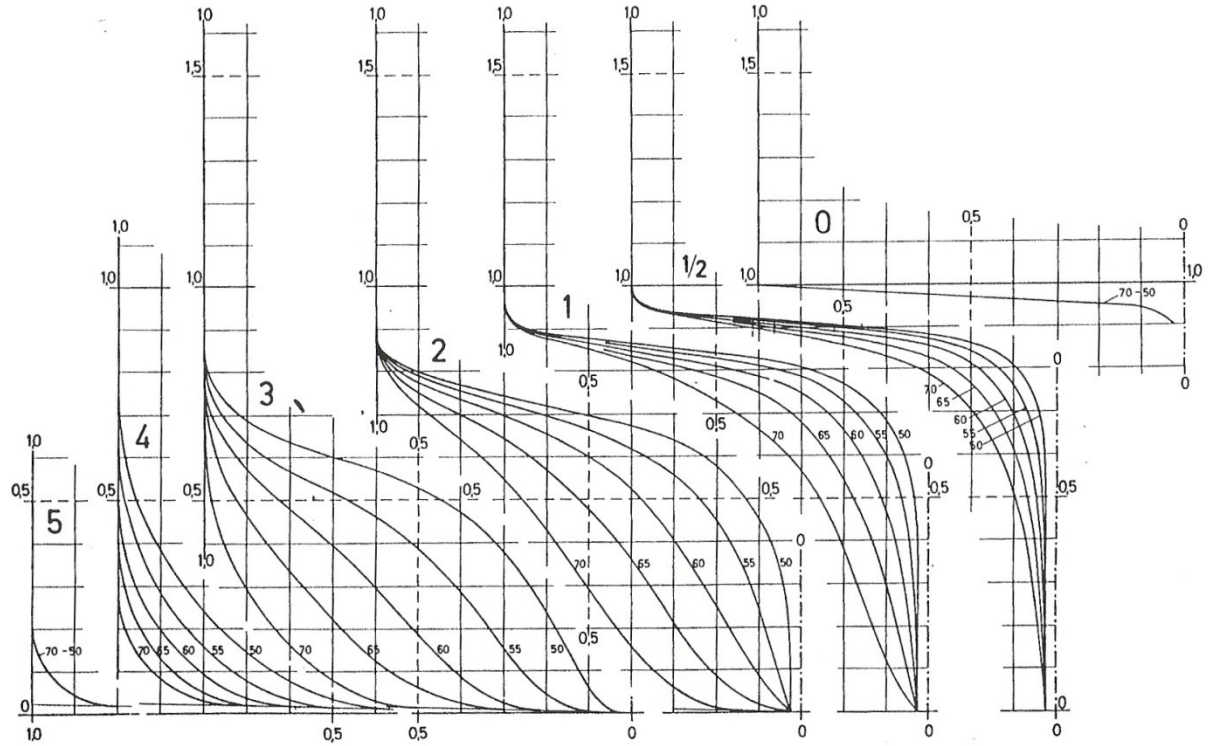
ΟΙΚΟΓΕΝΕΙΑ CC2A



ΟΙΚΟΓΕΝΕΙΑ CD1A



ΟΙΚΟΓΕΝΕΙΑ CD2A



8.2.Παράρτημα 2: «Catia Objects & Functions»

EXTREMUM

Object: Extremum

Function: AddNewExtremum (iObjet{[Reference](#)},
iDir{[HybridShapeDirection](#)},
iMinMax {long})

As [HybridShapeExtremum](#)

Description: Creates a new Extremum within the current body.

Parameters: iObjet : Element onto extremum is computed {[Reference](#)}
iDir : Extremum direction {[HybridShapeDirection](#)}
iMinMax : Maximum (GSMMax) or Minimum (GSMMin) {long}

Ext: The extremum object if succeeded

PLANE

Object: Plane

Function: AddNewPlaneOffsetPt (iPlane {[Reference](#)},
iPt {[Reference](#)})

As [HybridShapePlaneOffsetPt](#)

Description: Creates a new offset trough point plane within the current body

Parameters: iPlane : Reference plane {[Reference](#)},
iPt : Reference point {[Reference](#)},

Ext: Created plane

POINT

Object: Point

- 1) **Function:** AddNewPointCoord (iX{[Double](#)},
iY{[Double](#)},
iZ{[Double](#)})

As [HybridShapePointCoord](#)

Description: Creates a new point defined by its cartesian coordinates within the current body

Parameters: iX : X coordinate for the point
iY : Y coordinate for the point
iZ : X coordinate for the point

Ext: Created point

- 2) **Function:** AddNewPointOnCurveFromPercent (iCrv {[Reference](#)},
iLong {[Double](#)},
iOrientation {[Boolean](#)})

As [HybridShapePointOnCurve](#)

Description: Creates a new point on a curve from a ratio of distance to an extremity within the current body

Parameters: iCrv: support curve
iLong: ratio of curve length
iOrientation: Orientation = TRUE means that ratio is measured in the other orientation of the curve and from the other extremity

Ext: Created point

LINE

Object: Line

- 1) **Function:** AddNewLinePtPt (iPtOrigine {[Reference](#)},
iPtExtremite {[Reference](#)})

As [HybridShapeLinePtPt](#)

Description: Creates a new point-point line within the current body.

Parameters: iPtOrigine : Origin Point
iPtExtremite : Extremity Point

Ext: Created Line

- 2) **Function:** AddNewLinePtDir (iPt {[Reference](#)},
iDirection {[HybridShapeDirection](#)},
iBeginOffset {[Double](#)},
iOrientation {[Double](#)},
iEndOffset {[Boolean](#)})

As [HybridShapeLinePtDir](#)

Description: Creates a new point-direction line within the current body

Parameters: iPt : reference point
iDirection : direction
iBeginOffset : start offset
iOrientation : end offset
iEndOffset : Orientation allows to reverse the line direction from the reference point. For a line of L length, it is the same as creating this line with -L length

Ext: Created Line

SPLINE

Object: Spline

Function: AddNewSpline

As [HybridShapeSpline](#)

Description: Creates a new Spline within the current body.

Ext: Created Line

Methods: 1) AddPointWithConstraintFromCurve (ipIAPoint{[Reference](#)},
 ipIACurveCst{[Reference](#)},
 iTangencyNorm{[Double](#)},
 iInvertValue{[Long](#)},
 iCrvCstType{[Long](#)})

Description: Add a new point with tangency/curvature from a curve.

Parameters: ipIAPoint : Point element.
 ipIACurveCst : Curvature direction
 iTangencyNorm : tension factor for tangency
 iInvertValue : Orientation for tangent
 iCrvCstType : Continuity type for Curve Constraint
 (1=Tangency , 2=Curvature).

2) AddPoint (ipIAPoint{[Reference](#)})

Description: Add a new point

Parameters: ipIAPoint : Point element.

INTERSECTION

Object: Intersection

Function: AddNewIntersection (iObject1 {[Reference](#)},
iObject2 {[Reference](#)})

As [HybridShapeIntersection](#)

Description: Creates a new Intersection within the current body.

Parameters: iObject1 : First Element (line, curve, plane, surface)
iObject2 : Second Element (line, curve, plane, surface)

Ext: The intersection of the two objects

AFFINITY

Object: HybridShapeAffinity

Function: AddNewAffinity (iElement {[Reference](#)},
iXRatio {[Double](#)},
iYRatio {[Double](#)},
iZRatio {[Double](#)})

As [HybridShapeAffinity](#)

Description: Creates a new Affinity within the current body

Parameters: iElement : Point , curve, surface or solid
iXRatio : Ratio of Affinity in X direction
iYRatio : Ratio of Affinity in Y direction
iZRatio : Ratio of Affinity in Z direction

Ext: Created Affinity

Fill

Object: HybridShapeFill

Function: AddNewFill

As [HybridShapeFill](#)

Description: Creates a new Fill within the current body.

Ext: Fill object

Methods: 1)AddBound (iBoundary{[Reference](#)})

Description: Adds a boundary to the hybrid shape fill feature object.

Parameters: iBoundary: The boundary(curve) to be added to the hybrid shape fill feature object.

2)AddSupportAtBound (iBoundary{[Reference](#)}
iSupport{[Reference](#)})

Description: Inserts the support at specified boundary in the Fill.

Parameters: iBoundary: Reference of the boundary object to which support has to be added.

iSupport: Reference of the support object to be added.

JOIN

Object: HybridShapeAssemble

Function: AddNewJoin (Element1 {[Reference](#)}

Element2{[Reference](#)})

As [HybridShapeAssemble](#)

Description: Creates a new Join within the current body.

Parameters: Element1: First element to join (curve or surface)

Element2: Second element to join (same type of the first element)

Ext: Join result The default value used to join element is 0.001mm

SPLIT

Object: Split

Function: AddNewSplit (iSplittingElement {[Reference](#)},
iSplitSide {[CatSplitSide](#)})

as [Split](#)

Description: Creates and returns a new split operation within the current body

Parameters: iSplittingElement: The face or plane that will split the current body

iSplitSide: The specification for which side of the current body should be kept
at the end of the split operation

Ext: The created split operation

EXTRAPOLATE

Object: HybridShapeExtrapol

Function: AddNewExtrapolUntil (iBoundary {[Reference](#)},
iToExtrapol {[Reference](#)},
iUntil {[Reference](#)})

As [HybridShapeExtrapol](#)

Description: Creates a new Extrapol (until an element) within the current body.

Parameters: iBoundary: Boundary point of curve to extrapolate or boundary curve of
surface to extrapolate.

iToExtrapol: Curve or surface to extrapolate

iUntil: Extrapolation limit surface

Ext: Created Extrapolation

8.3. Παράρτημα 3: «Περιεχόμενα προγράμματος ανά γραμμή κώδικα»

final.py	
Block Name	Line No
INPUT SELECTION	21
PATHS	47
CATEGORIES 1-6	53
COPY STERN PROFILE	74
COPY STERN SECTIONS	82
COPY BOW SECTIONS	91
COPY BOW PROFILE	100
TOP POINTS CREATION	109
LOWER POINTS CREATION	121
DECK PLANE CREATION	151
FOB & FOS POINTS CREATION	163
REPLACE FSS POINTS TOO CLOSE TO DECK PLANE	166
FOS CATEGORY	168
FOS FORE TOP POINT AND FOS AFT TOP POINT CREATION	171
FOB CREATION	173
TRANSOM STERN DESIGN TYPE D	176
DECK LINE CREATION	290
TRANSOM STERN DESIGN TYPE ABC	293
CENTERL LINE CREATION	379
FOS CREATION	427
FOS WATERLINES CREATION	437
SCALE	444
SURFACE DESIGN	455
formdata.py	
Block Name	Line No
PRINCIPAL DIMENSIONS INPUT	14
BULB SELECTION	20
CB FORE CALCULATION	28
CB FORE ROUNDING	33
CB AFT CALCULATION	54
CB AFT ROUNDING	56
FORMDATA SERIES FORE AND AFT FAMILIES SELECTION	88
BULB SIZE DEFINITION	108
BOW PROFILE DEFINITION	114
STERN PROFILE DEFINITION	125
FoBFoSPoints.py	
Block Name	Line No
FOS POINTS CREATION	10

FOB POINTS CREATION	43
foscategycheck.py	
Block Name	Line No
foscategycheck.py	7
FoSForeAft.py	
Block Name	Line No
FOS FORE END CREATION	10
FOS FORE TOP POINT CREATION	27
FOS AFT END CREATION	39
FOS AFT TOP POINT CREATION	59
FOS FORE END CREATION	70
FOS FORE TOP POINT CREATION	87
FOS AFT END CREATION	98
FOS AFT TOP POINT CREATION	119
FoBCreation.py	
Block Name	Line No
GET FOB FREE POINTS Z COORDINATES INTO YCOORD LIST	9
FIND POINTS WITH SAME YCOORD	17
FOB WITH STRAIGHT PART	27
FOBLINE CREATION	42
FOB FORE SPLINE CREATION	53
FOB AFT SPLINE CREATION	70
FOBSPLINE CREATION (JOIN CURVES)	89
FOB WITHOUT STRAIGHT PART	103
FOB CREATION	105
CREATE FOB PLANE	117
CREATE FORE CL POINT	131
CREATE FOB FAR FORE SPLINE	159
CREATE AFT CL POINT	178
CREATE FOB FAR AFT SPLINE	196
CREATE FINAL FOB CURVE	215
DeckLineCreation.py	
Block Name	Line No
DECK LINE PARRALLEL CREATION	24
DECKLINE FORE CREATION	35
DECKLINE AFT CREATION	57
DECK LINE PARRALLEL CREATION	87
DECKLINE FORE CREATION	98
DECK LINE PARRALLEL CREATION	124
DECKLINE FORE CREATION	135
DECK LINE AFT CREATION	157
DECK LINE PARRALLEL CREATION	187
DECKLINE FORE CREATION	198
DECK LINE PARRALLEL CREATION	222

DECKLINE FORE CREATION	233
DECK LINE AFT CREATION	255
DECK LINE PARRALLEL CREATION	286
DECKLINE FORE CREATION	297
DECK LINE PARRALLEL CREATION	323
DECKLINE FORE CREATION	334
DECK LINE AFT CREATION	356
DECK LINE PARRALLEL CREATION	386
DECKLINE FORE CREATION	397
FoSCreation.py	
Block Name	Line No
GET FOS FREE POINTS Z COORDINATES INTO ZCOORD LIST	11
FIND POINTS WITH SAME ZCOORD	17
FOS LINE CREATION	41
FOS FORE SPLINE 2 CREATION	52
FOS AFT SPLINE 2 CREATION	75
FOS SPLINE CREATION (JOIN CURVES)	97
FOS FORE SPLINE CREATION	150
FOS AFT SPLINE CREATION	170
GET FOS FREE POINTS Z COORDINATES INTO ZCOORD LIST	195
FOS LINE 1	235
FORE FSS SPLINE	245
FOS LINE 2	270
MIDDLE FSS SPLINE	285
AFT FSS SPLINE	312
FOS LINE 1	345
FORE FSS SPLINE	355
AFT FSS SPLINE	378
AFT FSS SPLINE	409
FOS FORE SPLINE CREATION	427
FoSWaterLines.py	
Block Name	Line No
FORE FOS WL PLANE CREATION	11
FORE FOS WL INTERSECTIONS CREATION	23
FORE FOS WL CREATION	46
AFT FOS WL PLANE CREATION	66
AFT FOS WL INTERSECTIONS CREATION	81
AFT FOS WL CREATION	112
Affinity.py	
Block Name	Line No
SECTIONS AND PROFILES SCALE	8
FOB SCALE	32
CL SCALE	45
DECK LINE SCALE	60

FOS SCALE	80
TRANSOM SCALE	109
WATERLINES SCALE	129
replacefsspoints.py	
Block Name	Line No
FSS[0] Z COORDINATE	8
DELETE FSS[0]	14
REPLACE WITH TOP POINT FSS[0]	20
FSS[-1] Z COORDINATE	36
DELETE FSS[-1]	42
REPLACE WITH TOP POINT FSS[-1]	48
surface.py	
Block Name	Line No
SECTIONS CATEGORIZATION	8
JOIN CENTER LINE	75
JOIN DECKLINE	91
JOIN FOS PARTS	107
FLAT OF SIDE CREATION	154
FLAT OF BOTTOM CREATION	176
GENERAL CL CREATION	194
GENERAL DECK LINE CREATION	213
SPLIT SECTIONS	229
SUPPORT SURFACES	252
DECK LINE SUPPORT SURFACE	291
CL SUPPORT SURFACE	415
GENERAL CL SUPPORT SURFACE	462
GENERAL DECK LINE SUPPORT SURFACE	473
HIDE SUPPORT SURFACE	484
SURFACE BETWEEN FLAT OF SIDE AND FLAT OF BOTTOM	493
SPLIT SECTIONS FORWARD OF FSS	522
SURFACE UNDER FORE FOS WL	577
SURFACE OVER FORE FOS WL	618
FILL THE GAP	650
JOIN FORE SURFACE	687
FORE SURFACE CREATION	706
CREATE BOW SURFACE	749
FILL THE TIP	780
SPLIT SECTIONS AFT OF FSS	813
SPLIT AFT FOS WL	841
SURFACE UNDER AFT FOS WL	853
SURFACE OVER AFT FOS WL	886
FILL THE GAP	913
JOIN AFT SURFACE	950
AFT SURFACE CREATION	969

AFT SURFACE CREATION	1006
aft.py	
Block Name	Line No
CRUISER	7
TRANSOM D	268
TRANSOM ABC	471
bulb.py	
Block Name	Line No
FORE HELP SPLINE 1 CREATION	7
FORE HELP SPLINE 2 CREATION	79
BULB WL CREATION	152
SPLIT BOW	229
SPLIT SECTION 9.500	257
LOWER BULB SURFACE	269
FILL BULBOOUS BOW MIDDLE SURFACE	354
FILL BULBOOUS BOW TOP SURFACE	385
JOIN BULB SURFACES	446

9. ΒΙΒΛΙΟΓΡΑΦΙΑ

- Α.Παπανικολάου-Κ.Αναστασόπουλος, Μελέτη και Εξοπλισμός Πλοίου 1 (Μεθοδολογία Προμελέτης) Συλλογή Βοηθημάτων, Ε.Μ.Π., 3^η ανανεωμένη έκδοση, 2003
- Α.Παπανικολάου, Μελέτη Πλοίου Τόμος Α: Μεθοδολογία Προμελέτης, Εκδόσεις Συμείων, 1994
- Π.Δ. Κακλής, Γεωμετρική Σχεδίαση, Ε.Μ.Π., Μάιος 2005
- Π.Δ. Κακλής, Συστήματα CAD/CAM για την Σχεδίαση και Κατασκευή Πλοίων, Ε.Μ.Π., Μάρτιος 2009
- Mark Pilgrim , Dive Into Python 3, Apress, 2009
- Σαραϊδάρης Ιωάννης, Εξομάλυνση Συστηματικής Σειράς FORMDATA, Ε.Μ.Π., Ιούλιος 1998

