

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ ΜΕΤΑΦΟΡΩΝ ΚΑΙ
ΣΥΓΚΟΙΝΩΝΙΑΚΗΣ ΥΠΟΔΟΜΗΣ

ΕΠΟΠΤΕΣ : Π. Ψαράκη - Καλουπτσίδη
Φ. Μερτζάνης

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ
ΓΕΩΜΕΤΡΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ
ΓΙΑ ΤΗΝ ΣΤΕΡΕΟΜΕΤΡΙΚΗ
ΑΝΑΛΥΣΗ ΑΕΡΟΔΡΟΜΙΟΥ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΘΑΝΑΣΙΟΥ ΝΙΚΟΛΑΟΥ

ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2009

Σύνοψη διπλωματικής εργασίας

Στην παρούσα διπλωματική εργασία, με τίτλο «Σχεδιασμός και υλοποίηση γεωμετρικών αλγορίθμων για την στερεομετρική ανάλυση αεροδρομίου» ερευνάται ένα από τα κυριότερα χωρικά προβλήματα που παρουσιάζονται σε περιοχή εγγύτητας αεροδρομίου, ο υπολογισμός επιτρεπτού υψομέτρου σημείου, όπως ορίζεται από τις απαιτήσεις των επιφανειών περιορισμού εμποδίων. Για την ενασχόληση με το θέμα χρησιμοποιείται ο προγραμματισμός σε γλώσσα C++ και η υπολογιστική γεωμετρία, εργαλεία με τα οποία γίνεται η ανάπτυξη προγράμματος στο περιβάλλον του Visual Studio (Microsoft).

Παρουσιάζεται ο σχεδιασμός και η επεξήγηση των αλγοριθμικών μεθόδων που επιτρέπουν την προγραμματιστική επίλυση γεωμετρικών προβλημάτων. Έπειτα από μελέτη και παραμετροποίηση των επιφανειών περιορισμού εμποδίων, παρουσιάζεται το περιβάλλον του Visual Studio, όπου οι μέθοδοι συντίθενται σε μια λειτουργική ολότητα, το πρόγραμμα rPORT. Στην πορεία ανάπτυξης του προγράμματος, γνώμονας σχεδιασμού υπήρξε η πρακτικότητα του χειρισμού (δημιουργία γραφικού περιβάλλοντος αλληλεπίδρασης) και η επεκτασιμότητα, επί της οποίας γίνεται ιδιαίτερη μνεία στο τέλος.

Η χρήση του προγράμματος, μπορεί να γίνει στα πλαίσια σχεδιασμού ή επέκτασης ενός αερολιμένα, με σκοπό τον έλεγχο επιτρεπόμενων υψομέτρων των προτεινόμενων κατασκευών, ενώ επιχειρείται όπως θα δούμε και διεύρυνση της χρηστικότητας.

Abstract

The present Thesis, entitled “Design and implementation of geometric algorithms for airport’s solid geometry analysis ” is dealing with a main spatial issue that occurs in an airport’s area of proximity, and that is the calculation of a point’s allowable altitude, as it is defined by the obstacle limitation surfaces’ requirements. To approach the subject, computational geometry and C++ programming are being used to develop an application in Microsoft’s Visual Studio environment.

The design and epexegetis of the algorithmic methods that allow as to program solutions on geometric problems are being mapped out. After the study and parameterization of the obstacle limitation surfaces, we get introduced to the Visual Studio’s design environment, where the methods are combined into a functional sum, the rPORT program. Throughout the development of the application, main quality criteria have been the convenient usage (for example creation of Graphical User Interface) and the capability of expansion (for which a special reference is being made at the end of the thesis).

The rPORT program can be used during the design or expansion of an airport, to control the altitude of constructions, whereas an expansion of its usability is being attempted as we will see further on.

Περίληψη περιεχομένων κεφαλαίων

Η διπλωματική εργασία εκτείνεται σε ένδεκα κεφάλαια, το αδρομερές περιεχόμενο του καθενός συνοψίζεται ως ακολούθως:

Κεφάλαιο 1^ο – Εισαγωγή στην έννοια των αερομεταφορών, συσχετίσή τους με την επιστήμη των ηλεκτρονικών υπολογιστών και στοχοθέτηση της εργασίας.

Κεφάλαιο 2^ο – Γίνεται αρχικά ορισμός, διάκριση και κατηγοριοποίηση των αεροδρομίων. Έπειτα παρουσιάζεται η έννοια του αεροσκάφους σχεδιασμού και η περιβαλλοντική παράμετρος ως παράγοντες σχεδιασμού αεροδρομίων.

Κεφάλαιο 3^ο – Παρουσιάζονται οι επιφάνειες περιορισμού εμποδίων, όπως ορίζονται από το Annex 14 του ICAO, για τα διάφορα είδη διαδρόμων καθώς και οι υψομετρικοί περιορισμοί που αυτές επιβάλλουν στα διάφορα είδη αντικειμένων.

Κεφάλαιο 4^ο – Εξετάζεται η επιρροή μετεωρολογικών, τοπογραφικών, γεωμετρικών συνθηκών στο βασικό μήκος διαδρόμου και των ανέμων στην διεύθυνσή του.

Κεφάλαιο 5^ο – Παρουσιάζονται τα δεδομένα διαστασιολόγησης του διαδρόμου και βοηθητικών επιφανειών, καθώς και ο τρόπος με τον οποίο τα στοιχεία αυτά μορφοποιούν τη γεωμετρία του.

Κεφάλαιο 6^ο – Δηλώνεται και επεξηγείται η μορφή με την οποία τα απαραίτητα δεδομένα πρόκειται να αποθηκευτούν στο πρόγραμμα καθώς και οι χρησιμοποιούμενες συμβάσεις συντεταγμένων και ονοματολογίας.

Κεφάλαιο 7^ο – Γίνεται εισαγωγή στην έννοια των γεωμετρικών αλγορίθμων ενώ παρουσιάζονται και επεξηγούνται ο σχεδιασμός και η μερική υλοποίηση αυτών που χρησιμοποιούνται στο πρόγραμμα rPORT. Επίσης αναφερόμαστε και στις βιβλιοθήκες γεωμετρικών αλγορίθμων.

Κεφάλαιο 8^ο – Παρουσιάζεται η παραμετροποίηση των επιφανειών περιορισμού εμποδίων στο πρόγραμμα, δηλαδή ο τρόπος με τον οποίο το rPORT «βλέπει» τις επιφάνειες, ως σύνθεση δηλαδή των ελάχιστων δυνατών μεταβλητών.

Κεφάλαιο 9^ο – Γίνεται εισαγωγή στο περιβάλλον της Visual C++ 2008 και παρουσιάζονται οι κυριότερες έννοιες που απαιτούνται για τον προγραμματισμό σε αυτό. Ακολουθεί εκτέλεση του προγράμματος με παράλληλη παρακολούθηση της μεταβολής των αντίστοιχων μεταβλητών και του κώδικα που «τρέχει» στο παρασκήνιο (παρουσίαση δηλαδή της τελικής υλοποίησης) ενώ τέλος παρατίθεται τυπωμένη μια έξοδος του προγράμματος.

Κεφάλαιο 10^ο – Παρουσιάζονται προτάσεις επέκτασης και βελτίωσης του προγράμματος rPORT ενώ δίδονται και περιπτώσεις χρήσης του προγράμματος είτε ως έχει είτε με μικρές τροποποιήσεις.

Κεφάλαιο 11^ο – Παρατίθενται συμπεράσματα που προέκυψαν κατά την ανάπτυξη του rPORT και την ενασχόληση των σχετικά με αυτό αντικειμένων.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ – ΣΤΟΧΟΙ	1
1.1 – ΑΕΡΟΜΕΤΑΦΟΡΕΣ	2
1.2 – Η/Υ & ΑΕΡΟΜΕΤΑΦΟΡΕΣ	3
1.3 – ΣΤΟΧΟΙ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	4
ΚΕΦΑΛΑΙΟ 2 - ΑΕΡΟΔΡΟΜΙΑ – ΑΕΡΟΣΚΑΦΗ	5
2.1 – ΓΕΝΙΚΑ	6
2.2 – ΔΙΑΚΡΙΣΗ ΤΩΝ ΑΕΡΟΔΡΟΜΙΩΝ	7
2.3 – ΚΥΡΙΕΣ ΚΑΤΑΣΚΕΥΕΣ ΑΕΡΟΔΡΟΜΙΩΝ	8
2.4 – ΚΩΔΙΚΟΣ ΑΝΑΦΟΡΑΣ ΑΕΡΟΔΡΟΜΙΟΥ	8
2.5 – ΑΕΡΟΣΚΑΦΗ	10
2.6 – ΠΕΡΙΒΑΛΛΟΝΤΙΚΕΣ ΕΠΙΠΤΩΣΕΙΣ ΑΕΡΟΜΕΤΑΦΟΡΩΝ	12
ΚΕΦΑΛΑΙΟ 3 - ΠΕΡΙΟΡΙΣΜΟΣ ΚΑΙ ΑΠΟΜΑΚΡΥΝΣΗ ΕΜΠΟΔΙΩΝ	16
3.1 – ΕΠΙΦΑΝΕΙΕΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ	21
3.2 – ΑΠΑΙΤΗΣΕΙΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ	27
3.3 – ΑΝΤΙΚΕΙΜΕΝΑ ΕΚΤΟΣ ΤΩΝ ΕΠΙΦΑΝΕΙΩΝ	33
3.4 – ΑΛΛΑ ΑΝΤΙΚΕΙΜΕΝΑ	33
ΚΕΦΑΛΑΙΟ 4 - ΕΠΙΔΡΑΣΗ ΤΟΠΙΚΩΝ – ΓΕΩΜΕΤΡΙΚΩΝ ΣΥΝΘΗΚΩΝ ΣΤΟΝ ΣΧΕΔΙΑΣΜΟ ΑΕΡΟΔΡΟΜΙΩΝ	36
4.1 – ΕΠΙΡΡΟΗ ΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΣΥΝΘΗΚΩΝ	38
4.2 – ΕΠΙΡΡΟΗ ΤΩΝ ΑΝΕΜΩΝ - ΑΝΕΜΟΛΟΓΙΟ	39
4.3 – ΕΠΙΡΡΟΗ ΤΟΠΟΓΡΑΦΙΚΩΝ ΣΥΝΘΗΚΩΝ	41
4.4 – ΕΠΙΡΡΟΗ ΓΕΩΜΕΤΡΙΚΩΝ ΣΥΝΘΗΚΩΝ ΣΤΟ ΜΗΚΟΣ ΔΙΑΔΡΟΜΟΥ	42
ΚΕΦΑΛΑΙΟ 5 - ΓΕΩΜΕΤΡΙΚΗ ΜΕΛΕΤΗ ΑΕΡΟΔΡΟΜΙΩΝ	43
5.1 – ΔΙΑΔΡΟΜΟΣ - ΕΙΣΑΓΩΓΙΚΑ	45
5.2 – Δ ΗΛΟΥΜΕΝΕΣ ΑΠΟΣΤΑΣΕΙΣ	45
5.3 – ΖΩΝΕΣ ΑΣΦΑΛΕΙΑΣ	47
5.4 – ΦΥΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΔΙΑΔΡΟΜΟΥ	47
ΚΕΦΑΛΑΙΟ 6 - ΑΠΟΘΗΚΕΥΣΗ ΚΑΙ ΧΡΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ	51

6.1– ΔΟΜΗΣΗ ΔΕΔΟΜΕΝΩΝ	52
6.2 – ΣΥΣΤΗΜΑΤΑ ΣΥΝΤΕΤΑΓΜΕΝΩΝ	54
6.3 – ΟΝΟΜΑΤΟΛΟΓΙΑ ΜΕΤΑΒΛΗΤΩΝ	55
ΚΕΦΑΛΑΙΟ 7 - ΓΕΩΜΕΤΡΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ	57
7.1 – ΓΕΩΜΕΤΡΙΚΟΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ	58
7.2 – ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΣ ΤΡΙΩΝ ΣΗΜΕΙΩΝ	62
7.3 – Δ ΙΑΤΑΞΕΙΣ ΕΥΘΥΓΡΑΜΜΩΝ ΤΜΗΜΑΤΩΝ – ΕΛΕΓΧΟΣ ΤΟΜΗΣ	63
7.4 – ΠΕΡΙΛΗΨΗ ΣΕ ΕΠΙΦΑΝΕΙΑ	65
7.5 – ΥΠΟΛΟΓΙΣΜΟΣ ΥΨΟΜΕΤΡΩΝ – ΧΕΙΡΙΣΜΟΣ ΤΟΥ ΑΞΟΝΑ Ζ	72
7.6 – ΒΙΒΛΙΟΘΗΚΕΣ ΓΕΩΜΕΤΡΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ	74
ΚΕΦΑΛΑΙΟ 8 - ΟΙ ΕΠΙΦΑΝΕΙΕΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ ΣΤΟ ΠΡΟΓΡΑΜΜΑ	76
• ΑΞΟΝΑΣ ΔΙΑΔΡΟΜΟΥ	78
8.1 – ΖΩΝΗ ΑΣΦΑΛΕΙΑΣ	79
8.2 – ΕΣΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ ΕΠΙΦΑΝΕΙΑ	80
8.3 – ΚΩΝΙΚΗ ΕΠΙΦΑΝΕΙΑ	80
8.4 – ΕΞΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ ΕΠΙΦΑΝΕΙΑ	81
8.5 – ΕΠΙΦΑΝΕΙΑ ΠΡΟΣΕΓΓΙΣΗΣ	81
8.6 – ΕΠΙΦΑΝΕΙΑ ΑΠΟΓΕΙΩΣΗΣ	83
8.7 – ΜΕΤΑΒΑΤΙΚΗ ΕΠΙΦΑΝΕΙΑ	84
ΚΕΦΑΛΑΙΟ 9 - ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ Η/Υ	88
9.1 – ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ VISUAL C++ 2008	90
9.2 – ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	95
9.3 – ΔΟΚΙΜΑΣΤΙΚΗ ΕΞΟΔΟΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	101
9.4 – ΟΠΤΙΚΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	105
ΚΕΦΑΛΑΙΟ 10 - ΠΡΟΤΑΣΕΙΣ ΕΠΕΚΤΑΣΗΣ – ΧΡΗΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ rPORT	107
10.1 – ΒΕΛΤΙΩΣΕΙΣ	108
10.2 – ΠΡΟΣΘΗΚΗ ΕΡΓΑΛΕΙΩΝ	110
10.3 – ΧΡΗΣΕΙΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	111
ΚΕΦΑΛΑΙΟ 11 – ΣΥΜΠΕΡΑΣΜΑΤΑ	113
ΒΙΒΛΙΟΓΡΑΦΙΑ	116

ΠΑΡΑΡΤΗΜΑ Α – ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	118
ΠΑΡΑΡΤΗΜΑ Β – ΕΛΕΓΧΟΣ ΣΕ ΑΜΦΙΘΕΑΤΡΙΚΕΣ ΕΠΙΦΑΝΕΙΕΣ	199
ΠΑΡΑΡΤΗΜΑ Γ – ΠΙΝΑΚΕΣ, ΣΧΗΜΑΤΑ & ΕΙΚΟΝΕΣ	203
ΠΑΡΑΡΤΗΜΑ Δ – ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ	206

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ – ΣΤΟΧΟΙ

1.1 – ΑΕΡΟΜΕΤΑΦΟΡΕΣ

Πριν το 1903 και τους αδερφούς Wright καμία κατασκευή δεν είχε πετάξει, πριν το 1920 δεν υπήρχαν αερογραμμές όπως τις ξέρουμε και πριν το 1939 δεν υπήρχαν αεροσκάφη μορφής jet. Οι εναέριες μεταφορές διέπονται από το χαρακτηριστικό της ταχείας ανάπτυξης και εξέλιξης, τόσο που πλέον θεωρούνται περισσότερο δικαίωμα παρά προνόμιο. Αυτό μερικώς οφείλεται σε αλλαγές στην κοινή γνώμη, που αναγνωρίζει σήμερα το δικαίωμα της κινητικότητας (mobility) ως βαθιά ριζωμένη κοινωνική αξία, ως τρόπο ζωής.

Οι βασικές αρχές της πτήσης θεωρείται γενικά πως εδραιώθηκαν από τον Sir George Cayley. Οι πρώτες πτήσεις αερογραμμών έγιναν στην Γερμανία το 1910 με τεράστια αερόπλοια της εταιρίας Zeppelin, που πριν τον πρώτο Παγκόσμιο Πόλεμο είχαν μεταφέρει πάνω από 35000 επιβάτες. Το βομβαρδιστικό που κατασκευάστηκε από την Handley – Page το 1918 ήταν ίσως το πρώτο μεγάλης κλίμακας αεροσκάφος που χρησιμοποιήθηκε εμπορικά, όταν τροποποιήθηκε ώστε να μεταφέρει δώδεκα επιβάτες. Η πρώτη απόπειρα λειτουργίας τακτικής επιβατικής υπηρεσίας έγινε στη Φλόριντα το 1914 με την γραμμή υδροπλάνου Tampa – St Petersburg η οποία μετέφερε 1200 συνολικά επιβάτες σε τέσσερις μήνες, εκτελώντας δύο πτήσεις ανά ημέρα.

Παρότι μόνο μικρό ποσοστό του πληθυσμού κάθε χώρας εμπλέκεται στη σχεδίαση, κατασκευή ή τον χειρισμό ενός αεροσκάφους, ο καθένας μπορεί να θεωρηθεί έμμεσα ευεργετούμενος από τις αερομεταφορές. Το εμπόριο, η τραπεζική λειτουργία, η αγροτική ανάπτυξη, η βιομηχανία, μεγάλες και μικρές επιχειρήσεις και όλοι εκείνοι των οποίων οι ζωές βασίζονται στην επικοινωνία με ανθρώπους που βρίσκονται σε απόσταση θα έπρεπε να υποστούν μεγάλες απώλειες αν ο κλάδος των αερομεταφορών διαλυόταν ή περιοριζόταν.

Τα στοιχεία που χαρακτηρίζουν τις σύγχρονες αερομεταφορές είναι τα ακόλουθα :

- I. Συνεχής και δυναμική επέκταση, δηλαδή αύξηση των αερομεταφερόμενων επιβατών και εμπορευμάτων, αύξηση του αριθμού των πτήσεων, αύξηση του αριθμού και του μεγέθους των αεροδρομίων και τέλος κάλυψη περισσότερων περιοχών.
- II. Διάθεση μεγάλων οικονομικών πόρων για πάγια περιουσιακά στοιχεία (εγκαταστάσεις εδάφους, α/φ, έργα υποδομής κλπ) και για την προμήθεια και βελτίωση του αεροπορικού υλικού.
- III. Συνεχής αύξηση των απαιτήσεων σε ειδικευμένο προσωπικό (εδάφους, αέρα) για την υποστήριξη του συστήματος.
- IV. Ταυτόχρονη εξέλιξη πολλών συγγενικών τεχνολογικών τομέων, όπως η ηλεκτρονική, τα συστήματα αυτοματισμού, η εδαφομηχανική, η τεχνική της οδοστρωσίας.

- V. Ύπαρξη ανοιχτού πεδίου έρευνας και συνεχιζόμενου σχεδιασμού της εξέλιξης των μεταφορικών συστημάτων (υποσυστημάτων, δικτύων κλπ), σε συνδυασμό με τη στάθμη της μεταφορικής εξυπηρέτησης [βλ. Κ.Γ. Αμπακούμκιν (1990),2]

Ο όγκος του αερομεταφορικού έργου εικοσαπλασιάστηκε από την δεκαετία του '70 μέχρι τη δεκαετία του '90. Καθώς όλα προέβλεπαν παρόμοια εξέλιξη και πορεία, ο κλάδος των αερομεταφορών απορρόφησε τεράστια κεφάλαια και επενδύσεις, αφήνοντας και ανάλογες προσδοκίες. Όμως η αστάθεια στις τιμές των καυσίμων, το τρομοκρατικό χτύπημα της 11^{ης} Σεπτεμβρίου και η πρόσφατη (ακόμα σε εξέλιξη) οικονομική κρίση αποτέλεσαν ανασταλτικούς παράγοντες στην εξέλιξη και πολλές φορές ακόμα και τη βιωσιμότητα των αερομεταφορών.

Με τα νέα δεδομένα, ο φόβος της χρήσης ενός αεροσκάφους ως όπλο μαζικής καταστροφής, έθεσε νέα κριτήρια ασφαλείας και ανάγκασε τις σχετιζόμενες με τον κλάδο εταιρίες να εισέλθουν σε μια πολυέξοδη προσπάθεια επανάκτησης της εμπιστοσύνης του κοινού. Ακόμα, συγχωνεύσεις και εξαγορές εταιριών με αντίστοιχες απολύσεις προσωπικού και περιορισμό των δρομολογίων, ήταν τα μέσα με τα οποία επιχειρήθηκε η επίτευξη ανταγωνιστικότητας και μείωση των κομίστρων. Η ρευστότητα της παγκόσμιας οικονομίας, αποτρέπει τους ειδικούς από μακροπρόθεσμες (ή ακόμα και μεσοπρόθεσμες) προβλέψεις σχετικά με το σύνολο σχεδόν των επιχειρηματικών δραστηριοτήτων, κάτι που επιβάλλει και θα συνεχίζει να επιβάλλει, μεταξύ άλλων και στον κλάδο των αερομεταφορών να σχεδιάζει τις «κινήσεις» του με χειρουργική ακρίβεια και εξαντλητική προνοητικότητα.

1.2 – Η/Υ & ΑΕΡΟΜΕΤΑΦΟΡΕΣ

Στην προηγούμενη παράγραφο, αναφέρθηκε η ανάγκη ακρίβειας στον σχεδιασμό των αερομεταφορών. Στα πλαίσια της πολυπλοκότητας της ανάπτυξης του συστήματος των αερομεταφορών, αυτή είναι τεχνικά ανέφικτη χωρίς την αξιοποίηση της τεχνολογίας των Η/Υ. Πέρα από το πλαίσιο της σύγχρονης εποχής (ρευστότητα της παγκόσμιας οικονομίας), οι Η/Υ έχουν εδώ και καιρό αναδειχθεί απαραίτητο εργαλείο για την επίλυση προβλημάτων που ανακύπτουν κατά τον σχεδιασμό και την κατασκευή αερολιμένων και αεροσκαφών. Ειδικότερα η χρησιμοποίηση Η/Υ είναι απαραίτητη για :

- Την διατύπωση αξιόπιστων προοπτικών σε ότι αφορά την μελλοντική πορεία της αεροπορικής κίνησης
- Το σχεδιασμό και την χάραξη εθνικής αναπτυξιακής στρατηγικής στον τομέα των μεταφορών και στον κλάδο των αερομεταφορών (δημιουργία

- υποδομής, σύστημα ελέγχου αρτιότητας εκτέλεσης και ομαλής λειτουργίας των αεροδρομίων κλπ)
- Το σχεδιασμό της κατασκευής των αεροδρομίων και συμπληρωματικών εγκαταστάσεων, με την ευρύτερη έννοια των αερολιμένων (βλ. κεφ. 2)
 - Την προσομοίωση λειτουργίας ενός αεροδρομίου, για την βελτιστοποίηση είτε του σχεδιασμού είτε της λειτουργίας υπάρχοντος αερολιμένα.
 - Την κάλυψη των «υπολογιστικών αναγκών» που προκύπτουν από τη χρήση σύγχρονων συστημάτων ασφαλείας και ελέγχου.

1.3 – ΣΤΟΧΟΙ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η διπλωματική αυτή εργασία, με τίτλο «ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΓΕΩΜΕΤΡΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΤΗΝ ΣΤΕΡΕΟΜΕΤΡΙΚΗ ΑΝΑΛΥΣΗ ΑΕΡΟΔΡΟΜΙΟΥ», στόχο έχει την κατασκευή προγράμματος Η/Υ που με δεδομένο εισόδο τις συντεταγμένες ενός σημείου και τα χαρακτηριστικά αεροδρομίου, θα αποφαινεται περί του επιτρεπτού του υψομέτρου, όπως αυτό ορίζεται από τη δυσμενέστερη επιφάνεια περιορισμού εμποδίων. Κριτήρια ποιότητας για την κατασκευή του προγράμματος αποτέλεσαν η ευκολία χρήσης (δημιουργία γραφικού περιβάλλοντος αλληλεπίδρασης με τον χρήστη) και η δυνατότητα επέκτασης – βελτίωσης.

Για την κατασκευή ενός τέτοιου προγράμματος, απαραίτητο είναι να μελετηθούν διεξοδικά οι προαναφερθείσες επιφάνειες και να οριστούν οι μέθοδοι και τα γνωστικά πεδία (μαθηματικά εργαλεία) που μας επιτρέπουν να καταπιαστούμε με γεωμετρικά (στη βάση τους) προβλήματα σε προγραμματιστικά περιβάλλοντα.

Η όλη πορεία εργασίας έχει την κατεύθυνση εξεύρεσης μεθοδολογίας και αναζήτησης σχετικού θεωρητικού υποβάθρου, που θα προσφέρει μια πιο σύγχρονη οπτική στην επίλυση (φαινομενικά παραδοσιακών) προβλημάτων που αφορούν τον πολιτικό μηχανικό. Η Υπολογιστική Γεωμετρία και οι αντικειμενοστραφείς γλώσσες προγραμματισμού που μελετήθηκαν, χρησιμοποιήθηκαν και παρουσιάζονται στα επόμενα κεφάλαια, ελπίζουμε να πέτυχαν αυτόν το στόχο.

ΚΕΦΑΛΑΙΟ 2

ΑΕΡΟΔΡΟΜΙΑ – ΑΕΡΟΣΚΑΦΗ

2.1 – ΓΕΝΙΚΑ

Κατά τον επίσημο ορισμό του (Annex 14), αεροδρόμιο είναι μια ορισμένη επιφάνεια επί εδάφους ή ύδατος (συμπεριλαμβανομένου κάθε κτιρίου, εγκαταστάσεων και εξοπλισμού) με σκοπό να χρησιμοποιηθεί εξ ολοκλήρου ή μερικώς για την άφιξη, την αναχώρηση και την επίγεια κίνηση αεροσκαφών.

Ο συνώνυμος (αλλά γενικότερος όρος) «αερολιμένας» αφορά το σύνολο των κάθε φύσεως κατασκευών και εγκαταστάσεων που εξυπηρετούν τις αεροπορικές μεταφορές ατόμων και εμπορευμάτων. Από καθαρά συγκοινωνιακή άποψη είναι συγκοινωνιακός κόμβος, όπου το δίκτυο αεροπορικών μεταφορών συνδέεται με τα επίγεια δίκτυα μεταφορών.

Σε αναλυτικότερη (χωρίς διάκριση ωστόσο) προσέγγιση των δύο όρων, αεροδρόμιο ή αερολιμένας είναι ο χώρος και οι κατασκευές που εξυπηρετούν τις ακόλουθες ανάγκες:

- Σε σχέση με τα αεροσκάφη
 - Προσγειαοαπογειώσεις
 - Στάθμευση, κατά τον χρόνο που δεν χρησιμοποιούνται
 - Στάθμευση, κατά την επιβίβαση – αποβίβαση επιβατών και αποσκευών
 - Φορτοεκφόρτωση εμπορευμάτων κλπ
 - Ανεφοδιασμός, καθαρισμός κλπ
 - Συντήρηση και επισκευές
- Σε σχέση με τους επιβάτες που θα ταξιδέψουν
 - Σύντομη παραμονή επιβατών και των συνοδών τους
 - Έλεγχος εισιτηρίων, βάρους αποσκευών καθώς και παραλαβή των αποσκευών των επιβατών
 - Μεταφορά των αποσκευών και φόρτωση στα αεροσκάφη
 - Έλεγχοι ασφάλειας
 - Πρόσθετοι έλεγχοι για τους επιβάτες του εξωτερικού
 - Αναμονή των επιβατών μέχρι της επιβίβασης
 - Μεταφορά των επιβατών από τους χώρους αναμονής στο αεροσκάφος
 - Εξυπηρέτηση των μέσων μεταφοράς που χρησιμοποιούνται για μετάβαση στο αεροδρόμιο
- Σε σχέση με τους επιβάτες που ταξίδεψαν
 - Μεταφορά τους από το αεροσκάφος στο χώρο παραλαβής των αποσκευών
 - Παραλαβή των αποσκευών
 - Έλεγχοι για τους επιβάτες του εξωτερικού
 - Χώροι αναμονής για τους επιβάτες που θα συνεχίσουν το ταξίδι τους με το ίδιο ή άλλο αεροσκάφος

- Διακίνηση των επιβατών προς τα μέσα επίγειων μεταφορών που εξυπηρετούν το αεροδρόμιο και ευκολίες για τα μέσα αυτά
 - Σε σχέση με τα εμπορεύματα, ταχυδρομείο κλπ ανάλογες με αυτές των επιβατών ανάγκες
 - Σε σχέση με τις επιχειρήσεις που με οποιοδήποτε τρόπο συμμετέχουν στην κάλυψη των παραπάνω αναγκών
- Εξυπηρέτηση του προσωπικού τους (διακίνηση, εργασία κλπ)
- Εξυπηρέτηση των ίδιων των επιχειρήσεων, με χώρους γραφείων κλπ

2.2 – ΔΙΑΚΡΙΣΗ ΤΩΝ ΑΕΡΟΔΡΟΜΙΩΝ

Τα αεροδρόμια ανάλογα με το σκοπό που εξυπηρετούν, ή αν εξυπηρετούν περισσότερους του ενός, ανάλογα με το βασικό τους σκοπό, διακρίνονται σε:

- Πολιτικά (για δημόσια ή ιδιωτική χρήση)
- Στρατιωτικά

Τα κριτήρια σχεδιασμού ενός αεροδρομίου εξαρτώνται από τον εξυπηρετούμενο σκοπό. Για παράδειγμα ένα πολιτικό αεροδρόμιο τοποθετείται σχεδόν πάντοτε κοντά σε ένα αστικό κέντρο και γενικότερα σ' ένα κέντρο που αποτελεί πόλο έλξης-γενέσεως αερομεταφορικού έργου (για παράδειγμα κάποιο μεγάλο βιομηχανικό συγκρότημα ή μια έντονα τουριστική περιοχή). Ένα στρατιωτικό αεροδρόμιο αντίθετα, ανάλογα με τον ειδικότερο σκοπό του μπορεί να τοποθετηθεί κοντά σε μια μεγάλη βάση ή σε κατάλληλη στρατηγική θέση. Άλλο κριτήριο σχεδιασμού είναι η βαρύτητα της οικονομικής ανταποδοτικότητας : ενώ για ένα πολιτικό αεροδρόμιο η επίτευξη καλής ανταποδοτικότητας αποτελεί μείζον κριτήριο, ένα στρατιωτικό αεροδρόμιο πρακτικά αγνοεί τον στόχο αυτόν.

Η παρούσα διπλωματική εργασία αναφέρεται στα σύγχρονα πολιτικά αεροδρόμια δημόσιας χρήσεως που εξυπηρετούν τις αερομεταφορές, εσωτερικού και διεθνείς. Αυτά άλλωστε είναι και τα πολυπλοκότερα από πλευράς καλύψεως απαιτήσεων και λειτουργιών. Είναι όμως γεγονός πως ότι πολλά στοιχεία, όπως οι μεθοδολογίες γεωμετρικού σχεδιασμού, υπολογισμού οδοστρωμάτων κλπ, είναι ίδιες και για τα αεροδρόμια άλλων κατηγοριών. Η διαφορά στις περιπτώσεις αυτές έγκειται στους κανονισμούς και στα μεγέθη υπολογισμού.

Τα πολιτικά αεροδρόμια διακρίνονται ανάλογα με τη χρήση τους σε κανονικά και εναλλακτικά. Τα εναλλακτικά χρησιμοποιούνται σε έκτακτες μόνο

περιπτώσεις και κατά κανόνα είναι στρατιωτικά αεροδρόμια που μπορούν να εξυπηρετήσουν και την πολιτική αεροπορία.

Μια άλλη διάκριση των αεροδρομίων είναι σε χερσαία, υδάτινα και επί πάγου. Στο τέλος της δεκαετίας του 1930 υπήρχε ένα σημαντικό πλήθος υδάτινων αεροδρομίων, ωστόσο η μεγάλη εξέλιξη των αεριοθούμενων, μετά τον Β' Παγκόσμιο πόλεμο, απαξίωσε τις σχετικές εγκαταστάσεις. Έτσι σήμερα διατηρείται σε χρήση ένας πολύ περιορισμένος αριθμός τέτοιων αεροδρομίων. Αντίθετα τα αεροδρόμια σε πάγο που εξυπηρετούν τις πολικές περιοχές παρουσιάζουν σημαντική εξέλιξη, παρότι έχουν τη δυνατότητα να εξυπηρετήσουν αεροσκάφη μικρού βάρους.

2.3 – ΚΥΡΙΕΣ ΚΑΤΑΣΚΕΥΕΣ ΑΕΡΟΔΡΟΜΙΩΝ

Οι κύριες κατασκευές ενός αεροδρομίου, τα βασικά στοιχεία δηλαδή που καλύπτουν τις ανάγκες του είναι:

- Ο(οι) διάδρομος(οι) ή σύστημα διαδρόμων που εξυπηρετεί την προσγείωση και απογείωση των αεροσκαφών.
- Τα δάπεδα σταθμεύσεως, όπου τα αεροσκάφη σταθμεύουν για φορτοεκφόρτωση, ανεφοδιασμό κλπ
- Το τροχοδρομικό σύστημα που συνδέει το διάδρομο ή το σύστημα διαδρόμων με τα δάπεδα στάθμευσης
- Ο επιβατικός αεροσταθμός που εξυπηρετεί την διακίνηση των επιβατών μεταξύ των επιφανειακών δικτύων μεταφοράς και των αεροσκαφών
- Ο εμπορευματικός αεροσταθμός για τη διακίνηση εμπορευμάτων
- Οι συνδέσεις του(των) αεροσταθμών με τα επιφανειακά δίκτυα μεταφοράς και οι σχετικές τερματικές εγκαταστάσεις εξυπηρετήσεως των μέσων αυτών
- Ο πύργος ελέγχου
- Λοιπές εγκαταστάσεις που εξυπηρετούν την ασφάλεια γενικότερα (Πυροσβεστικός σταθμός κλπ), την διακίνηση των επιβατών, ταχυδρομείου και εμπορευμάτων, τον ανεφοδιασμό, ελέγχους, συντήρηση κλπ των αεροσκαφών καθώς και του υπόλοιπου μηχανικού εξοπλισμού του αεροδρομίου.

2.4 – ΚΩΔΙΚΟΣ ΑΝΑΦΟΡΑΣ ΑΕΡΟΔΡΟΜΙΟΥ

Εισαγωγική σημείωση: Σκοπός του κωδικού αναφοράς είναι η παροχή μιας απλής μεθόδου συσχέτισης των πολυάριθμων προδιαγραφών που αφορούν τα

χαρακτηριστικά ενός αεροδρομίου ώστε να δοθεί μια σειρά αερολιμένων, κατάλληλων για τα αντίστοιχα αεροσκάφη που πρόκειται να εξυπηρετήσουν. Ο κωδικός δεν προσφέρεται για καθορισμό μήκους διαδρόμων ή απαιτήσεων αντοχής δαπέδων και συντίθεται από δύο στοιχεία που σχετίζονται με τα χαρακτηριστικά επιδόσεων και τις διαστάσεις του αεροσκάφους. Το στοιχείο 1, είναι αριθμός σχετιζόμενος με το μήκος πεδίου αναφοράς αεροπλάνου και το στοιχείο 2 είναι γράμμα σχετιζόμενο με το άνοιγμα φτερών και την εξωτερική απόσταση των τροχών του κύριου εξοπλισμού αεροσκάφους. Το καθένα από τα στοιχεία συνδέεται με το αεροσκάφος του οποίου τα χαρακτηριστικά είναι κρίσιμα για τις εγκαταστάσεις. Στην εφαρμογή των οδηγιών του Annex 14, πρώτα επιλέγονται τα εξυπηρετούμενα αεροσκάφη και έπειτα αναγνωρίζεται ο κωδικός αριθμός του αεροδρομίου.

- Ο κωδικός αναφοράς αεροδρομίου (κωδικός αριθμός και γράμμα) ο οποίος επιλέγεται για τον σχεδιασμό του, θα πρέπει να καθορίζεται σύμφωνα με τα χαρακτηριστικά του αεροσκάφους το οποίο πρόκειται να εξυπηρετήσουν οι εγκαταστάσεις.
- Τα δύο στοιχεία του κωδικού έχουν την έννοια που δίδεται στον πίνακα 2.1

Σημειώσεις

- Ο καθορισμός του μήκους πεδίου αναφοράς είναι αποκλειστικά για την επιλογή κωδικού αριθμού (στοιχείο 1) και δεν επηρεάζει το πραγματικό μήκος διαδρόμου που παρέχεται
- Οδηγίες για την επιλογή κωδικού αναφοράς αεροδρομίου από την αρμόδια αρχή δίδονται στο “Aerodrome design manual, Parts 1 & 2”

Κωδικός αναφοράς αεροδρομίου

Στοιχείο 1 του κωδικού		Στοιχείο 2 του κωδικού		
Κωδικός αριθμός (1)	Μήκος πεδίου αναφοράς αεροπλάνου (2)	Κωδικό γράμμα (3)	Άνοιγμα φτερών (4)	Εξωτερικό άνοιγμα κύριου εξοπλισμού τροχών (5)
1	< 800 μ	A	< 15 μ	< 4,5 μ
2	[800 , 1200) μ	B	[15,24) μ	[4,5 , 6) μ
3	[1200 , 1800) μ	C	[24,36) μ	[6 , 9) μ
4	1800 μ και πάνω	D	[36,52) μ	[9 , 14) μ
		E	[52,65) μ	[9 , 14) μ
		F	[65,80) μ	[14 , 16) μ

(5) Απόσταση των εξωτερικών άκρων των κύριων τροχών

ΠΙΝΑΚΑΣ 2.1

Ορισμός: Μήκος πεδίου αναφοράς αεροπλάνου είναι το ελάχιστο μήκος πεδίου που απαιτείται για απογείωση με τη μέγιστη εγκεκριμένη μάζα απογείωσης, στο επίπεδο θαλάσσης, με πρότυπες ατμοσφαιρικές συνθήκες, άνευ ανέμου και μηδενική κλίση διαδρόμου, όπως παρουσιάζεται στο σχετικό εγχειρίδιο πτήσης του αεροσκάφους που καθορίζεται από την αρχή πιστοποίησης ή αντίστοιχα δεδομένα από τον κατασκευαστή του αεροσκάφους.

2.5 – ΑΕΡΟΣΚΑΦΗ

Στην πολιτική αεροπορία χρησιμοποιείται μια πολύ μεγάλη ποικιλία αεροσκαφών. Τα κυριότερα χαρακτηριστικά ενός τύπου που ενδιαφέρουν τον σχεδιασμό ενός αεροδρομίου είναι:

- Αριθμός κινητήρων
- Ωστική δύναμη ανά κινητήρα
- Μέγιστο βάρος απογείωσης
- Μέγιστο βάρος καυσίμων
- Ωφέλιμο φορτίο
- Ολικό μήκος
- Άνοιγμα φτερών
- Ολικό ύψος
- Ακτίνα ενεργείας με μέγιστο ωφέλιμο φορτίο

- Ύψος πτήσεως
- Ταχύτητα πτήσεως
- Χωρητικότητα αποθήκης
- Αριθμός επιβατών

Τα χαρακτηριστικά των διαφόρων τύπων αεροσκαφών επηρεάζουν τον σχεδιασμό των επιμέρους στοιχείων ενός αεροδρομίου. Για κάθε επιμέρους κατασκευή ενός αεροδρομίου αναγνωρίζεται ο τύπος που θα έχει τα δυσμενέστερα (πιο απαιτητικά στην ικανοποίηση) χαρακτηριστικά, εκείνα βεβαίως που επηρεάζουν την συγκεκριμένη κατασκευή, από όλους τους τύπους που προβλέπεται να χρησιμοποιήσουν την κατασκευή αυτή. Ο επιλεγόμενος, για τον σχεδιασμό μιας ορισμένης κατασκευής, τύπος αεροσκάφους ονομάζεται «αεροσκάφος σχεδιασμού» της κατασκευής αυτής. Οι κατασκευές ενός αεροδρομίου μπορεί να έχουν, σύμφωνα με τα παραπάνω, έναν αριθμό αεροσκαφών σχεδιασμού, γεγονός που είναι συνηθισμένο στα μεγάλα διεθνή αεροδρόμια. Εκεί το α/φ σχεδιασμού του οδοστρώματος ενός διαδρόμου είναι γενικά διάφορο από αυτό του μήκους του, ενώ ένα τρίτο α/φ επιλέγεται για το σχεδιασμό των υποστέγων συντήρησης, επισκευών κλπ. Για άλλα στοιχεία του αεροδρομίου όπως λχ τα δάπεδα σταθμεύσεως, είναι σκόπιμη η εισαγωγή της έννοιας του τύπου μίξεως. Ως μίξη θεωρείται η ποσοστιαία κατανομή του συνόλου των α/φ που χρησιμοποιούν το στοιχείο. Στα μικρά πολιτικά αεροδρόμια τώρα, και ιδιαίτερα τα ιδιωτικά μπορεί μόνο ένας τύπος αεροσκάφους να αρκεί για τον σχεδιασμό όλων των κατασκευών.

Στις Ηνωμένες πολιτείες ο κώδικας αναφοράς αεροδρομίου είναι ένας δείκτης δύο στοιχείων που σχετίζει τα πρότυπα που χρησιμοποιούνται στον σχεδιασμό ενός αεροδρομίου, με έναν συνδυασμό διαστάσεων και λειτουργικών χαρακτηριστικών του(των) αεροσκάφους(ών) σχεδιασμού. Το πρώτο στοιχείο (κατηγορία προσέγγισης αεροσκάφους) αντιστοιχεί στην ομαδοποίηση βάση ταχυτήτων κατά ICAO PANS-OPS ενώ το δεύτερο (κατηγορία σχεδίασης αεροσκάφους – airplane design group) στην κατηγοριοποίηση βάση ανοίγματος φτερών (βλ. πίνακα 2.1). Αντιπροσωπευτικά παραδείγματα αεροσκαφών ανά κλάση είναι τα παρακάτω:

ΚΛΑΣΗ	ΤΥΠΟΣ Α/Φ
A	Boeing 707, 747, 720 Douglas DC-8, DC-10 Lockheed L-1011
B	Boeing 727, 737, Douglas DC-9, BAC-111
C	Μικρά α/φ αερογραμμών(κινητήρες P)
D	Μικρά α/φ γενικής χρήσης (κινητήρες 2P)
E	Μικρά α/φ (κινητήρας 1P)

Airport Reference Code (ARC)		
Κατηγορία προσέγγισης αεροσκάφους	Αντίστοιχος κωδικός του Annex 14	Αεροδρόμιο σχεδιασμένο για
A	1	B747-400 ARC D-V
B	2	
C	3	
D	4	
E	—	
Κατηγορία σχεδίασης αεροσκάφους	Αντίστοιχο κωδικό γράμμα του Annex 14	
I	A	
II	B	
III	C	
IV	D	
V	E	
VI	F(προτεινόμενο)	

ΠΙΝΑΚΑΣ 2.2

2.6 – ΠΕΡΙΒΑΛΛΟΝΤΙΚΕΣ ΕΠΙΠΤΩΣΕΙΣ ΑΕΡΟΜΕΤΑΦΟΡΩΝ

Οι παγκόσμιες αερομεταφορές, ανεξάρτητα από την μακροχρόνια εντυπωσιακή ανοδική τους πορεία, αντιμετωπίζουν σήμερα σημαντικά προβλήματα, αλλά και περιβαλλοντικές επιπτώσεις, οι οποίες είναι καθοριστικές στην κοινωνική αποδοχή, η οποία καθορίζει με τη σειρά της την ανάπτυξη και την βιωσιμότητα σε συγκεκριμένη θέση.

Ειδικότερα όμως, οι αρνητικές περιβαλλοντικές επιπτώσεις του τομέα των αερομεταφορών, όπως η ηχητική ρύπανση και η ρύπανση του αέρα, έχουν φέρει στο προσκήνιο με ιδιαίτερη έμφαση, τη σημασία του κατάλληλου χωροταξικού κυρίως σχεδιασμού των αεροδρομίων και την αξιοποίηση των υποδομών. Ήδη, οι εκπομπές ρύπων από τα αεροσκάφη αντιστοιχούν στο 3% των συνολικών εκπομπών στην Ευρωπαϊκή Ένωση των 27 και στο 13% των εκπομπών που προέρχονται από τον τομέα των μεταφορών.

Σε σύγκριση με το αυτοκίνητο, το αεροπλάνο δεν εκπέμπει πολύ περισσότερο διοξείδιο του άνθρακα (CO₂) ανά μίλι, όμως η επίδρασή του στο περιβάλλον

είναι πολλαπλάσια γιατί εκπέμπει ρύπους ψηλά στην ατμόσφαιρα που σημαίνει μεγαλύτερη επίδραση στην αλλαγή του κλίματος, ενώ σε δεδομένο χρόνο μπορεί να διανύσει ασύγκριτα μεγαλύτερες αποστάσεις με αναλόγως χειρότερες επιπτώσεις για το περιβάλλον. Επίσης, τα αεροδρόμια επιβαρύνουν και έμμεσα την ατμόσφαιρα λόγω συγκέντρωσης παράπλευρων δραστηριοτήτων, (όπως τα ταξί, οχήματα ανεφοδιασμού των αεροσκαφών κ.λπ). Ως συνέπεια τα πολυσύχναστα αεροδρόμια κατατάσσονται μεταξύ των μεγαλύτερων παραγόντων ρυπάνσεως μαζί με τις βιομηχανίες παραγωγής χημικών και τα διυλιστήρια. Η εξεύρεση τρόπων συγκερασμού της περιβαλλοντικής σταθερότητας με την οικονομική ανάπτυξη των αεροδρομίων είναι μείζον πολιτικό θέμα. Η Ε. Ε. και παράγοντες του κλάδου ευρίσκονται σήμερα σε προωθημένες συζητήσεις πάνω σε αυτό το θέμα. Στα υπό συζήτηση μέτρα εξετάζεται τόσο πανευρωπαϊκά όσο και παγκόσμια και η συμμετοχή του κλάδου στο σύστημα εμπορίας ρύπων και η επιβολή ενός περιβαλλοντικού τέλους ανάλογα με το κόστος στο περιβάλλον. Το ζητούμενο είναι η εξεύρεση μιας υγιούς ισορροπίας ανάμεσα στο περιβαλλοντικό και οικονομικό κόστος.

Αυτό το περιβαλλοντικό κόστος που έπρεπε να καταβάλλουν οι εταιρείες προς το παρόν δεν έχει βρει πρακτική εφαρμογή. Η καύση της κηροζίνης από τις μηχανές των αεροπλάνων έχει σαν αποτέλεσμα τις εκπομπές ρυπαντών όπως CO₂, NO_x, SO₂, CH₄ κ.λπ. Τα αεροπλάνα είναι γνωστό ότι πετούν στην τροπόσφαιρα (ως 12 Km ύψος) και την στρατόσφαιρα (ως 50 Km). Στα ύψη αυτά οι ρυπαντές διαχέονται σε μεγαλύτερο ποσοστό στην ατμόσφαιρα της γης προκαλώντας σοβαρές επιπτώσεις. Σε παγκόσμια κλίμακα η ρύπανση που προξενούν οι αεροπορικές μεταφορές υπολογίζονται σε 3,5% για το φαινόμενο του θερμοκηπίου και 1,2% για τη μείωση της στοιβάδας του όζοντος. Τα ποσοστά αυτά φαινομενικά είναι μικρά, όμως οι ρυθμοί ανάπτυξης του τομέα αυτού σύντομα θα αναδείξουν τις αερομεταφορές σε μείζον παράγοντα ρύπανσης. Μέχρι το 2025 υπολογίζεται η συμμετοχή στο φαινόμενο του θερμοκηπίου θα είναι τουλάχιστον 8% και για την καταστροφή της στοιβάδας του όζοντος σε 6,45%

Οι τοπικές επιπτώσεις αεροδρομίου είναι σχεδόν πάντα αποτέλεσμα κακού (μη προνοητικού) σχεδιασμού. Θα μπορούσαμε επομένως να μιλάμε για επιπτώσεις που είναι δυνατόν να αποφευχθούν ή κατ' ελάχιστο να μετριασθούν όταν αναφερόμαστε σε :

- *Θόρυβος*: Το επίπεδο της στάθμης θορύβου στην γειτνιάζουσα περιοχή πρέπει να παραμένει εντός θεσμοθετημένων ορίων (ανάλογα αν η περιοχή είναι κατοικημένη ή όχι) ειδικά η υπέρβαση είναι πολλές φορές μη αναστρέψιμη

και μπορεί να συνηγορήσει (πχ στις περιπτώσεις οξείας ηχορρύπανσης) μέχρι και στην απομάκρυνση του αερολιμένα από την περιοχή.

- *Υγρά απόβλητα*: Απαιτούνται εγκαταστάσεις επεξεργασίας για τα αστικά λύματα από τα κτίρια του αεροδρομίου και τα απόβλητα των αεροσκαφών από τις χημικές τους τουαλέτες, ειδικά η κοντινή (ή σχετιζόμενη) θαλάσσια περιοχή διατρέχει κίνδυνο μόλυνσης. Σε περίπτωση ύπαρξης κοντινού δικτύου με βιολογικό καθαρισμό, προτείνεται η επέκτασή του ως την περιοχή του αερολιμένα.

- *Στερεά απόβλητα*: Μπορούν να αποτελέσουν πρόβλημα όταν δεν προβλέπεται παρακείμενος ΧΥΤΑ ικανός να καλύψει τη απόθεσή τους. Χαρακτηριστικά να αναφέρουμε πως στο αεροδρόμιο Ν. Καζαντζάκης Κρήτης, έναν αερολιμένα μικρού μεγέθους, η ποσότητα των παραγομένων απορριμμάτων έφθανε σε 1800 τόνους ετησίως, 12 περίπου τόνοι ανά ημέρα αιχμής. Η προέλευση των στερεών αποβλήτων είχε την εξής κατανομή: 1706 τόνοι από τους επιβάτες, 10 τόνοι ανά έτος από τους επισκέπτες και 89 τόνοι από τους εργαζόμενους. Πρόσθετο πρόβλημα, ήταν και η αποκομιδή των απορριμμάτων η οποία αποτελούσε σημείο τριβής για τους Δήμους Ηρακλείου και Αλικαρνασσοῦ

- *Αέρια απόβλητα*

- *Συγκοινωνία*: Η έλλειψη κατάλληλου συγκοινωνιακού σχεδιασμού οδηγεί σε κυκλοφοριακή ασφυξία τους δρόμους που συνδέουν το αεροδρόμιο με το επίγειο δίκτυο. Η μελέτη κατάλληλων συνδετήριων οδών είναι ζωτικής σημασίας για την ομαλή λειτουργία του αεροδρομίου.

- *Αισθητική*: Η μελέτη για ομαλή ένταξη των εγκαταστάσεων αερολιμένα στο τοπίο περιοχής και οι εξωραϊστικές παρεμβάσεις στα κτίριά του αποτελούν αναγκαίες προϋποθέσεις για να αποφευχθεί η υποβάθμιση του αισθητικού επιπέδου μιας περιοχής.

- *Χρήσεις γης*: Ένα αεροδρόμιο δεσμεύει τεράστιες εκτάσεις γύρω από αυτό, που δεν μπορούν να χρησιμοποιηθούν για γενική κατοικία. Η «σπατάλη» αυτή μπορεί να μετριασθεί με χρήση του χώρου για εμπορικές αποθήκες, χονδρεμπόριο, βιομηχανίες, βιοτεχνίες.

- *Επικινδυνότητα*: Η μη τήρηση των κανονισμών ασφαλείας κατά τις προσγειαυογειώσεις αεροσκαφών και η πτήση σε χαμηλό υψόμετρο στην περιοχή εγγύτητας του αερολιμένα αυξάνουν τον κίνδυνο στον οποίο εκτίθενται οι κάτοικοι των γύρω περιοχών. Για παράδειγμα η δημιουργία ωστικού κύματος διαφέρει ανάλογα με την ισχύ των μηχανών ενός αεροσκάφους, επομένως απαιτούνται και διαφορετικές αποστάσεις ασφαλείας, για την περίπτωση εκτόξευσης υλικών (πέτρες, λαμαρίνες κα) ή και την πρόκληση καταστροφών και κινδύνων στο γύρω περιβάλλον (ένα Μπόινγκ 747 κατά την απογείωση μπορεί να προκαλέσει ωστικό κύμα που σπάει δέντρα, τα οποία παρασύρονται με επικίνδυνες ταχύτητες).

Όπως προκύπτει από τα προαναφερόμενα, βασικός παράγοντας για την αποφυγή επιπτώσεων από την κατασκευή και λειτουργία αερολιμένων είναι "η θέση του στο χώρο". Αυτό αποτελεί βασική αρχή του Συντάγματος (αρ. 24. παρ. 2) σύμφωνα με το οποίο:

"Η χωροταξική αναδιάρθρωση της χώρας, η διαμόρφωση, η ανάπτυξη, η πολεοδόμηση και η επέκταση των πόλεων και οικιστικών γενικά περιοχών υπάγεται στη ρυθμιστική αρμοδιότητα και τον έλεγχο του Κράτους, με σκοπό να εξυπηρετείται η λειτουργικότητα και η ανάπτυξη των οικισμών και να εξασφαλίζονται οι καλύτεροι δυνατοί όροι διαβίωσης. Οι σχετικές τεχνικές επιλογές και σταθμίσεις γίνονται κατά τους κανόνες της επιστήμης".

Η αντιμετώπιση των σοβαρών επιπτώσεων σε οποιαδήποτε αεροδρόμιο, βρίσκεται στην εξεύρεση τρόπων συγκερασμού της περιβαλλοντικής σταθερότητας και της οικονομικής ανάπτυξης της περιοχής. Επομένως ο σχεδιασμός ενός "Νέου Αεροδρομίου", πρέπει να είναι πολύ υψηλών προδιαγραφών σε ότι αφορά στην προσβασιμότητα, τη λειτουργικότητα και το καλό αισθητικό αποτέλεσμα, δηλαδή την αρμονική ένταξη του αεροδρομίου στο ιστορικό, πολιτιστικό και φυσικό περιβάλλον μιας περιοχής.

ΚΕΦΑΛΑΙΟ 3

ΠΕΡΙΟΡΙΣΜΟΣ ΚΑΙ ΑΠΟΜΑΚΡΥΝΣΗ ΕΜΠΟΔΙΩΝ

Ο σκοπός των προδιαγραφών που περιγράφονται σε αυτό το κεφάλαιο είναι να καθορίσουν τον εναέριο χώρο γύρω από τα αεροδρόμια που πρέπει να διατηρηθεί ελεύθερος εμποδίων, έτσι ώστε να επιτραπεί να εκτελούνται με ασφάλεια οι επιθυμητές λειτουργίες των αεροσκαφών και να αποτρέψουν τα αεροδρόμια από το να γίνουν μη χρηστικά εξαιτίας της ανάπτυξης εμποδίων γύρω τους. Αυτό επιτυγχάνεται εγκαθιστώντας μια σειρά από επιφάνειες που θέτουν τα όρια πέραν των οποίων δεν επιτρέπεται να προεξέχουν αντικείμενα προς τον εναέριο χώρο. Οι επιφάνειες αυτές ονομάζονται περιορισμού εμποδίων και καθορίζονται με βάση τους συνήθεις και ακραίους ελιγμούς των διαφόρων τύπων αεροσκαφών που θα εξυπηρετήσει το αεροδρόμιο.

Η μορφή των επιφανειών αυτών είναι προκαθορισμένη, ωστόσο οι διαστάσεις και η διάταξή τους καθορίζονται

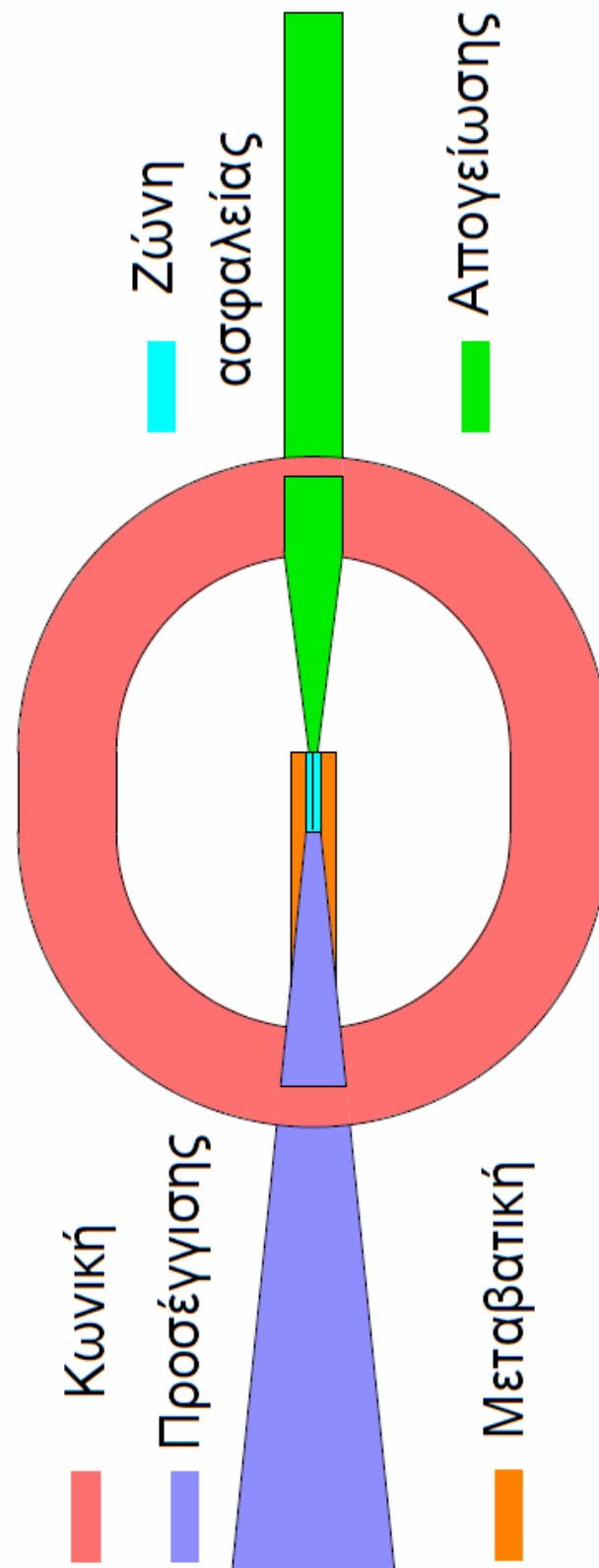
i. από την προτιθέμενη χρήση του διαδρόμου, δηλαδή απ' το αν ο διάδρομος προορίζεται για προσγείωση ή απογείωση. Σε περιπτώσεις που εξυπηρετεί και τις δύο λειτουργίες ανά κατεύθυνση, η απομάκρυνση εμποδίων (καθορισμός επιτρεπτών υψομέτρων) προκύπτει από υπέρθεση των επιμέρους επιφανειών, των επιφανειών που προκύπτουν δηλαδή για κάθε λειτουργία ξεχωριστά, και λήψη των δυσμενέστερων (μικρότερων) υψομέτρων. Επίσης ανάλογα με τον τύπο προσέγγισης (ενόργανο ή μη) έχουμε διαφοροποιήσεις στις επιφάνειες.

ii. Από τον κωδικό αριθμό (τάξη) του αεροδρομίου (1,2,3 ή 4)

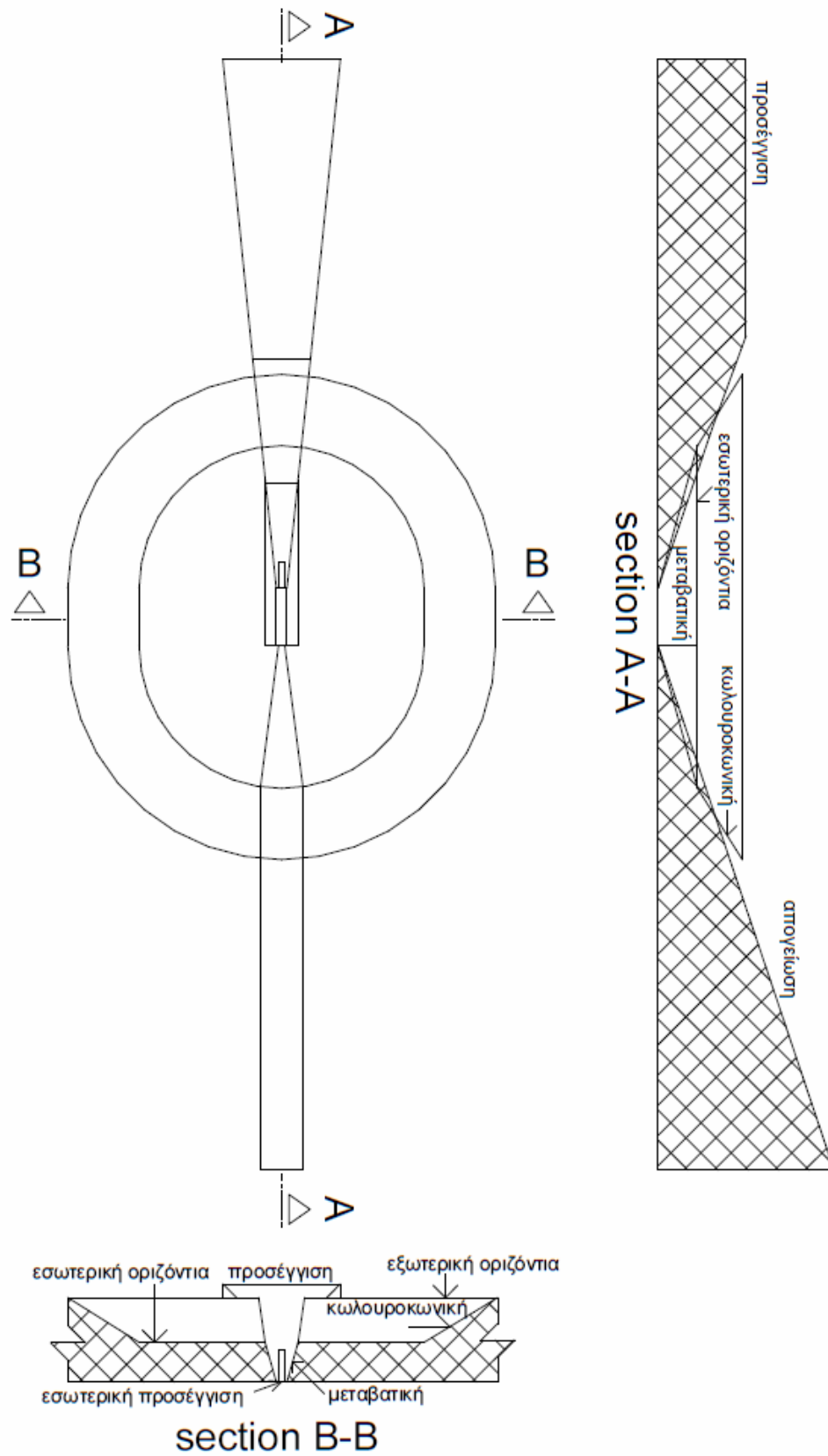
Κατά τους κανονισμούς του ICAO που ακολουθεί και η Ελλάδα, οι επιφάνειες δίνονται στο σχήμα 1 για όλες τις τάξεις αεροδρομίων ενώ επεξηγούνται με τις σχετικές τους τομές στο σχήμα 2 και σε προοπτικό στο σχήμα 3.

Αρχικά καθορίζεται μια ελεύθερη ζώνη ασφαλείας που περιβάλλει το διάδρομο (βλ. κεφ. «ΓΕΩΜΕΤΡΙΚΗ ΜΕΛΕΤΗ ΑΕΡΟΔΡΟΜΙΩΝ») και από τα όρια αυτής ξεκινούν οι επιφάνειες περιορισμού εμποδίων, τα χαρακτηριστικά των οποίων αναλύονται παρακάτω. Από την περιβάλλουσα των επιφανειών αυτών προκύπτει η επιβαλλόμενη απομάκρυνση εμποδίων. Επειδή ο επιθυμητός περιορισμός εμποδίων είναι ιδιαίτερα αυστηρός και πολλές φορές πρακτικά ή οικονομικά ανέφικτος, οι αρχές που διοικούν το αεροδρόμιο και οι αντίστοιχες κρατικές αρχές έχουν τη δυνατότητα, για ορισμένες υποπεριοχές του χώρου που ορίζεται από την εν λόγω περιβάλλουσα επιφάνεια, να επιτρέπουν την προβολή (ύπαρξη) εμποδίων με πρόβλεψη κατάλληλης επισήμανσης. Πρέπει επομένως να εκλάβουμε τις περιγραφές του κεφαλαίου αυτού ως καθορίζουσες τις επιφάνειες που καταρχήν είναι σκόπιμο να μην διαπερνώνται από εμπόδια, αλλά όπως λέει και ο όρος «περιορισμού εμποδίων» αν σε κάποιες θέσεις αυτό δεν μπορεί να επιτευχθεί, τα εμπόδια πρέπει να είναι αποδεκτά από τις αρμόδιες αρχές και επιπλέον να επισημαίνονται

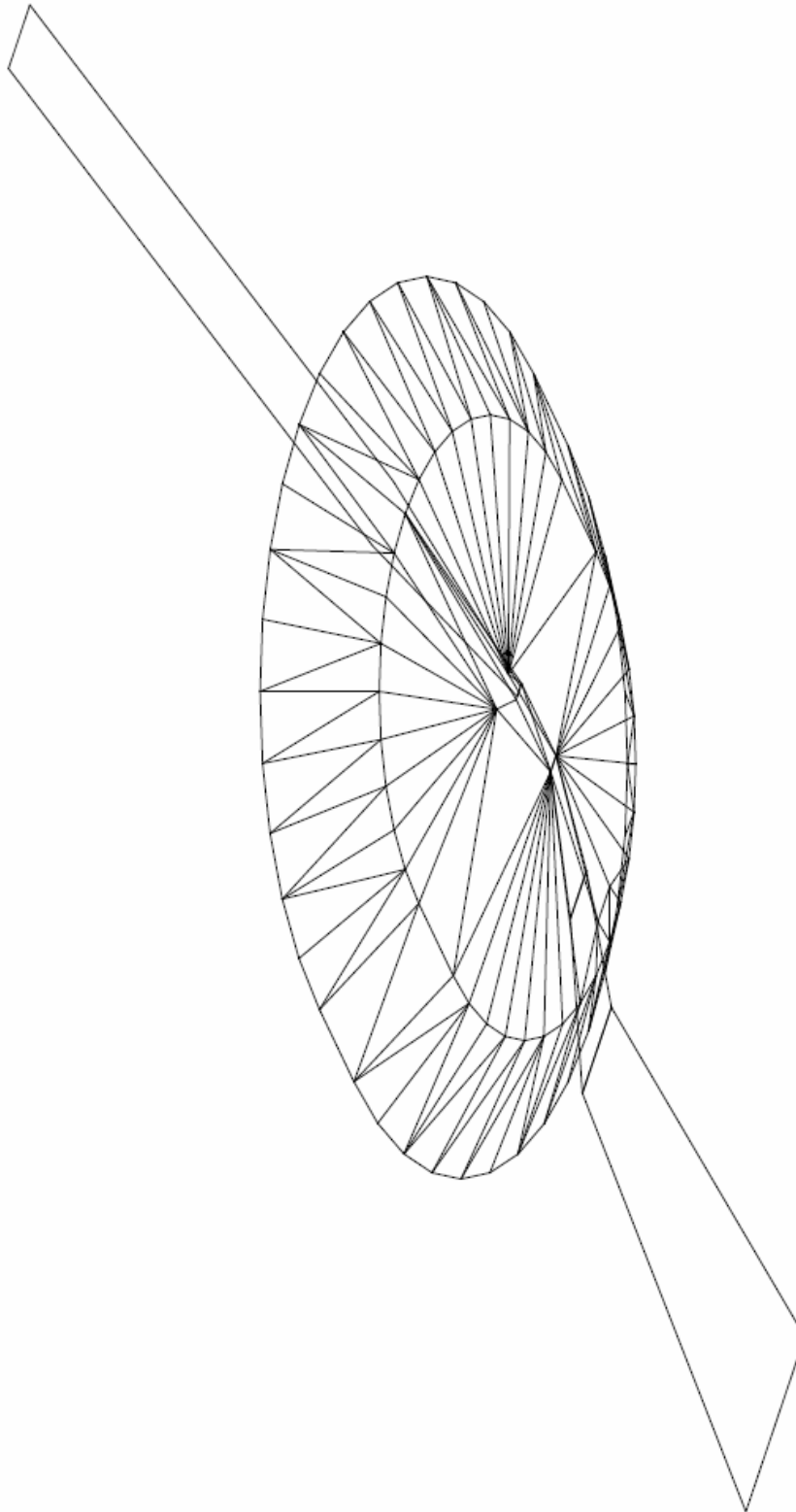
ΕΠΙΦΑΝΕΙΕΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ



ΣΧΗΜΑ 3.1



ΣΧΗΜΑ 3.2



ΣΧΗΜΑ 3.3

3.1– ΕΠΙΦΑΝΕΙΕΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ

I. ΕΣΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ ΕΠΙΦΑΝΕΙΑ (Inner horizontal surface)

Περιγραφή: Επιφάνεια τοποθετημένη σε οριζόντιο επίπεδο πάνω από το αεροδρόμιο και τα περίχωρά του.

Χαρακτηριστικά: Η ακτίνα ή τα εξωτερικά όρια μετρώνται από σημείο ή σημεία αναφοράς εγκατεστημένα για το σκοπό αυτό. Στην πρώτη περίπτωση (ένα σημείο αναφοράς) η επιφάνεια έχει κυκλικό σχήμα ενώ στη δεύτερη αποκτά «αμφιθεατρικό» όπως εικονίζεται στο σχήμα 3 για έναν και δύο διαδρόμους. Οι διαστάσεις (ακτίνα - ύψος) ανά κατηγορία αεροδρομίου παρατίθενται στον πίνακα 1 στο τέλος του κεφαλαίου.

Ύψος: Μετράται πάνω από επιφάνεια αναφοράς που ορίζεται γι' αυτό τον σκοπό (συνήθως διερχόμενη εκ του κέντρου του αντίστοιχου διαδρόμου).

II. ΚΩΝΙΚΗ ΕΠΙΦΑΝΕΙΑ (Conical surface)

Περιγραφή: Επιφάνεια κεκλιμένη προς τα πάνω και έξω σχετικά με την περιφέρεια της εσωτερικής οριζόντιας επιφάνειας.

Χαρακτηριστικά: Τα όρια της κωνικής επιφάνειας περιλαμβάνουν

- Ένα κάτω άκρο που συμπίπτει με την περιφέρεια της εσωτερικής οριζόντιας επιφάνειας
- Ένα άνω άκρο (αποτελούμενο από καμπύλες ομόκεντρες του κάτω) τοποθετούμενο σε συγκεκριμένο ύψος πάνω από την εσωτερική οριζόντια επιφάνεια.

Επομένως το σχήμα της κωνικής ακολουθεί το σχήμα της εσωτερικής οριζόντιας επιφάνειας. Το ύψος της (που καθορίζει και το άνω άκρο της) ανά κατηγορία αεροδρομίου παρατίθεται στον πίνακα 1 στο τέλος του κεφαλαίου.

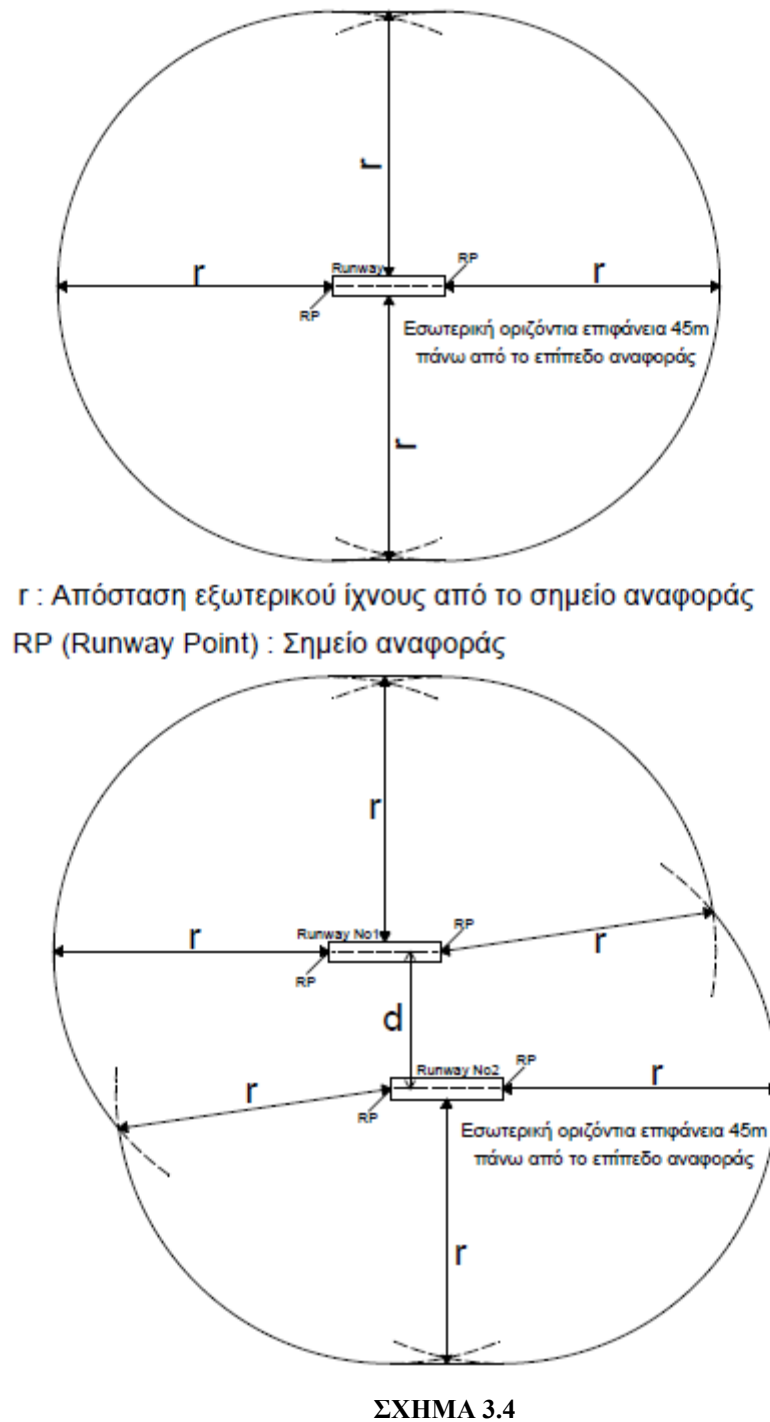
Κλίση: Μετράται σε κατακόρυφο επίπεδο, κάθετο στην περιφέρεια της εσωτερικής οριζόντιας επιφάνειας και είναι ίση με 5%.

III. ΕΞΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ ΕΠΙΦΑΝΕΙΑ (Outer horizontal surface)

Περιγραφή: Επιφάνεια τοποθετημένη σε οριζόντιο επίπεδο πάνω από το αεροδρόμιο και τα περίχωρά του.

Χαρακτηριστικά: Εκτείνεται πέρα από το εσωτερικό της όριο που συμπίπτει με το άνω άκρο της κωνικής επιφάνειας, επομένως το σχήμα της ακολουθεί το σχήμα της κωνικής και της εσωτερικής οριζόντιας επιφάνειας.

Ύψος: Συμπίπτει με το ύψος του άνω άκρου της κωνικής επιφάνειας.



IV. ΕΠΙΦΑΝΕΙΑ ΠΡΟΣΕΓΓΙΣΗΣ (Approach surface)

Περιγραφή: Κεκλιμένο επίπεδο ή συνδυασμός επιπέδων που προηγούνται του κατωφλίου.

Χαρακτηριστικά: Τα όρια της επιφάνειας προσέγγισης περιλαμβάνουν

- a. Μία εσωτερική πλευρά καθορισμένου μήκους, κάθετη στην προέκταση του άξονα του διαδρόμου και τοποθετημένη σε καθορισμένη απόσταση από το κατώφλι.
- b. Δύο πλευρές εκ των άκρων της εσωτερικής πλευράς που αποκλίνουν ομοιόμορφα και σε καθορισμένο βαθμό από την κεντρική γραμμή του διαδρόμου.
- c. Μία εξωτερική πλευρά παράλληλη στην εσωτερική ενώ
- d. Οι άνωθεν επιφάνειες θα ποικίλουν όταν χρησιμοποιούνται πλευρικά μετατοπισμένες, μετατοπισμένες ή καμπύλες προσεγγίσεις. Ειδικότερα θα «ακολουθούν» δύο πλευρές εκ των άκρων της εσωτερικής πλευράς και αποκλίνουσες ομοιόμορφα και σε καθορισμένο βαθμό από τον εκτεταμένο ή καμπυλωμένο άξονα του διαδρόμου.

Ύψος εσωτερικής πλευράς: Ίσο με το ύψος του μέσου του κατωφλίου

Κλίση: Οι κλίσεις μετρώνται στο κατακόρυφο επίπεδο που περιέχει τον άξονα του διαδρόμου (τομή ΑΑ)

V. ΕΣΩΤΕΡΙΚΗ ΕΠΙΦΑΝΕΙΑ ΠΡΟΣΕΓΓΙΣΗΣ (Inner Approach surface)

Περιγραφή: Ένα ορθογώνιο τμήμα της επιφάνειας προσέγγισης που προηγείται του κατωφλίου (βλ. σχήμα 5)

Χαρακτηριστικά: Τα όρια της επιφάνειας περιλαμβάνουν

- a. Μια εσωτερική πλευρά συμπίπτουσα με την εσωτερική πλευρά της επιφάνειας προσέγγισης αλλά με το δικό της καθορισμένο μήκος
- b. Δύο πλευρές εκ των άκρων της εσωτερικής και εκτεινόμενες παράλληλα στο κατακόρυφο επίπεδο που περιέχει τον άξονα του διαδρόμου
- c. Ένα εξωτερικό άκρο παράλληλο στο εσωτερικό

VI. ΜΕΤΑΒΑΤΙΚΗ ΕΠΙΦΑΝΕΙΑ (Transitional surface)

Περιγραφή: Σύνθετη επιφάνεια κατά μήκος της πλευράς της ζώνης ασφαλείας και μέρους της επιφάνειας προσέγγισης, η οποία κλίνει προς τα πάνω και έξω μέχρι την εσωτερική οριζόντια επιφάνεια.

Χαρακτηριστικά: Τα όρια της επιφάνειας περιλαμβάνουν

- a. Ένα κάτω άκρο που ξεκινάει στην τομή της πλευράς της επιφάνειας προσέγγισης με την εσωτερική οριζόντια επιφάνεια και εκτείνεται καθοδικά στην πλευρά της επιφάνειας προσέγγισης μέχρι την εσωτερική πλευρά της και από εκεί κατά μήκος της πλευράς της ζώνης ασφαλείας
- b. Ένα άνω άκρο ευρισκόμενο στο επίπεδο της εσωτερικής οριζόντιας επιφάνειας

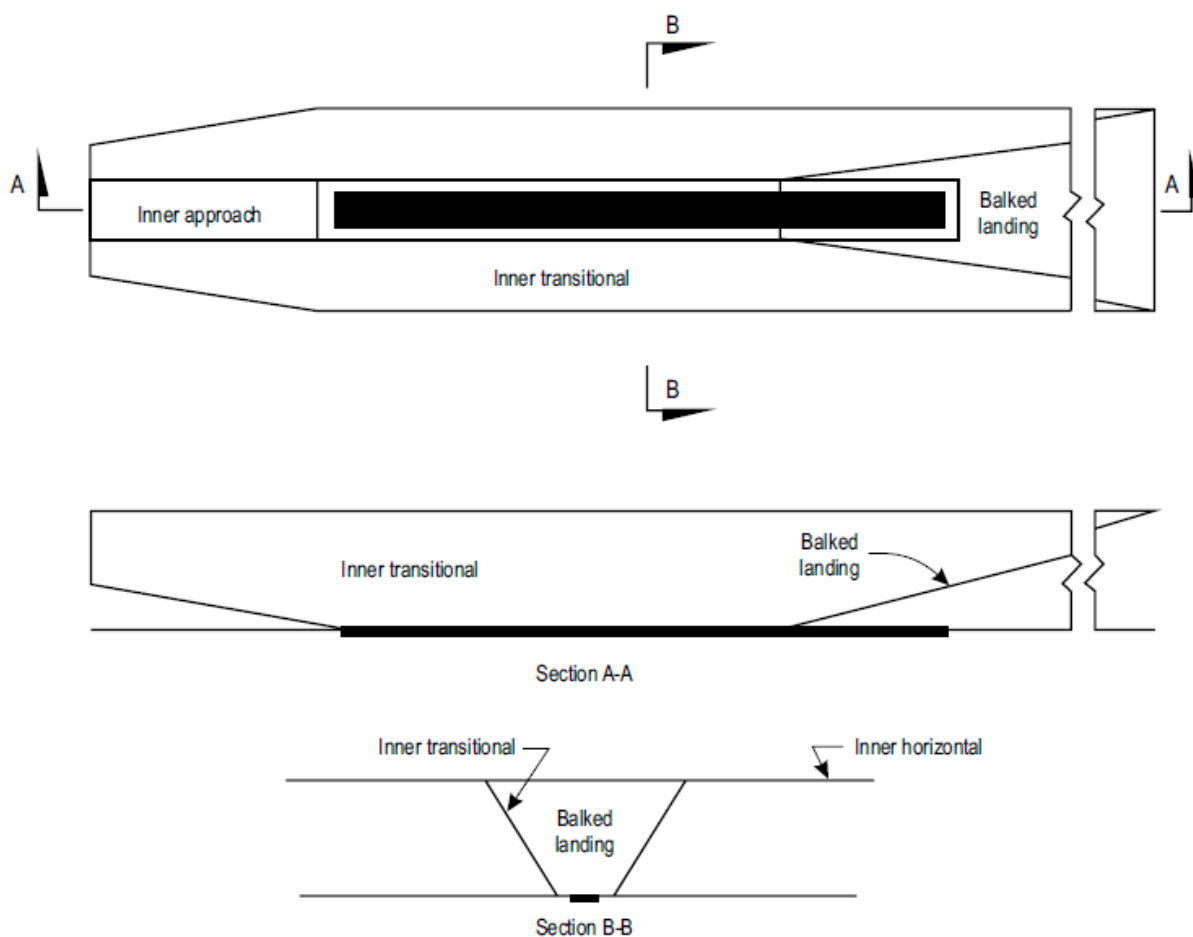
Ανύψωση σημείου χαμηλότερου άκρου: Κατά μήκος της πλευράς της επιφάνειας προσέγγισης ίση με την ανύψωση της επιφάνειας προσέγγισης ενώ κατά μήκος της πλευράς της ζώνης ασφαλείας ίση με την ανύψωση του σημείου που

βρίσκεται εγγύτερα στον άξονα του διαδρόμου (ή την προέκτασή του)

Παρατήρηση – Ως αποτέλεσμα των παραπάνω η μεταβατική επιφάνεια θα καμπυλώνεται κατά μήκος της ζώνης ασφαλείας αν το προφίλ του διαδρόμου είναι καμπυλωμένο ή θα είναι επίπεδη αν το προφίλ του διαδρόμου είναι ευθεία γραμμή. Αντίστοιχα θα καμπυλώνεται ή όχι (ανάλογα με το προφίλ του διαδρόμου) και η τομή της μεταβατικής με την εσωτερική οριζόντια επιφάνεια.

Κλίση: Μετράται σε κατακόρυφο επίπεδο, σε δεξιές γωνίες από την κεντρική γραμμή του διαδρόμου

VII. ΕΣΩΤΕΡΙΚΗ ΜΕΤΑΒΑΤΙΚΗ ΕΠΙΦΑΝΕΙΑ (Inner Transitional surface)



ΣΧΗΜΑ 3.5

Σημείωση: Σκοπός της εσωτερικής μεταβατικής επιφάνειας είναι να λειτουργήσει ως η ελεγκτική επιφάνεια περιορισμού των επιφανειακών εμποδίων που σχετίζονται με ραδιοβοηθήματα για την προσγείωση, αεροσκάφη και άλλα οχήματα τα οποία πρέπει να είναι κοντά στον διάδρομο και δεν πρέπει να διαπερνάται παρά μόνο από εύθραυστα αντικείμενα. Η μεταβατική επιφάνεια, όπως περιγράφεται στο VI, παραμένει η επιφάνεια ελέγχου για κτίρια κλπ.

Περιγραφή: Επιφάνεια όμοια με τη μεταβατική αλλά πιο κοντά στο διάδρομο (βλ. σχήμα 5).

Χαρακτηριστικά: Τα όρια της επιφάνειας περιλαμβάνουν

- a. Ένα κάτω άκρο εκ του τέλους της εσωτερικής επιφάνειας προσέγγισης και εκτεινόμενο καθοδικά στην πλευρά της επιφάνειας αυτής ως την εσωτερική πλευρά της, από εκεί (κατά μήκος της ζώνης) παράλληλα στον άξονα του διαδρόμου ως την εσωτερική πλευρά της επιφάνειας ματαιούμενης προσγείωσης (balked landing surface) και από εκεί ανοδικά στην πλευρά της επιφάνειας αυτής μέχρι να τμηθεί η εσωτερική οριζόντια επιφάνεια.
- b. Ένα άνω άκρο ευρισκόμενο στο επίπεδο της εσωτερικής οριζόντιας επιφάνειας.

Ανύψωση σημείου χαμηλότερου άκρου: Θα είναι

- a. Κατά μήκος της εσωτερικής επιφάνειας προσέγγισης και της επιφάνειας απογείωσης μετά από αποτυχημένη προσέγγιση (balked landing) ίση με την ανύψωση της συγκεκριμένης επιφάνειας στο σημείο αυτό
- b. Κατά μήκος της ζώνης ίση με την ανύψωση του κοντινότερου στον άξονα του διαδρόμου ή την προέκτασή του σημείου.

Παρατήρηση – Ως αποτέλεσμα των παραπάνω η εσωτερική μεταβατική επιφάνεια θα καμπυλώνεται κατά μήκος της ζώνης, αν το προφίλ του διαδρόμου είναι καμπυλωμένο ή θα είναι επίπεδη αν το προφίλ του διαδρόμου είναι ευθεία γραμμή. Αντίστοιχα θα καμπυλώνεται ή όχι (ανάλογα με το προφίλ του διαδρόμου) και η τομή της εσωτερικής μεταβατικής με την εσωτερική οριζόντια επιφάνεια.

Κλίση: Μετράται σε κατακόρυφο επίπεδο σε δεξιές γωνίες από την κεντρική γραμμή του διαδρόμου.

VIII. ΕΠΙΦΑΝΕΙΑ ΜΑΤΑΙΟΥΜΕΝΗΣ ΠΡΟΣΓΕΙΩΣΗΣ (Balked landing surface)

Περιγραφή: Κεκλιμένο επίπεδο που βρίσκεται σε καθορισμένη απόσταση από το κατώφλι και εκτείνεται ανάμεσα στις εσωτερικές μεταβατικές επιφάνειες (βλ. σχήμα 5). Αποτελεί την επιφάνεια ελέγχου εμποδίων για απογείωση μετά από

αποτυχημένη προσέγγιση.

Χαρακτηριστικά: Τα όρια της επιφάνειας περιλαμβάνουν (βλ. σχήμα 5):

- a. Ένα κατώτερο οριζόντιο ίχνος (κάθετο στον άξονα του διαδρόμου) στο υψόμετρο του άξονα του διαδρόμου, καθοριζόμενου μήκους, σε καθοριζόμενη απόσταση μετά (κατά τη φορά κίνησης) το κατώφλι του διαδρόμου.
- b. Ένα κεκλιμένο επίπεδο που αρχίζει από το ως άνω ίχνος με καθορισμένη κλίση, το οποίο περατώνεται στην εσωτερική οριζόντια επιφάνεια.
- c. Δύο πλευρικά επίπεδα που αρχίζουν από το ως άνω (κατώτερο) ίχνος και ορίζονται από ίχνη των οποίων η οριζόντια προβολή έχει ως προς τον άξονα του διαδρόμου καθορισμένη κλίση.

Ανύψωση σημείου χαμηλότερου άκρου: ίση με την ανύψωση της κεντρικής γραμμής του διαδρόμου στη θέση του εσωτερικού άκρου.

Κλίση: για την επιφάνεια αυτή μετράται σε κατακόρυφο επίπεδο που περιέχει την κεντρική γραμμή του διαδρόμου.

ΙΧ. ΕΠΙΦΑΝΕΙΑ ΑΠΟΓΕΙΩΣΗΣ (Take-off climb surface)

Περιγραφή: Κεκλιμένο επίπεδο ή άλλη καθορισμένη επιφάνεια πέραν του τέλους ενός διαδρόμου ή ελεύθερου καθορισμένου χώρου (clearway), βλ. σχήμα 1 και σχήμα 2-τομή Α-Α.

Χαρακτηριστικά: Τα όρια της επιφάνειας περιλαμβάνουν:

- a. Ένα εσωτερικό άκρο, επί οριζοντίου επιπέδου και κάθετο στην κεντρική γραμμή του διαδρόμου, ευρισκόμενο είτε σε καθορισμένη απόσταση πέραν του τέλους του διαδρόμου είτε στο τέλος του ελεύθερου καθορισμένου χώρου (clearway), όταν παρέχεται τέτοιος και το μήκος του ξεπερνά την καθορισμένη απόσταση.
- b. Δύο πλευρές εκ των άκρων της εσωτερικής, που αποκλίνουν ομοιόμορφα σε καθορισμένο βαθμό από την πίστα απογείωσης, μέχρις ενός καθορισμένου τελικού εύρους και συνεχίζουν έπειτα με το ίδιο πλάτος για το υπόλοιπο του μήκους της επιφάνειας απογείωσης.
- c. Ένα εξωτερικό άκρο, κάθετο στην καθορισμένη πίστα απογείωσης.

Ανύψωση σημείου χαμηλότερου άκρου: Ίση με το υψηλότερο σημείο της εκτεινόμενης κεντρικής γραμμής του διαδρόμου, ανάμεσα στο τέλος του διαδρόμου και στο εσωτερικό άκρο, εκτός και αν προνοείται ο ελεύθερος χώρος (clearway), οπότε και είναι ίση με το υψηλότερο σημείο του εδάφους στην κεντρική γραμμή του ελεύθερου αυτού χώρου.

Κλίση: Στην περίπτωση πτήσης με απευθείας απογείωση μετράται σε κατακόρυφο επίπεδο που περιέχει την κεντρική γραμμή του διαδρόμου. Στην περίπτωση πτήσης με απογείωση που περιλαμβάνει στροφή, η επιφάνεια

απογείωσης είναι μια σύνθετη επιφάνεια με μια κανονική κεντρική γραμμή και η κλίση της πρέπει να είναι ίση με αυτή για πτήση με απ' ευθείας απογείωση.

3.2 - ΑΠΑΙΤΗΣΕΙΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ

Σημείωση – Οι απαιτήσεις για τις επιφάνειες περιορισμού εμποδίων καθορίζονται βάσει της προτιθέμενης χρήσης ενός διαδρόμου (πχ για απογείωση ή προσγείωση) και του τύπου προσέγγισης και εφαρμόζονται όταν ο διάδρομος χρησιμοποιείται κατά τον οριζόμενο τρόπο. Σε περιπτώσεις που οι λειτουργίες γίνονται από ή προς και τις δύο κατευθύνσεις ενός διαδρόμου, ορισμένες επιφάνειες μπορεί να ακυρωθούν εξαιτίας αυστηρότερων απαιτήσεων από άλλες χαμηλότερες επιφάνειες.

I. ΜΗ ΕΝΟΡΓΑΝΟΙ ΔΙΑΔΡΟΜΟΙ ΠΡΟΣΕΓΓΙΣΗΣ (Non-Instrument approach runways)

Οι ακόλουθες επιφάνειες περιορισμού εμποδίων θα εγκαθίστανται για έναν μη ενόργανο διάδρομο προσγείωσης:

- Κωνική επιφάνεια
- Εσωτερική οριζόντια επιφάνεια
- Επιφάνεια προσέγγισης
- Μεταβατικές επιφάνειες

Κλίσεις: Τα ύψη και οι κλίσεις των επιφανειών δεν πρέπει να είναι μεγαλύτερες, και οι υπόλοιπες διαστάσεις μικρότερες, από τις τιμές που ορίζονται στον πίνακα 1 στο τέλος του κεφαλαίου.

Νέα αντικείμενα ή επεκτάσεις ήδη υφιστάμενων αντικειμένων δεν πρέπει να επιτρέπονται πάνω από την επιφάνεια προσέγγισης ή τη μεταβατική επιφάνεια εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το καινούριο αντικείμενο ή η επέκταση θα καλυπτόταν από ένα ήδη υπάρχον αμετακίνητο αντικείμενο.

Σύσταση – Νέα αντικείμενα ή επεκτάσεις ήδη υφιστάμενων αντικειμένων δεν πρέπει να επιτρέπονται πάνω από την κωνική επιφάνεια ή την εσωτερική οριζόντια επιφάνεια εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το καινούριο αντικείμενο ή η επέκταση θα καλυπτόταν από ένα ήδη υπάρχον αμετακίνητο αντικείμενο ή μετά από αεροναυτική μελέτη διαπιστωθεί ότι το αντικείμενο ή η προέκταση δεν θα επηρεάσει αρνητικά την ασφάλεια και την ομαλότητα των χειρισμών των αεροσκαφών.

Σύσταση – Υπάρχοντα αντικείμενα πάνω από οποιαδήποτε επιφάνεια πρέπει,

όσο είναι εφικτό, να απομακρύνονται εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το αντικείμενο καλύπτεται από ένα ήδη υπάρχον αμετακίνητο αντικείμενο ή μετά από αεροναυτική μελέτη διαπιστωθεί ότι το αντικείμενο δεν θα επηρεάσει αρνητικά την ασφάλεια και την ομαλότητα των χειρισμών των αεροσκαφών.

Σημείωση – Εξαιτίας εγκάρσιων ή κατά μήκος κλίσεων σε μια ζώνη ασφαλείας, σε ορισμένες περιπτώσεις, το εσωτερικό άκρο ή τμήμα του εσωτερικού άκρου της επιφάνειας προσέγγισης μπορεί να βρίσκεται κάτω από το αντίστοιχο ύψος της ζώνης. Δεν είναι απαραίτητο η ζώνη να διαμορφώνεται ώστε να συμφωνεί με το εσωτερικό άκρο της επιφάνειας προσέγγισης, ούτε τα αντικείμενα που βρίσκονται πάνω από την επιφάνεια προσέγγισης μετά το τέλος της ζώνης, αλλά κάτω από το επίπεδό της, να απομακρύνονται, εκτός αν θεωρηθεί ότι θέτουν σε κίνδυνο το αεροσκάφος.

Σύσταση – Στην αξιολόγηση προτεινόμενων κατασκευών πρέπει να λαμβάνεται υπόψη η πιθανή μελλοντική ανάπτυξη ενόργανου διαδρόμου, με συνέπεια την απαίτηση αυστηρότερων οριακών επιφανειών.

II. ΜΗ ΑΚΡΙΒΕΙΣ ΕΝΟΡΓΑΝΟΙ ΔΙΑΔΡΟΜΟΙ ΠΡΟΣΕΓΓΙΣΗΣ (Non-precision approach runways)

Οι ακόλουθες επιφάνειες περιορισμού εμποδίων θα εγκαθίστανται για έναν μη ακριβή ενόργανο διάδρομο προσγείωσης:

- Κωνική επιφάνεια
- Εσωτερική οριζόντια επιφάνεια
- Επιφάνεια προσέγγισης
- Μεταβατικές επιφάνειες

Κλίσεις: Τα ύψη και οι κλίσεις των επιφανειών δεν πρέπει να είναι μεγαλύτερες, και οι υπόλοιπες διαστάσεις μικρότερες, από τις τιμές που ορίζονται στον πίνακα 1 στο τέλος του κεφαλαίου, εκτός από την περίπτωση του οριζόντιου τμήματος της επιφάνειας προσέγγισης (βλ. επόμενη παράγραφο).

Η επιφάνεια προσέγγισης πρέπει να είναι οριζόντια μετά το σημείο στο οποίο η 2,5% κλίση τέμνει οποιοδήποτε εκ των ακολούθων είναι υψηλότερο:

- a. Ένα οριζόντιο επίπεδο 150μ πάνω από το ύψος του κατωφλίου
ή
- b. Το οριζόντιο επίπεδο που περνάει από την κορυφή οποιουδήποτε αντικειμένου που «κυριαρχεί» στα όρια απουσίας εμποδίων.

Νέα αντικείμενα ή επεκτάσεις ήδη υφιστάμενων αντικειμένων δεν πρέπει να επιτρέπονται πάνω από την επιφάνεια προσέγγισης εντός των 3000μ από το εσωτερικό άκρο ή τη μεταβατική επιφάνεια εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το καινούριο αντικείμενο ή η επέκταση θα καλυπτόταν από ένα ήδη υπάρχον αμετακίνητο αντικείμενο.

Σύσταση – Νέα αντικείμενα ή επεκτάσεις ήδη υφισταμένων αντικειμένων δεν πρέπει να επιτρέπονται πάνω από την επιφάνεια προσέγγισης πέρα από τα 3000μ από το εσωτερικό άκρο, την κωνική επιφάνεια ή την εσωτερική οριζόντια επιφάνεια εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το καινούριο αντικείμενο ή η επέκταση θα καλυπτόταν από ένα ήδη υπάρχον αμετακίνητο αντικείμενο ή μετά από αεροναυτική μελέτη διαπιστωθεί ότι το αντικείμενο ή η προέκταση δεν θα επηρεάσει αρνητικά την ασφάλεια και την ομαλότητα των χειρισμών των αεροσκαφών.

Σύσταση – Υπάρχοντα αντικείμενα πάνω από οποιαδήποτε επιφάνεια πρέπει, όσο είναι εφικτό, να απομακρύνονται εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το αντικείμενο καλύπτεται από ένα ήδη υπάρχον αμετακίνητο αντικείμενο ή μετά από αεροναυτική μελέτη διαπιστωθεί ότι το αντικείμενο δεν θα επηρεάσει αρνητικά την ασφάλεια και την ομαλότητα των χειρισμών των αεροσκαφών.

Σημείωση – Εξαιτίας εγκάρσιων ή κατά μήκος κλίσεων σε μια ζώνη ασφαλείας, σε ορισμένες περιπτώσεις, το εσωτερικό άκρο ή τμήμα του εσωτερικού άκρου της επιφάνειας προσέγγισης μπορεί να βρίσκεται κάτω από το αντίστοιχο ύψος της ζώνης. Δεν είναι απαραίτητο η ζώνη να διαμορφώνεται ώστε να συμφωνεί με το εσωτερικό άκρο της επιφάνειας προσέγγισης, ούτε τα αντικείμενα που βρίσκονται πάνω από την επιφάνεια προσέγγισης μετά το τέλος της ζώνης, αλλά κάτω από το επίπεδό της, να απομακρύνονται, εκτός αν θεωρηθεί ότι θέτουν σε κίνδυνο το αεροσκάφος.

III. ΑΚΡΙΒΕΙΣ ΕΝΟΡΓΑΝΟΙ ΔΙΑΔΡΟΜΟΙ ΠΡΟΣΕΓΓΙΣΗΣ (Precision approach runways)

Οι ακόλουθες επιφάνειες περιορισμού εμποδίων θα εγκαθίστανται για έναν ακριβή ενόργανο διάδρομο προσγείωσης:

- Κωνική επιφάνεια
- Εσωτερική οριζόντια επιφάνεια
- Επιφάνεια προσέγγισης
- Μεταβατικές επιφάνειες

Σύσταση: Οι ακόλουθες επιφάνειες περιορισμού εμποδίων πρέπει να εγκαθίστανται για έναν ακριβή ενόργανο διάδρομο προσγείωσης κατηγορίας I:

- Επιφάνεια ματαιούμενης προσγείωσης
- Εσωτερική επιφάνεια προσέγγισης
- Εσωτερικές μεταβατικές επιφάνειες

Σύσταση: Οι ακόλουθες επιφάνειες περιορισμού εμποδίων πρέπει να εγκαθίστανται για έναν ακριβή ενόργανο διάδρομο προσγείωσης κατηγορίας II ή III:

- Κωνική επιφάνεια
- Επιφάνεια προσέγγισης και εσωτερική επιφάνεια προσέγγισης
- Μεταβατικές επιφάνειες
- Επιφάνεια ματαιούμενης προσγείωσης
- Εσωτερική οριζόντια επιφάνεια
- Εσωτερικές μεταβατικές επιφάνειες

Κλίσεις: Τα ύψη και οι κλίσεις των επιφανειών δεν πρέπει να είναι μεγαλύτερες, και οι υπόλοιπες διαστάσεις μικρότερες, από τις τιμές που ορίζονται στον πίνακα 1 στο τέλος του κεφαλαίου, εκτός από την περίπτωση του οριζόντιου τμήματος της επιφάνειας προσέγγισης (βλ. επόμενη παράγραφο).

Η επιφάνεια προσέγγισης πρέπει να είναι οριζόντια μετά το σημείο στο οποίο η 2,5% κλίση τέμνει οποιοδήποτε εκ των ακολούθων είναι υψηλότερο:

- a. Ένα οριζόντιο επίπεδο 150μ πάνω από το ύψος του κατωφλίου
ή
- b. Το οριζόντιο επίπεδο που περνάει από την κορυφή οποιουδήποτε αντικειμένου που «κυριαρχεί» στα όρια απουσίας εμποδίων.

Σταθερά αντικείμενα δεν πρέπει να επιτρέπονται πάνω από την εσωτερική επιφάνεια προσέγγισης, την εσωτερική μεταβατική επιφάνεια ή την επιφάνεια ματαιούμενης προσγείωσης, εκτός από εύθραυστα αντικείμενα, τα οποία εξαιτίας της λειτουργίας τους είναι ανάγκη να βρίσκονται επί της ζώνης ασφαλείας. Μετακινήσιμα αντικείμενα δεν πρέπει να επιτρέπονται πάνω από τις επιφάνειες αυτές κατά τη διάρκεια της χρήσης του διαδρόμου για προσγείωση.

Νέα αντικείμενα ή επεκτάσεις ήδη υφιστάμενων αντικειμένων δεν πρέπει να επιτρέπονται πάνω από την επιφάνεια προσέγγισης ή τη μεταβατική επιφάνεια εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το καινούριο αντικείμενο ή η επέκταση θα καλυπτόταν από ένα ήδη υπάρχον αμετακίνητο αντικείμενο.

Σύσταση – Νέα αντικείμενα ή επεκτάσεις ήδη υφιστάμενων αντικειμένων δεν πρέπει να επιτρέπονται πάνω από την κωνική επιφάνεια ή την εσωτερική οριζόντια επιφάνεια εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το καινούριο αντικείμενο ή η επέκταση θα καλυπτόταν από ένα ήδη υπάρχον αμετακίνητο αντικείμενο ή μετά από αεροναυτική μελέτη διαπιστωθεί ότι το αντικείμενο ή η προέκταση δεν θα επηρεάσει αρνητικά την ασφάλεια και την ομαλότητα των χειρισμών των αεροσκαφών.

Σύσταση – Υπάρχοντα αντικείμενα πάνω από μια επιφάνεια προσέγγισης, μια μεταβατική επιφάνεια, την κωνική επιφάνεια και την εσωτερική οριζόντια επιφάνεια πρέπει, όσο είναι εφικτό, να απομακρύνονται εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το αντικείμενο καλύπτεται από ένα ήδη υπάρχον αμετακίνητο αντικείμενο ή μετά από αεροναυτική μελέτη διαπιστωθεί ότι το αντικείμενο δεν θα επηρεάσει αρνητικά την ασφάλεια και την ομαλότητα των χειρισμών των αεροσκαφών.

Σημείωση – Εξαιτίας εγκάρσιων ή κατά μήκος κλίσεων σε μια ζώνη ασφαλείας, σε ορισμένες περιπτώσεις, το εσωτερικό άκρο ή τμήμα του εσωτερικού άκρου της επιφάνειας προσέγγισης μπορεί να βρίσκεται κάτω από το αντίστοιχο ύψος της ζώνης. Δεν είναι απαραίτητο η ζώνη να διαμορφώνεται ώστε να συμφωνεί με το εσωτερικό άκρο της επιφάνειας προσέγγισης, ούτε τα αντικείμενα που βρίσκονται πάνω από την επιφάνεια προσέγγισης μετά το τέλος της ζώνης, αλλά κάτω από το επίπεδό της, να απομακρύνονται, εκτός αν θεωρηθεί ότι θέτουν σε κίνδυνο το αεροσκάφος.

IV. ΔΙΑΔΡΟΜΟΙ ΑΠΟΓΕΙΩΣΗΣ (Runways meant for take-off)

Οι ακόλουθες επιφάνειες περιορισμού εμποδίων θα εγκαθίστανται για έναν ακριβή διάδρομο που προορίζεται για απογείωση:

- Επιφάνεια απογείωσης

Οι διαστάσεις της επιφάνειας δεν πρέπει να είναι μικρότερες από τις τιμές που ορίζονται στον πίνακα 2 στο τέλος του κεφαλαίου, εκτός αν υιοθετηθεί μικρότερο μήκος για την επιφάνεια απογείωσης, συνεπές με τα μέτρα που έχουν ληφθεί για τις απογείώσεις.

Σύσταση – Τα χαρακτηριστικά των χειρισμών των αεροσκαφών για τα οποία προβλέπεται η χρήση του διαδρόμου πρέπει να εξεταστούν αν είναι επιθυμητό να μειωθεί η κλίση που καθορίζεται στον πίνακα 2, όταν πρόκειται να ληφθεί μέριμνα για κρίσιμες συνθήκες χειρισμών. Αν η καθορισμένη κλίση μειωθεί,

αντίστοιχη διευθέτηση πρέπει να γίνει για το μήκος της επιφάνειας απογειώσεως, έτσι ώστε το ύψος να είναι μικρότερο των 300μ.

Σημείωση – Όταν οι τοπικές συνθήκες διαφέρουν κατά πολύ από τις πρότυπες ατμοσφαιρικές στο επίπεδο της θάλασσας, συνίσταται οι κλίσεις στον πίνακα 2 να μειώνονται. Η έκταση αυτής της μείωσης εξαρτάται από την απόκλιση μεταξύ τοπικών συνθηκών και πρότυπων ατμοσφαιρικών στο επίπεδο της θάλασσας και από τα χαρακτηριστικά επιδόσεων και τις λειτουργικές απαιτήσεις των αεροσκαφών για τα οποία προορίζεται ο διάδρομος.

Νέα αντικείμενα ή επεκτάσεις ήδη υφιστάμενων αντικειμένων δεν πρέπει να επιτρέπονται πάνω από την επιφάνεια απογειώσεως εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το καινούριο αντικείμενο ή η επέκταση θα καλυπτόταν από ένα ήδη υπάρχον αμετακίνητο αντικείμενο.

Σύσταση – Αν κανένα αντικείμενο δεν τέμνει την κλίση 2% (1:50) της επιφάνειας απογειώσεως, τα καινούρια αντικείμενα θα πρέπει να περιορίζονται ώστε να διατηρούν την υπάρχουσα ελεύθερη από εμπόδια επιφάνεια ή μια επιφάνεια με κλίση 1,6% (1:62,5).

Σύσταση – Υπάρχοντα αντικείμενα που προεξέχουν πάνω από την επιφάνεια απογειώσεως πρέπει, όσο είναι εφικτό, να απομακρύνονται εκτός εάν, κατά τη γνώμη της αρμόδιας υπηρεσίας, το αντικείμενο καλύπτεται από ένα ήδη υπάρχον αμετακίνητο αντικείμενο ή μετά από αεροναυτική μελέτη διαπιστωθεί ότι το αντικείμενο δεν θα επηρεάσει αρνητικά την ασφάλεια και την ομαλότητα των χειρισμών των αεροσκαφών.

Σημείωση – Εξαιτίας εγκάρσιων ή κατά μήκος κλίσεων της ζώνης ασφαλείας ή του ελεύθερου χώρου (clearway), σε ορισμένες περιπτώσεις, τμήμα του εσωτερικού άκρου της επιφάνειας απογειώσεως μπορεί να βρίσκεται κάτω από το αντίστοιχο ύψος της ζώνης ή του ελεύθερου χώρου. Δεν είναι απαραίτητο η ζώνη να διαμορφώνεται ώστε να συμφωνεί με το εσωτερικό άκρο της επιφάνειας απογειώσεως, ούτε το έδαφος ή τα αντικείμενα που βρίσκονται πάνω από την επιφάνεια απογειώσεως μετά το τέλος της ζώνης ή του ελεύθερου χώρου, αλλά κάτω από το επίπεδό τους, να απομακρύνονται, εκτός αν θεωρηθεί ότι θέτουν σε κίνδυνο τα αεροσκάφη.

Παρόμοιες θεωρήσεις εφαρμόζονται στην τομή ενός ελεύθερου χώρου (clearway) και της ζώνης ασφαλείας, όπου υπάρχουν διαφορές στις εγκάρσιες κλίσεις.

3.3 – ΑΝΤΙΚΕΙΜΕΝΑ ΕΚΤΟΣ ΤΩΝ ΕΠΙΦΑΝΕΙΩΝ

Σύσταση – Για προτεινόμενες κατασκευές, εκτός των ορίων των επιφανειών περιορισμού εμποδίων, που ξεπερνούν το ύψος το οποίο έχει οριστεί από την αρμόδια υπηρεσία, πρέπει να γίνεται αεροναυτική μελέτη των συνεπειών τέτοιων κατασκευών στους χειρισμούς των αεροσκαφών.

Σύσταση – Τα αντικείμενα που ανέρχονται σε ύψος 150μ ή περισσότερο, πάνω από το υψόμετρο του εδάφους πρέπει να θεωρούνται ως εμποδια, εκτός εάν μια ειδική αεροναυτική μελέτη δείξει ότι δεν αποτελούν κίνδυνο για τα αεροσκάφη.

Παρατήρηση – Αυτή η μελέτη αφορά τη φύση των χειρισμών που λαμβάνουν χώρα και μπορεί να διακρίνει μεταξύ ημερησίων και νυχτερινών χειρισμών.

3.4 - ΑΛΛΑ ΑΝΤΙΚΕΙΜΕΝΑ

Σύσταση – Αντικείμενα τα οποία σχεδιάζονται εκτός της επιφάνειας προσέγγισης, που όμως παρόλα αυτά θα επηρέαζαν αρνητικά τις ενόργανες ή μη ενόργανες λειτουργίες, πρέπει, όσο είναι εφικτό, να απομακρύνονται.

Σύσταση – Οτιδήποτε το οποίο κατά τη γνώμη της αρμόδιας υπηρεσίας μετά από αεροναυτική μελέτη, θέτει σε κίνδυνο τα αεροσκάφη στην περιοχή κίνησης ή στον αέρα εντός των ορίων της εσωτερικής οριζόντιας και της κωνικής επιφάνειας πρέπει να θεωρείται ως εμπόδιο και πρέπει, όσο είναι εφικτό, να απομακρύνεται.

Παρατήρηση – Σε συγκεκριμένες περιπτώσεις, αντικείμενα τα οποία δεν προβάλλουν πάνω από τις επιφάνειες που απαριθμούνται στον πίνακα 1, μπορεί να θέτουν σε κίνδυνο τα αεροσκάφη, όπως για παράδειγμα, όπου υπάρχουν ένα ή περισσότερα απομονωμένα αντικείμενα στη γειτονιά του αεροδρομίου.

ΔΙΑΔΡΟΜΟΙ ΠΡΟΣΓΕΙΩΣΗΣ											
ΚΑΤΑΤΑΞΗ ΔΙΑΔΡΟΜΟΥ											
ΕΙΔΟΣ ΠΡΟΣΕΓΓΙΣΗΣ		ΜΗ ΕΝΟΡΓΑΝΗ				ΜΗ ΑΚΡΙΒΗΣ			ΑΚΡΙΒΗΣ		
ΚΩΔΙΚΟΣ ΑΡΙΘΜΟΣ		1	2	3	4	1,2	3	4	I	II	III
ΕΠΙΦΑΝΕΙΑ		ΔΙΑΣΤΑΣΗ ^α									
ΚΩΝΙΚΗ	Κλίση	5%	5%	5%	5%	5%	5%	5%	5%	5%	5%
	Υψος (m)	35	55	75	100	60	75	100	60	100	100
ΕΣΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ	Υψος (m)	45	45	45	45	45	45	45	45	45	45
	Απόσταση από το(τα) σημείο(α) αναφοράς (m)	2000	2500	4000	4000	3500	4000	4000	3500	4000	4000
ΕΣΩΤΕΡΙΚΗ ΠΡΟΣΕΓΓΙΣΗΣ	Πλάτος (m)	-	-	-	-	-	-	-	90	120	120
	Απόσταση του ίχνους αρχής από το κατώφλι (m)	-	-	-	-	-	-	-	60	60	60
	Μήκος (m)	-	-	-	-	-	-	-	900	900	900
	Κλίση	-	-	-	-	-	-	-	2%	2%	2%
ΠΡΟΣΕΓΓΙΣΗΣ	Πλάτος επί του εδάφους (στο ίχνος αρχής) (m)	60	80	150	150	150	300	300	150	300	300
	Απόσταση του ίχνους αρχής από το κατώφλι (m)	30	60	60	60	60	60	60	60	60	60
	Διεύρυνση (από κάθε πλευρά)	10%	10%	10%	10%	15%	15%	15%	15%	15%	15%
	Πρώτο τμήμα										
	Μήκος (m)	1600	2500	3000	3000	2500	3000	3000	3000	3000	3000
	Κλίση	5%	4%	3,33%	2,50%	3,33%	2%	2%	2,50%	2%	2%
	Δεύτερο τμήμα										
	Μήκος (m)	-	-	-	-	-	3600 ^β	3600 ^β	12000	3600 ^β	3600 ^β
	Κλίση	-	-	-	-	-	2,50%	2,50%	3%	2,50%	2,50%
Οριζόντιο τμήμα	Μήκος (m)	-	-	-	-	-	8400 ^β	8400 ^β	-	8400 ^β	8400 ^β
	Ολικό μήκος (m)						15000	15000	15000	15000	15000
ΜΕΤΑΒΑΤΙΚΗ	Κλίση	20%	20%	14,3%	14,3%	20%	14,3%	14,3%	14,3%	14,3%	14,3%
ΕΣΩΤΕΡΙΚΗ ΜΕΤΑΒΑΤΙΚΗ	Κλίση	-	-	-	-	-	-	-	40%	33,3%	33,3%
ΜΑΤΑΙΟΥΜΕΝΗΣ ΠΡΟΣΓΕΙΩΣΗΣ	Πλάτος επί του εδάφους (στο ίχνος αρχής) (m)	-	-	-	-	-	-	-	90	120	120
	Απόσταση από το κατώφλι (m)	-	-	-	-	-	-	-	γ	1800 ^δ	1800 ^δ
	Διεύρυνση (από κάθε πλευρά)	-	-	-	-	-	-	-	10%	10%	10%
	Κλίση	-	-	-	-	-	-	-	4%	3,33%	3,33%

(α) Όλες οι διαστάσεις μετρώνται σε οριζόντια προβολή εκτός αν καθορίζεται διαφορετικά

(β) Μεταβλητό μήκος

(γ) Απόσταση ως το πέρας της ζώνης του διαδρόμου

(δ) Η απόσταση δεν μπορεί να υπερβαίνει το μισό του μήκους του διαδρόμου

ΠΙΝΑΚΑΣ 3.2

ΔΙΑΔΡΟΜΟΙ ΑΠΟΓΕΙΩΣΗΣ				
ΚΩΔΙΚΟΣ ΑΡΙΘΜΟΣ		1	2	3 'Η 4
ΔΙΑΣΤΑΣΗ ^α				
ΕΠΙΦΑΝΕΙΑ ΑΠΟΓΕΙΩΣΗΣ	Μήκος εσωτερικής πλευράς (m)	60	80	180
	Απόσταση ίχνους αρχής από το άκρο του διαδρόμου ^β (m)	30	60	60
	Διεύρυνση (εφαπτόμενη γωνίας οριζόντιας προβολής οριογραμμών κατωτέρας επιφάνειας με τον άξονα του διαδρόμου)	10%	10%	12,5%
	Τελικό εύρος (m)	380	580	1200
				1800 ^γ
	Μήκος (m)	1600	2500	15000
	Κλίση (%)	5	4	2
(α) Όλες οι διαστάσεις μετρώνται σε οριζόντια προβολή εκτός αν καθορίζεται διαφορετικά				
(β) Η κεκλιμένη επιφάνεια αρχίζει από το πέρας της ζώνης ασφαλείας αν το μήκος της ξεπερνά αυτό που ορίζουν οι ακόλουθες αποστάσεις				
(γ) 1800m όταν ο αντίστοιχος διάδρομος περιλαμβάνει αλλαγές πορείας > 15° για λειτουργίες εκτελούμενες σε IMC, VMC τη νύχτα				

ΚΕΦΑΛΑΙΟ 4

ΕΠΙΔΡΑΣΗ ΤΟΠΙΚΩΝ – ΓΕΩΜΕΤΡΙΚΩΝ ΣΥΝΘΗΚΩΝ ΣΤΟΝ ΣΧΕΔΙΑΣΜΟ ΑΕΡΟΔΡΟΜΙΩΝ

Είναι γεγονός πως ένας αερολιμένας αναπτύσσεται «γύρω» από τους διαδρόμους του. Το τροχοδρομικό σύστημα και τα δάπεδα στάθμευσης χωροθετούνται και διαστασιολογούνται ανάλογα με τους διαδρόμους που επιλέξαμε να χρησιμοποιήσουμε, οι οποίοι με τη σειρά τους επιλέγονται σχετιζόμενοι με το μέγεθος και τη σύσταση της προβλεπόμενης κίνησης. Επίσης οι επιφάνειες περιορισμού εμποδίων εξαρτώνται από τους διαδρόμους και έτσι επηρεάζεται το επιτρεπτό υψόμετρο και επομένως και η μορφή των κτιρίων – εγκαταστάσεων στο χώρο του αερολιμένα. Τελικά θα μπορούσαμε να πούμε πως ο σχεδιασμός ενός αερολιμένα (σε μεγάλο βαθμό και φυσικά αφού έχουν επιλεγεί οι διάδρομοι) σχετίζεται με τους διαδρόμους.

Στο κεφάλαιο 2.4 ορίσαμε το μήκος πεδίου αναφοράς αεροπλάνου (aeroplane field length κατά Annex 14) ως: *Το ελάχιστο μήκος πεδίου που απαιτείται για απογείωση με τη μέγιστη εγκεκριμένη μάζα απογείωσης, στο επίπεδο θαλάσσης ($\pm 0,00$), με πρότυπες ατμοσφαιρικές συνθήκες, άνευ ανέμου και μηδενική κλίση διαδρόμου, όπως παρουσιάζεται στο σχετικό εγχειρίδιο πτήσης του αεροσκάφους που καθορίζεται από την αρχή πιστοποίησης ή σε αντίστοιχα δεδομένα από τον κατασκευαστή του αεροσκάφους.* Το μήκος αυτό αντιστοιχεί και μπορεί να ονομαστεί βασικό μήκος διαδρόμου για το συγκεκριμένο α/φ. Είναι δε το εξισορροπημένο (βασικό) μήκος L_b του διαδρόμου ή το απαιτητό μήκος απογείωσης υπό τις προαναφερθείσες συνθήκες.

Το βασικό αυτό μήκος του διαδρόμου επιβάλλεται να υποστεί διορθώσεις που οφείλονται σε μεταβολές παραμέτρων που θεωρήθηκαν σταθερές κατά τον ορισμό του. Αυτό είναι επιτακτικό εξαιτίας της συνεπαγόμενης μεταβολής της πίεσης του αέρα. Η ισορροπία κατά την πτήση είναι αποτέλεσμα της αεροδυναμικής διαφοράς πιέσεων στην κάτω και άνω επιφάνεια των φτερών του α/φ, κάτι που εξαρτάται από την μορφή των φτερών και τη σχετική πυκνότητα του αέρα (το όλο φαινόμενο δημιουργείται από διαφορετικές ταχύτητες ροής του αέρα στην άνω και κάτω επιφάνεια των φτερών)· έτσι κάθε αίτιο που προκαλεί μεταβολή της πυκνότητας οδηγεί σε μεταβολή του απαιτούμενου μήκους διαδρόμου και κατά συνέπεια πρέπει να λαμβάνεται υπόψη στο σχεδιασμό των αεροδρομίων. Τα κύρια αίτια τέτοιων μεταβολών είναι η θερμοκρασία, το υψόμετρο και η κατά μήκος κλίση του διαδρόμου.

Επίσης το βασικό μήκος διαδρόμου τροποποιείται ανάλογα με το μήκος ταξιδιού που πρόκειται να κάνει ένα αεροσκάφος. Η σχέση ισχύος – βάρους επηρεάζει την απαιτούμενη απόσταση απογείωσης και αφού μεγάλο ποσοστό του βάρους αποτελούν τα καύσιμα, ο όγκος των οποίων καθορίζεται από την απόσταση που θα διανύσει το αεροσκάφος, το μήκος του ταξιδιού που προβλέπεται να πραγματοποιηθεί επηρεάζει ουσιαστικά το απαιτούμενο μήκος διαδρόμου.

Ο σχεδιασμός του διαδρόμου δεν επηρεάζεται μόνο ως προς το μήκος του αλλά και ως προς τη διεύθυνσή του. Καθώς στην ανάπτυξη της αεροδυναμικής ανώσεως η σχετική ταχύτητα α/φ – αέρα παίζει πρωτεύοντα ρόλο, τα α/φ επιχειρούν πάντοτε να εκτελούν τις προσγειοαπογεώσεις τους αντίθετα προς τον άνεμο. Αυτός και οι παραπάνω παράγοντες αναλύονται στο παρόν κεφάλαιο.

4.1 – ΕΠΙΡΡΟΗ ΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΣΥΝΘΗΚΩΝ

Γενικά

Η ορατότητα, η δημιουργία ομιχλών, η υψηλή βροχόπτωση επιδρούν ανασταλτικά στη λειτουργία του αεροδρομίου.

Ειδικότερα ο άνεμος και η θερμοκρασία είναι μετεωρολογικές συνθήκες που επηρεάζουν τη λήψη αποφάσεων για τη θέση, το μέγεθος και τη διάταξη του αεροδρομίου.

Η θερμοκρασία επιδρά στο μήκος των διαδρόμων, αφού οι υψηλές θερμοκρασίες απαιτούν μεγαλύτερα μήκη διαδρόμων.

Ευνοϊκοί άνεμοι μειώνουν το μήκος του διαδρόμου που χρειάζεται για την κανονική λειτουργία του αεροδρομίου, ενώ αντίθετα μη ευνοϊκοί το αυξάνουν.

Επιρροή της θερμοκρασίας

Η θερμοκρασία επιδρά στην πυκνότητα του αέρα αλλά και στην απόδοση των κινητήρων. Οι μηχανές των αεριωθούμενων αεροπλάνων είναι ιδιαίτερα ευαίσθητες στη θερμοκρασία. Έτσι για παράδειγμα η ωστική δύναμη στην έξοδο ενός στροβιλοκινητήρα (που χρησιμοποιείται στο Boeing 707-120) ελαττώνεται αισθητά από μια μικρή αύξηση της θερμοκρασίας. Η ελάττωση αυτή της ωθήσεως έχει σαν αποτέλεσμα αντίστοιχη αύξηση στο απαιτούμενο μήκος διαδρόμου.

Είναι λοιπόν προφανές ότι δύο αεροδρόμια που εξυπηρετούν τον ίδιο τύπο αεροσκάφους, δεν είναι ανάγκη να έχουν και το ίδιο μήκος διαδρόμου αν η θερμοκρασία τους είναι διαφορετική. Έτσι, αεροδρόμια σε τοποθεσίες με ψυχρό κλίμα μπορεί να απαιτούν μήκος διαδρόμων πολλές εκατοντάδες μέτρα μικρότερο απ' ό τι σε τοποθεσίες με θερμό κλίμα.

Για παράδειγμα ένα Boeing 707-320 με βάρος απογειώσεως 320000lb, στο επίπεδο της θάλασσας, χωρίς επικρατούντες ανέμους και χωρίς κλίση, απαιτεί μήκος διαδρόμου 3500μ, όταν η θερμοκρασία είναι 15°C, ενώ απαιτεί 4720μ με θερμοκρασία 37°C

Διόρθωση μήκους διαδρόμου λόγω θερμοκρασίας

Πιο συγκεκριμένα, ορίζεται αρχικά η συμβατική θερμοκρασία T του αεροδρομίου από τον τύπο:

$$T = T_1 + \frac{1}{3}(T_2 - T_1), \text{ όπου}$$

T_1 : μέση θερμοκρασία του θερμότερου μήνα του έτους (θερμότερος θεωρείται ο μήνας που περιλαμβάνει τη μέγιστη μέση ημερήσια θερμοκρασία)

T_2 : μέγιστη μέση ημερήσια θερμοκρασία του ίδιου μήνα

Για κάθε 1°C που η T υπερβαίνει τη θερμοκρασία της σταθερής ατμόσφαιρας για το ίδιο υψόμετρο (υψόμετρο του αεροδρομίου), το μήκος του διαδρόμου πρέπει να αυξάνεται κατά 1%.

4.2 – ΕΠΙΡΡΟΗ ΤΩΝ ΑΝΕΜΩΝ - ΑΝΕΜΟΛΟΓΙΟ

Η ισορροπία του α/φ στον αέρα, όπως έχουμε αναφέρει παραπάνω, είναι αεροδυναμική και εξαρτάται και από τη σχετική ταχύτητα α/φ – αέρα. Στα μεγάλα ύψη η διατάραξη της ισορροπίας δεν είναι ιδιαίτερα επικίνδυνη σε αντίθεση με τα χαμηλά ύψη, λίγο πριν την απογείωση και αμέσως μετά την προσγείωση. Βέβαια η μεταβολή της διεύθυνσης του ανέμου δεν είναι δυνατόν να παρακολουθείται, παρά μόνο σε τρόπο που η απογείωση (ή προσγείωση) να ελαχιστοποιεί την εγκάρσια, προς την διεύθυνση τροχοδρομήσεως του α/φ, συνιστώσα του ανέμου.

Σύμφωνα με τα παραπάνω ο διάδρομος, ή οι διάδρομοι, επιδιώκεται να έχουν διεύθυνση συγγραμική προς τη διεύθυνση των κρατούντων ανέμων (μειωμένη δηλαδή κάθετη συνιστώσα), ενώ επιβάλλεται και η απαγόρευση της προσγειοαπογείωσης α/φ, όταν η εγκάρσια συνιστώσα του ανέμου υπερβαίνει κάποια τιμή ασφαλείας. Όσο μεγαλύτερο είναι το βάρος του α/φ και κυρίως ο λόγος ισχύος/βάρους απογείωσης, τόσο μεγαλύτερη είναι η μέγιστη εγκάρσια συνιστώσα του ανέμου που γίνεται δεκτή. Συνεπώς είναι δυνατόν, κάτω από συγκεκριμένες συνθήκες ανέμου, να απαγορευθεί η χρήση ενός διαδρόμου για α/φ ενός τύπου και να επιτρέπεται για άλλα.

Για τα πολιτικά αεροδρόμια που χρησιμοποιούνται από τα σύγχρονα επιβατικά και εμπορευματικά α/φ, αναγνωρίζεται μια τιμή κάθετης, προς τη διεύθυνση του διαδρόμου, συνιστώσας ανέμου, που να είναι παραδεκτή από όλους τους σύγχρονους τύπους α/φ. Η τιμή αυτή είναι 15 mi/h (24 km/h).

Τα πολιτικά αεροδρόμια μάλιστα που εξυπηρετούν τακτικές αερογραμμές και μάλιστα διεθνείς, πρέπει να μην αχρηστεύονται λόγω ανέμων, στο 95% τουλάχιστον του χρόνου.

Για την επιλογή του κατάλληλου προσανατολισμού των διαδρόμων απαιτείται η συλλογή και στατιστική επεξεργασία των ανεμολογικών στοιχείων της περιοχής του αεροδρομίου. Τα στοιχεία αυτά πρέπει να καλύπτουν ένα, όσο το

δυνατόν μεγαλύτερο, συνεχές χρονικό διάστημα, τουλάχιστον 5 ετών. Με τη βοήθεια των στοιχείων αυτών και των παραπάνω δεσμεύσεων για την ταχύτητα των πλαγίων ανέμων και το χρόνο χρήσης του αεροδρομίου, μπορεί να κατασκευαστεί ανεμολόγιο όπως φαίνεται στο σχήμα 4.1.

Η κατασκευή ανεμολογίου έχει ως εξής: Επί της περιφέρειας ενός κύκλου αναγράφονται 16 συνήθως διευθύνσεις ανέμου (βλ. σχ. 4.1) ώστε κάθε διεύθυνση ν' αντιστοιχεί σ' ένα κυκλικό τομέα $22,5^\circ$. Τέσσερις κατηγορίες ταχυτήτων ανέμων ($0 \div 8$), ($8 \div 24$), ($24 \div 48$), ($48 \div 73$) km/h αντιστοιχούν σε κυκλικούς δακτυλίους όπου ο πρώτος αντιστοιχεί στον κύκλο ακτίνας 8km/h.

Στην τομή κάθε κυκλικού δακτυλίου και κυκλικού τομέως αναγράφεται το αντίστοιχο ποσοστό των ανέμων. Τούτο είναι το ποσοστό του χρόνου κατά την διάρκεια του οποίου έχουν πνεύσει άνεμοι με ταχύτητα μέσα στα όρια του αντίστοιχου δακτυλίου και με τη διεύθυνση του αντίστοιχου τομέα.

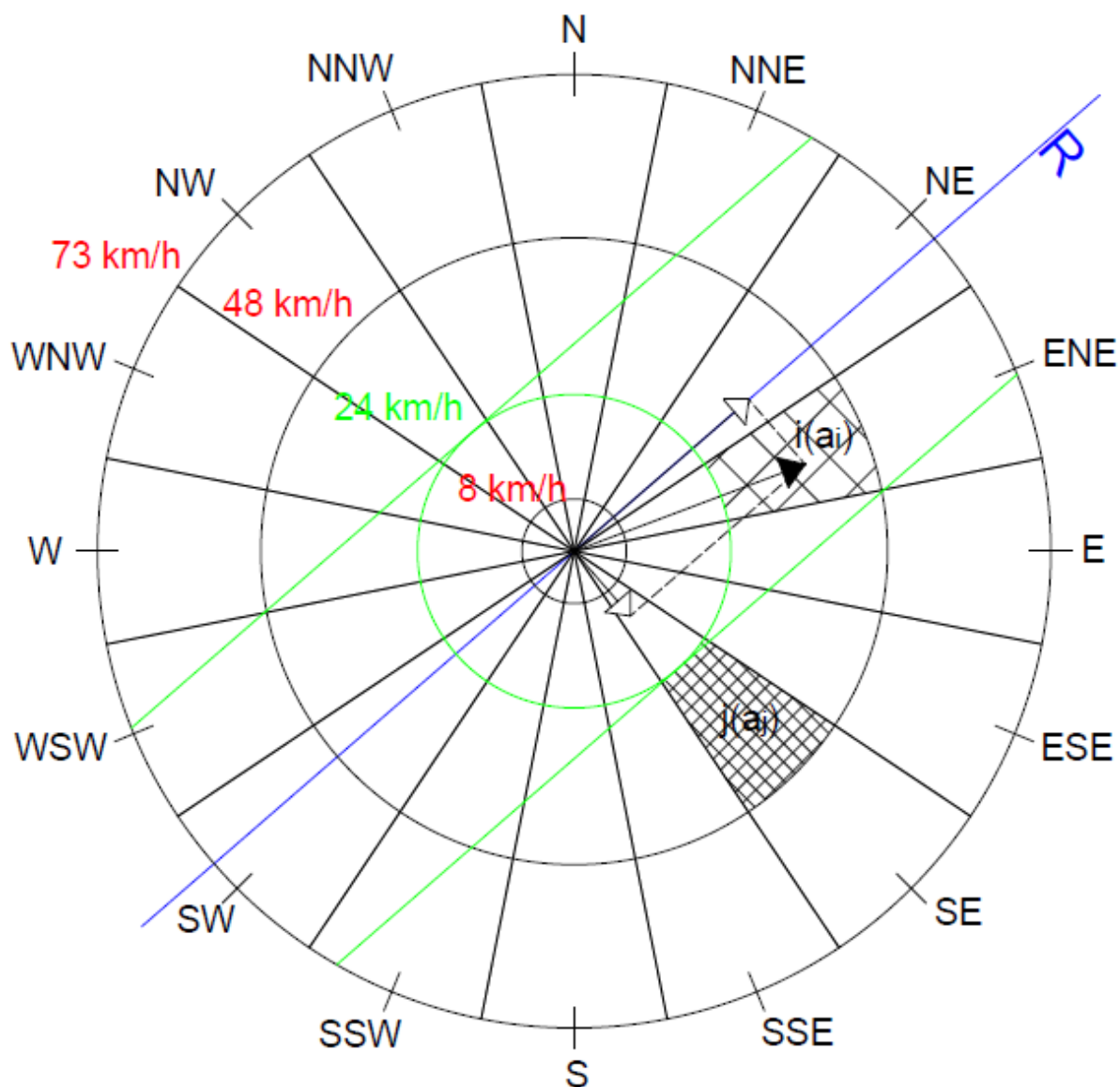
Αν στο ανεμολόγιο χαραχθεί ο προσανατολισμένος άξονας του διαδρόμου [R] και χαραχθούν δύο παράλληλες εφαπτόμενες στον κύκλο των 24 km/h, τότε η μεταξύ των δύο παραλλήλων περιοχή του κύκλου, δείχνει τα ποσοστά του χρόνου, κατά τον οποίο οι πνέοντες άνεμοι έχουν συνιστώσα κάθετη προς τη διεύθυνση του διαδρόμου, μικρότερη των 24 km/h, σε αντίθεση με τις υπόλοιπες περιοχές του κύκλου. Το άθροισμα των ποσοστών των περιοχών του κύκλου που βρίσκονται ανάμεσα στις δύο παράλληλες πρέπει να είναι για διεθνές αεροδρόμιο ποσοστό μεγαλύτερο του 95%. Με τον τρόπο αυτό βρίσκεται η βέλτιστη διεύθυνση του διαδρόμου.

Πραγματικά η επιβατική ακτίνα προς το τυχόν σημείο i του σχήματος 4.1 παριστάνει ποσοστό χρόνου που πνέει άνεμος ορισμένης ταχύτητας και διεύθυνσεως. Αν το i εκλεγεί στο κεντροειδές της επιφάνειας, που είναι τομή του αντίστοιχου κυκλικού τομέα και δακτυλίου (σκιαγραφημένη επιφάνεια), τότε το ποσοστό a_i (πχ $a_i = 3\%$) παριστάνει το ποσοστό του χρόνου κατά το οποίο οι άνεμοι έπνεαν από ENE και είχαν ταχύτητα μεταξύ των ορίων του δακτυλίου.

Όπως εύκολα φαίνεται η κάθετη προς την R συνιστώσα (πλάγιοι άνεμοι), δεν υπερβαίνει την οριακή τιμή των 24 km/h αν η επιφάνεια i βρίσκεται μέσα στην περιοχή που εγκλείουν οι δύο παράλληλες προς την R (σχ. 4.1). Αντίθετα αν j είναι σημείο, ή μικρή περιοχή εκτός των παραλλήλων, τότε το a_j παριστάνει ποσοστό χρόνου που η κάθετη προς R συνιστώσα των ανέμων υπερβαίνει την τιμή 24 km/h.

Βέλτιστη είναι η διεύθυνση R για την οποία το άθροισμα των a_i που αντιστοιχούν σε τμήματα εκτός των παραλλήλων είναι ελάχιστο : $\sum_j a_j \rightarrow \min$

Για διεθνές αεροδρόμιο πρέπει : $\sum_j a_j < 5\%$



ΣΧΗΜΑ 4.1

4.3 – ΕΠΙΡΡΟΗ ΤΟΠΟΓΡΑΦΙΚΩΝ ΣΥΝΘΗΚΩΝ

Γενικά

Είναι γεγονός ότι η τοπογραφία της περιοχής, λόγω του υψομέτρου, της κλίσης του εδάφους, της τοποθεσίας και ποικιλίας των φυσικών χαρακτηριστικών, όπως η φυσική βλάστηση, υδάτινες πηγές και η ύπαρξη ανθρώπινων κατασκευών, είναι καθοριστικός παράγοντας για τον όγκο των χωματουργικών εργασιών, των έργων αποχετεύσεως και αποστραγγίσεως για διαμορφώσεις κλπ.

Έδαφος το οποίο έχει μικρές κλίσεις κοντά στα επιθυμητά όρια και που αποχετεύεται και αποστραγγίζεται σε ικανοποιητικό βαθμό, μπορεί να μειώσει κατά πολύ το κόστος κατασκευής.

Η γεωλογική σύσταση των εδαφών είναι σημαντική από την άποψη του κόστους κατασκευής των έργων.

Η τοποθεσία των λατομείων και των υδάτινων πηγών είναι επίσης σχετική, γιατί επηρεάζει το κόστος κατασκευής του έργου, την μετέπειτα συντήρησή του και τα έργα αναπτύξεώς του.

Στα αεροδρόμια που κατασκευάζονται σε τροπικές περιοχές, ή ζώνες με ενδημικές ασθένειες, θα πρέπει να προβλέπονται μέτρα που να περιορίζουν τις πιθανότητες μετάδοσης των ασθενειών. Έτσι λχ θα πρέπει να αποξηρανθούν τυχόν έλη που βρίσκονται σε μικρή απόσταση από τον αερολιμένα.

Η τοπογραφία ενδιαφέρει και στη γύρω από το αεροδρόμιο περιοχή και ιδίως σε ζώνες μέχρι και 15 χλμ από τα πέρατα των διαδρόμων και κατ επέκτασή τους. Ο έλεγχος των διαθέσιμων ελευθέρων υψών επιβάλλει πολλές φορές δαπανηρές αφαιρέσεις εμποδίων, το κόστος των οποίων αποτελεί σε αρκετές περιπτώσεις σοβαρό ποσοστό του συνολικού κόστους της κατασκευής.

Επιρροή υψομέτρου αεροδρομίου στο μήκος διαδρόμου

Πέρα από την προαναφερθείσα συσχέτιση του υψομέτρου κατασκευής ενός αεροδρομίου με τις χωματουργικές εργασίες, το (μέσο) υψόμετρο ενός αεροδρομίου έχει και αυξητική επιρροή στο μήκος του διαδρόμου, λόγω της μείωσης της πυκνότητας του αέρα με την αύξηση του υψομέτρου. Η αύξηση αυτή επιβάλλεται επειδή το μήκος τροχοδρομήσεως, τόσο στην απογείωση όσο και στην προσγείωση είναι αυξημένο. Συγκεκριμένα και στις δύο φάσεις το α/φ για να «στηριχθεί» στον αέρα χρειάζεται μεγαλύτερη ταχύτητα, δηλαδή μεγαλύτερο μήκος τροχοδρομήσεως.

Διόρθωση μήκους διαδρόμου λόγω υψομέτρου

Οι κανονισμοί συνιστούν αύξηση του μήκους του διαδρόμου 7% για κάθε 300 μέτρα μεταβολή υψομέτρου.

4.4 – ΕΠΙΡΡΟΗ ΓΕΩΜΕΤΡΙΚΩΝ ΣΥΝΘΗΚΩΝ ΣΤΟ ΜΗΚΟΣ ΔΙΑΔΡΟΜΟΥ

Η κατά μήκος κλίση του διαδρόμου μεταβάλλει τα υψόμετρα, με αποτέλεσμα να επηρεάζει έμμεσα και το μήκος του διαδρόμου. Φυσικά η κατά μήκος κλίση του διαδρόμου δεν πρέπει να υπερβαίνει τις τιμές που αναγράφονται στο κεφάλαιο 5.

Διόρθωση μήκους διαδρόμου λόγω κατά μήκος κλίσης διαδρόμου

Η αύξηση των διαδρόμων A,B,C πρέπει να λαμβάνεται 10% για κάθε 1% μέγιστη κατά μήκος κλίση του διαδρόμου. Ισχύει γραμμική παρεμβολή.

ΚΕΦΑΛΑΙΟ 5

ΓΕΩΜΕΤΡΙΚΗ ΜΕΛΕΤΗ ΑΕΡΟΔΡΟΜΙΩΝ

Η γεωμετρική μελέτη ενός αεροδρομίου αναφέρεται σε όλα τα επιμέρους στοιχεία του: τον διάδρομο, τους τροχόδρομους, τις συνδέσεις τους (εξόδους) από τον διάδρομο καθώς και τα δάπεδα αναμονής. Περιλαμβάνει επίσης τις επιφάνειες περιορισμού εμποδίων που επιβάλλεται να εξασφαλίζονται στην εγγύς και την ευρύτερη περιοχή του αεροδρομίου. Η γεωμετρική μελέτη περιλαμβάνει επίσης την κατάρτιση του γενικού σχεδίου (master plan) του αεροδρομίου, το οποίο περιέχει τη χωροθέτηση όλων των βασικών στοιχείων του αεροδρομίου.

Η γεωμετρική μελέτη προϋποθέτει επίσης ότι έχουν γίνει οι κατάλληλες αναλύσεις και μελέτες για τον καθορισμό

- του πλήθους και προσανατολισμού των απαιτούμενων διαδρόμων
- των στοιχείων εισόδου για τον υπολογισμό των συντελεστών επαύξησης του βασικού μήκους διαδρόμου ώστε να προκύψει το πραγματικό – εφαρμοστέο μήκος.

Το παρόν κεφάλαιο εστιάζει στους διαδρόμους καθώς η τελική μορφολογία τους παίζει καθοριστικό ρόλο στα γεωμετρικά χαρακτηριστικά των επιφανειών περιορισμού εμποδίων, η στερεομετρική ανάλυση των οποίων αποτελεί (γενικά) τον βασικό σκοπό της διπλωματικής εργασίας.

5.1 – ΔΙΑΔΡΟΜΟΣ - ΕΙΣΑΓΩΓΙΚΑ

Διάδρομος (Runway ή RW) είναι μια καθορισμένη ορθογωνική περιοχή ενός χερσαίου αεροδρομίου, που είναι προετοιμασμένη για την προσγείωση και απογείωση αεροσκαφών (ορισμός του Annex 14). Πρέπει ωστόσο να διευκρινιστούν τα παρακάτω:

- Η «περιοχή» έχει τη μορφή λωρίδας
- Η «ορθογωνικότητα» αναφέρεται στην οριζόντια προβολή της
- Η προετοιμασία κατά κανόνα σημαίνει μια πλήρη οδοστρωσία, ικανή να παραλάβει τα φορτία των αεροσκαφών κατά τον τρόπο και την επαναληπτικότητα επιβολής τους, καθώς και γεωμετρία επιφάνειας υπαγορευμένης από οικείους κανονισμούς.

Σε κάθε διάδρομο ορίζεται για την διαδικασία προσγείωσης μια αρχή που επισημαίνεται (σήμανση ημέρας και νύχτας) ευκρινώς για τους χειριστές των προσγειουμένων α/φ. Λέγεται **κατώφλι (threshold)** και κατά ICAO είναι η αρχή (νοείται προφανώς κατά τη φορά προσγείωσης) αυτού του τμήματος του διαδρόμου που μπορεί να χρησιμοποιηθεί για προσγείωση (βλ. σχ. 5.1). Το κατώφλι μπορεί να είναι στο άκρο του διαδρόμου ή εσώτερα. Στη δεύτερη περίπτωση ονομάζεται «μετατοπισμένο κατώφλι» (displaced threshold).

5.2 – ΔΗΛΟΥΜΕΝΕΣ ΑΠΟΣΤΑΣΕΙΣ

Στον διάδρομο αλλά και πέραν αυτού στην έννοια του μήκους καθώς και παραπλεύρως αυτού, καθορίζονται αντιστοίχως χαρακτηριστικές αποστάσεις και πρόσθετες περιοχές που εξυπηρετούν την ασφάλεια των διαδικασιών προσγείωσης και απογείωσης. Οι αποστάσεις δηλώνονται και καθορίζουν τον διάδρομο. Οι κυριότερες εξ αυτών είναι οι ακόλουθες (βλ. σχ. 5.1):

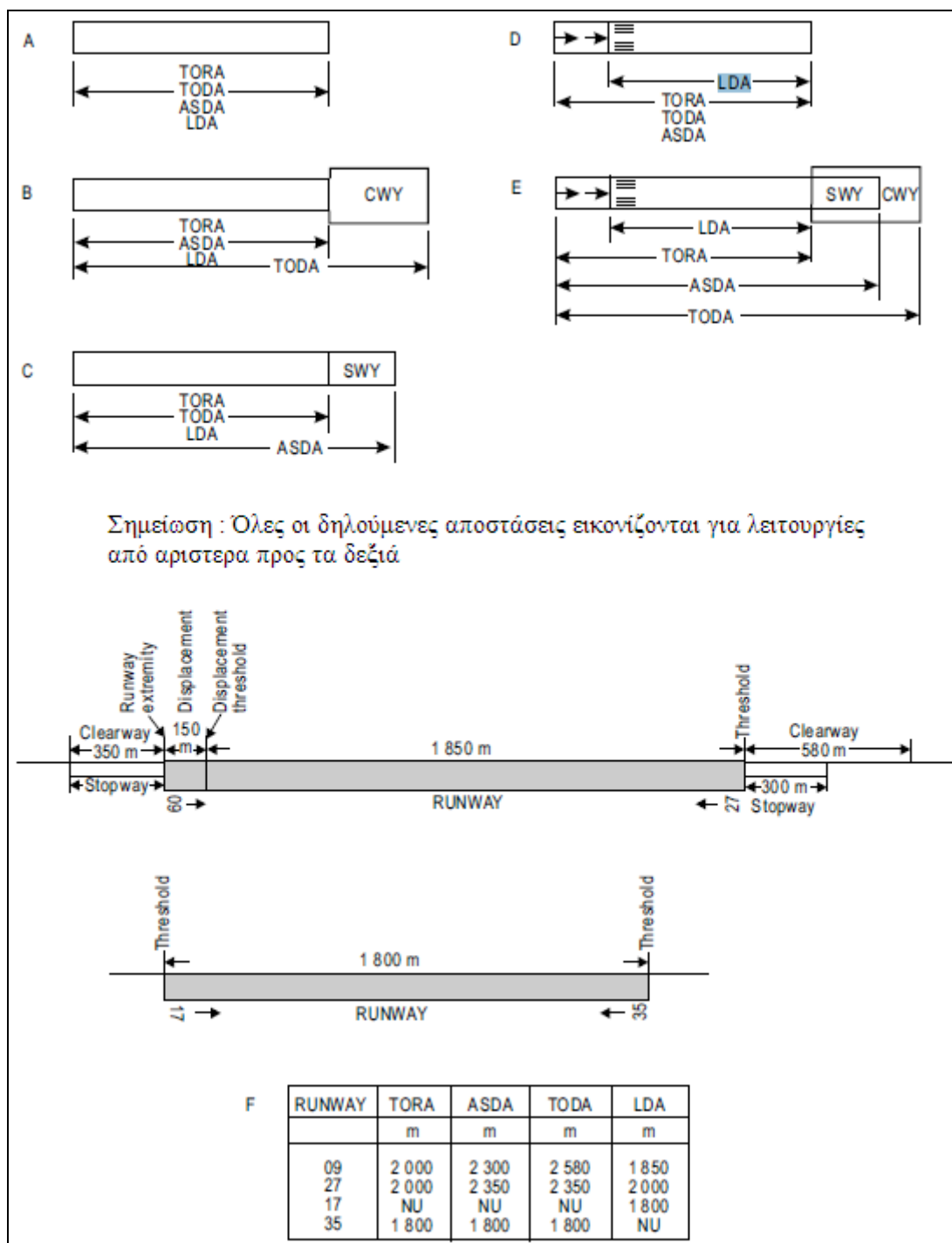
TORA – Διαθέσιμο μήκος τροχοδρόμησης απογείωσης (Take off run available): Είναι το μήκος του διαδρόμου που έχει δηλωθεί ως διαθέσιμο και είναι κατάλληλο για την προς απογείωση τροχοδρόμηση ενός α/φ

TODA – Διαθέσιμο μήκος απογείωσης (Take off distance available): Είναι η απόσταση μέσα στην οποία το α/φ απογειώνεται μέχρι ύψους 35ft τουλάχιστον

ASDA – Διαθέσιμο μήκος επιταχυνόμενης τροχοδρόμησης – στάσης (Accelerated – Stop distance available): Είναι το TORA προσαυξημένο κατά το τμήμα SWY, όταν αυτό προβλέπεται. Το μήκος αυτό είναι τόσο ώστε το απογειούμενο α/φ, να μπορεί να ματαιώσει την απογείωση και να σταματήσει τροχοδρομώντας επί της επιστρωμένης επιφάνειας.

LDA – Διαθέσιμο μήκος προσγείωσης (Landing distance available): Είναι το τμήμα του διαδρόμου που δηλώνεται ως διαθέσιμο και είναι κατάλληλο για την

τροχοδρόμηση προσγειούμενου α/φ. Το μήκος αυτό αρχίζει από το καθοριζόμενο και επισημαινόμενο κατώφλι και πρέπει να είναι τόσο ώστε σε απόσταση όχι μεγαλύτερη από 60% LDA να είναι δυνατή η στάση α/φ που προσγειώνεται και επίσης α/φ που απογειώνεται υπερπιτάμενο του κατωφλίου να βρίσκεται σε ύψος 50 ft.



ΣΧΗΜΑ 5.1

5.3 – ΖΩΝΕΣ ΑΣΦΑΛΕΙΑΣ

Πέραν των άκρων του διαδρόμου πρέπει να προβλέπονται ειδικές ζώνες που παρέχουν ασφάλεια στις περιπτώσεις κανονικής αλλά και ματαιούμενης απογείωσης. Οι ζώνες αυτές μπορεί να παραλείπονται όταν η κατασκευή του διαδρόμου προεκτείνεται σε ικανό μήκος που να διασφαλίζει τις απαιτούμενες αποστάσεις. Η ύπαρξή τους, όταν προβλέπονται, εξυπηρετεί οικονομικούς λόγους, μειώνοντας το μήκος του πλήρους αντοχής οδοστρώματος του διαδρόμου, χωρίς να παραβλάπτεται η ασφάλεια των διαδικασιών προσγείωσης – απογείωσης. Οι ζώνες αυτές φαίνονται στο σχήμα 5.1 και ορίζονται ως ακολούθως:

CWY (Clearway) – Ακραία ζώνη ασφαλείας απογείωσης: Είναι μια καθορισμένη ορθογωνική επιφάνεια στο έδαφος που έχει επιλεγεί και προετοιμαστεί κατάλληλα, υπεράνω της οποίας το α/φ μπορεί να πραγματοποιήσει μέρος της διαδικασίας ανόδου μέχρι ορισμένου ύψους. Το CWY βρίσκεται πέραν του άκρου του διαδρόμου και πρέπει να είναι ελεύθερο εμποδίων. Μόνο φώτα επισήμανσης επιτρέπεται να εξέχουν όταν έχουν ύψος όχι μεγαλύτερο από 26 in και διατάσσονται παρά την επέκταση των πλευρών του διαδρόμου. Η κλίση του επιπέδου που αρχίζει από το άκρο του διαδρόμου / αρχή CWY είναι 1,25% ανηφορική. Δεν είναι αναγκαία η επίστρωση του CWY.

SWY (Stopway) – Ακραία ζώνη ασφαλείας στάσης (στην περίπτωση ματαιωθείσας απογείωσης): Το SWY είναι οπτικά προέκταση του διαδρόμου. Διασφαλίζει το μήκος ASDA, αν το πραγματικό μήκος του διαδρόμου δεν το υπερκαλύπτει.

Σημειώνουμε πως στο σχήμα 5.1 εικονίζονται διάφορες περιπτώσεις χρήσης των προαναφερθέντων αποστάσεων και για τις τελευταίες δύο περιπτώσεις διαδρόμων δίνονται παραδείγματα πραγματικών αποστάσεων καθώς και αντίστοιχος πίνακας πληροφόρησης. Η ένδειξη NU σημαίνει «μη επιτρεπτή χρησιμοποίηση» (not usable)

5.4 – ΦΥΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΔΙΑΔΡΟΜΟΥ

Μήκος διαδρόμου

Το μήκος του διαδρόμου προκύπτει από το βασικό μήκος του α/φ σχεδιασμού (βλ. κεφ. 2.4), έπειτα από τις απαιτούμενες διορθώσεις (βλ. κεφ. 4)

Πλάτος διαδρόμου

Κατά ICAO συστήνεται το πλάτος του διαδρόμου (χωρίς τα ερείσματα) να μην είναι μικρότερο από τις παρακάτω τιμές

Κωδικός Αριθμός	Κωδικό γράμμα				
	A	B	C	D	E
1*	18	18	23	-	-
2*	23	23	30	-	-
3	30	30	30	45	-
4	-	-	45	45	45

ΠΙΝΑΚΑΣ 5.1 – ΕΛΑΧΙΣΤΟ ΠΛΑΤΟΣ ΔΙΑΔΡΟΜΟΥ (m)

*Το πλάτος ενός διαδρόμου που χρησιμοποιείται για προσγειώσεις με ενόργανη ακριβή προσέγγιση δεν πρέπει να είναι μικρότερο από 30m

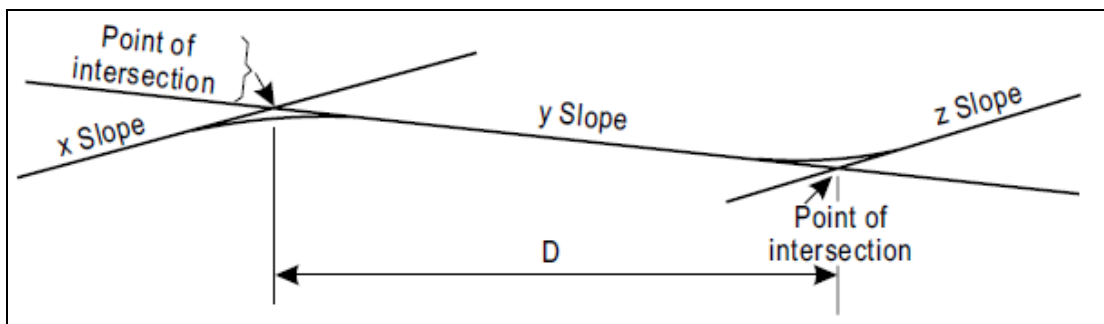
Ερείσματα

Τα ερείσματα επεκτείνουν την επίστρωση πέραν του πλάτους του διαδρόμου και με τον τρόπο αυτόν το μεν διασφαλίζουν από πιθανές αθέλητες τροχοδρομήσεις με άξονα διαφορετικό από τον διαμήκη άξονα του διαδρόμου, το δε διασφαλίζουν από την αναρρόφηση/απόσπαση κόκκων και εμπλοκή τους στις μηχανές. Οι κυριότερες συστάσεις κατά ICAO είναι:

- Συμμετρική (ως προς άξονα) διάταξη των ερεισμάτων
- Πλάτος τόσο ώστε ιδιαίτερα για διαδρόμους με κωδικό D και E το συνολικό πλάτος διαδρόμου και ερεισμάτων να είναι τουλάχιστον 60 m.
- Η επιφάνεια να διαμορφώνεται με εγκάρσια κλίση προς τα έξω 2,5%.
- Η αντοχή τους να είναι κατάλληλη για να φέρει τα φορτία τροχών α/φ που μπορεί να εκτραπούν (που όμως, λόγω της μικρής συχνότητας, μπορεί να είναι μικρότερης αντοχής από το διάδρομο).

Μηκοτομή διαδρόμου

Η μηκοτομή ενός διαδρόμου διαμορφώνεται με ήπιες κλίσεις και μεγάλες ακτίνες καμπυλών κατακόρυφης συναρμογής. Οι κατά ICAO συνιστώμενοι περιορισμοί συνοψίζονται στον πίνακα 5.2 ενώ ένα προφίλ κεντρικού άξονα (μέσης γραμμής) διαδρόμου φαίνεται στο σχήμα 5.2



ΣΧΗΜΑ 5.2

Στοιχείο σχεδιασμού	Κωδικός αριθμός διαδρόμου			
	1	2	3	4
Μέγιστη κλίση στον άξονα (μέση γραμμή) διαδρόμου	2%	2%	1%	1%
Μέγιστη κλίση κατά μήκος οποιασδήποτε γραμμής παράλληλης με τον άξονα Α. Γενικά Β. Ακραία τέταρτα μήκους	2%	2%	1,5%	1,25%
Μέγιστη διαφορά δύο διαδοχικών κατά μήκος κλίσεων	2%	2%	0,8%	0.8%
Ελάχιστη ακτίνα καμπυλότητας κατά μήκος συναρμογής (m)	1500	7500	15000	30000

ΠΙΝΑΚΑΣ 5.2 – ΠΕΡΙΟΡΙΣΜΟΙ ΜΗΚΟΤΟΜΗΣ ΔΙΑΔΡΟΜΟΥ

Εγκάρσιες κλίσεις διαδρόμου

Οι εγκάρσιες κλίσεις πρέπει να περιορίζονται ενώ παράλληλα να εξασφαλίζουν την απορροή των ομβρίων. Οι συνιστώμενες (και γενικά μη υπερβατές τιμές είναι)

- 1,5% για διαδρόμους κατηγορίας C, D, E
- 2% για διαδρόμους με κωδικό A, B

και εκτός από τις περιοχές διασταύρωσης διαδρόμου – τροχοδρόμου (που μπορεί να απαιτούνται μικρότερες κλίσεις), η εγκάρσια κλίση συνίσταται να μην είναι μικρότερη από 1,0% .

Ζώνη ασφαλείας διαδρόμου

Περί τον διάδρομο πρέπει να υπάρχει ζώνη ασφαλείας κατάλληλα διαμορφωμένη, που έχει στόχο

- να μειώνει τον κίνδυνο ζημιάς ενός α/φ που εκτρέπεται από τον διάδρομο
- να προστατεύει κάθε α/φ που υπερίπταται είτε κατά την απογείωση, ή την προσγείωση.

Το πλάτος ορίζεται ανάλογα με την κατηγορία του διαδρόμου. Ο πίνακας 5.3 συνοψίζει τις σχετικές κατά ICAO απαιτήσεις.

Κωδικός αριθμός διαδρόμου	Απόσταση εξωτερικού ορίου από τον άξονα του διαδρόμου	
	Μη ενόργανη προσέγγιση	Ενόργανη προσέγγιση
1	30	75
2	40	75
3	75	150
4	75	150

ΠΙΝΑΚΑΣ 5.3 – ΕΛΑΧΙΣΤΕΣ ΑΠΑΙΤΗΣΕΙΣ ΠΛΑΤΟΥΣ ΖΩΝΗΣ ΑΣΦΑΛΕΙΑΣ (m)

Το μήκος της ζώνης εκτείνεται πέραν του διαδρόμου και του τυχόν υπάρχοντος SWY τουλάχιστον κατά

- 60 m για διαδρόμους με κωδικό 2, 3 και 4
- 60 m για διαδρόμους με κωδικό 1 και ενόργανη προσέγγιση
- 30 m για διαδρόμους με κωδικό 1 και προσέγγιση VFR

Όλη η ζώνη πρέπει να είναι ελεύθερη σταθερών εμποδίων, εκτός από οπτικά βοηθήματα κατάλληλα επί εύθραυστης βάσης. Οπωσδήποτε τούτο ισχύει για πλάτος :

- 60 m από τον άξονα του διαδρόμου κατηγορίας 3 ή 4 ακριβούς ενόργανης προσέγγισης
- 45 m ακριβούς ενόργανης προσέγγισης κατηγορίας I για διάδρομο κατηγορίας 1 ή 2

Εντός των ανωτέρω αποστάσεων απαγορεύεται επίσης η ύπαρξη κινητών εμποδίων κατά την προσγείωση ή απογείωση.

Η ζώνη ασφαλείας πρέπει να απισώνεται και έχει ελεγχόμενη γεωμετρία τόσο ως προς τις κατά μήκος κλίσεις όσο και τις εγκάρσιες. Για παράδειγμα για διάδρομο 4 η κατά μήκος κλίση δεν πρέπει να υπερβαίνει τα 1,5% και η εγκάρσια συνιστάται 2,5% για να διασφαλίζεται η απορροή.

ΚΕΦΑΛΑΙΟ 6

ΑΠΟΘΗΚΕΥΣΗ ΚΑΙ ΧΡΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ

Το παρόν κεφάλαιο περιγράφει την μορφή με την οποία τα απαραίτητα δεδομένα αποθηκεύονται στο πρόγραμμα. Με την γενικότερη έννοια «μορφή» εννοείται ο χρησιμοποιούμενος τύπος δεδομένων, σε επίπεδο γλώσσας (κώδικα) καθώς και οι χρησιμοποιούμενες γεωμετρικές απεικονίσεις. Σημειώνουμε ότι κάθε αναφορά σε κώδικα απευθύνεται στη γλώσσα C++.

6.1 – ΔΟΜΗΣΗ ΔΕΔΟΜΕΝΩΝ

Το θεμελιώδες αντικείμενο των γεωμετρικών αλγορίθμων είναι το *σημείο* (*point*) το οποίο θεωρούμε ως τριάδα συντεταγμένων στο τρισδιάστατο καρτεσιανό σύστημα. Κάθε συντεταγμένη (όπως θα δούμε και παρακάτω) αποθηκεύεται σε μεταβλητή τύπου *double*, η οποία έχει μέγεθος 64 bit (σε περιβάλλον των 32-bit) και εύρος από 1,7E-308 έως 1,7E+308.

Η *γραμμή* (*line*) είναι ένα ζευγάρι *σημείων* τα οποία αντιμετωπίζουμε (σε προγραμματιστικό επίπεδο) θεωρώντας πως ενώνονται με ένα ευθύγραμμο τμήμα.

Το *πολύγωνο* είναι μια λίστα (πίνακας) *σημείων*. Υποθέτουμε πως συνεχόμενα *σημεία* ενώνονται με *γραμμές* και πως το τελευταίο *σημείο* ενώνεται με το πρώτο μορφώνοντας ένα κλειστό σχήμα. Στο πρόγραμμα συνεχόμενα είναι τα σημεία που ακολουθούν το ένα το άλλο σε μια αντιωρολογιακή διαγραφή του περιγράμματος του πολυγώνου. Καθώς (όπως θα δούμε και στο κεφάλαιο 8 «ΟΙ ΕΠΙΦΑΝΕΙΕΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ ΣΤΟ ΠΡΟΓΡΑΜΜΑ») οι κορυφές των σχημάτων υπολογίζονται από το πρόγραμμα, η φορά διαγραφής έχει ήδη ληφθεί υπόψη κατά την αρίθμηση των κορυφών στη σχεδίαση των αλγορίθμων. Σε διαφορετική περίπτωση πχ στη δημιουργία κυρτού πολυγώνου από τυχαία σημεία θα έπρεπε να σχεδιαστεί αλγόριθμος εύρεσης κλειστού περιβλήματος (Convex Hull algorithm) όπως ο “Package Wrapping”, που ουσιαστικά είναι αλγόριθμος ταξινόμησης της λίστας *σημείων* ώστε κατά την ανάγνωση συνεχόμενων (στη λίστα) στοιχείων να δημιουργείται το επιθυμητό (κυρτό) σχήμα.

Για να εργαστούμε με τα γεωμετρικά αντικείμενα *σημείο* και *γραμμή* χρησιμοποιούμε την ακόλουθη αναπαράσταση πληροφορίας

- ΣΗΜΕΙΟ:

```
struct mypoint {  
    double x,y,z ;  
    char *description;  
};
```

- ΕΥΘΕΙΑ:

```
struct myline {
    mypoint p1, p2;
};
```

Η δεσμευμένη λέξη `struct` αναφέρεται στον τύπο δεδομένων «δομή» της C++. Στη C++ η δομή ορίζει νέο τύπο δεδομένων και υποστηρίζει τον αντικειμενοστραφή προγραμματισμό αντίθετα με την C. Ως συγκεντρωτικός τύπος δεδομένων (aggregate data type) περιλαμβάνει τα ακόλουθα μέλη:

- Στο σημείο :
 - Τρεις μεταβλητές τύπου `double` που αποθηκεύουν τις συντεταγμένες του σημείου
 - Έναν δείκτη σε πίνακα χαρακτήρων προς αποθήκευση αλφαριθμητικών που αφορούν το σημείο (λεκτικές πληροφορίες)
- Στην ευθεία :
 - Δύο μεταβλητές τύπου `mypoint` (η δομή που ορίστηκε προηγούμενα)

Για παράδειγμα ο ορισμός του σημείου «κέντρο του πρώτου διαδρόμου» (RUNWAY 1) γίνεται ως εξής: `mypoint aRW_M ;`

Η προσπέλαση των μελών μιας δομής γίνεται με τον τελεστή τελεία (.) έτσι η ακόλουθη ανάθεση υλοποιεί το σημείο `aRW_M`:

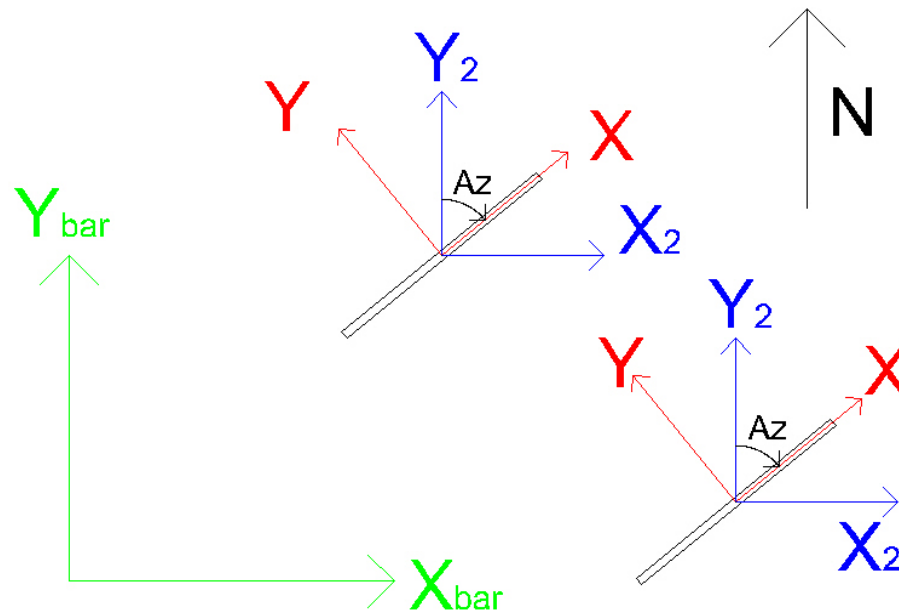
```
aRW_M.x=0;
aRW_M.y=0;
aRW_M.z=0;
aRW_M.description= "Centre of the runway" ;
```

Για την αναπαράσταση *πολυγώνων* χρησιμοποιούμε πίνακες δομών `mypoint`. Ως παράδειγμα δηλώνουμε την επιφάνεια προσέγγισης του πρώτου διαδρόμου:

```
mypoint approach[6] ;
```

Η δήλωση αυτή κατασκευάζει έναν πίνακα έξι στοιχείων τύπου (δομής) `mypoint`. Τα έξι στοιχεία του πίνακα τοποθετούνται στις θέσεις `approach[0]` ως `approach[5]`. Τα στοιχεία `approach[1]` ως `approach[4]` αποθηκεύουν τις συντεταγμένες των κορυφών του τετραπλεύρου ενώ τα `approach[0]` και `approach[5]` χρησιμεύουν στην ροή του προγράμματος (έλεγχος βρόγχων κλπ) και ως τιμές «φύλακες» (sentinel values $\rightarrow p[0]=p[N] \ \& \ p[1]=p[N+1]$). Γενικά για πολυγώνο p , N κορυφών χρησιμοποιούμε τη δήλωση `mypoint p[N+2]`.

6.2 – ΣΥΣΤΗΜΑΤΑ ΣΥΝΤΕΤΑΓΜΕΝΩΝ



ΣΧΗΜΑ 6.1

Χρησιμοποιούνται τα εικονιζόμενα 5 συστήματα συντεταγμένων που ανήκουν σε τρεις ουσιαστικά κατηγορίες:

- I. Τοπικές συντεταγμένες. Ορίζονται δύο καρτεσιανά συστήματα (X, Y) τοπικών συντεταγμένων (ένα για κάθε διάδρομο) με τον ακόλουθο τρόπο: Το κέντρο του συστήματος τοποθετείται στο κέντρο του διαδρόμου και ο άξονας των X είναι ευθυγραμμισμένος με τον άξονα του διαδρόμου με την θετική φορά να δείχνει προς την επιφάνεια απογείωσης (διάδρομος προσγείωσης αριστερά του διαδρόμου απογείωσης).
- II. Τοπικές περιστραμμένες συντεταγμένες. Έχουν την κατάληξη 2 και προκύπτουν από την περιστροφή $(X, Y) \rightarrow (X_2, Y_2)$ ώστε ο Y_2 άξονας να έρθει σε παράλληλη θέση με τον βορρά. Κάθε διάδρομος απεικονίζει την σχετική με αυτόν γεωμετρία στο δικό του περιστραμμένο σύστημα συντεταγμένων, δηλαδή υπάρχουν δύο περιστραμμένα συστήματα συντεταγμένων (X_2, Y_2)
- III. Καθολικές συντεταγμένες. Έχουν την κατάληξη bar και η απεικόνιση σε αυτές είναι κοινή για σημεία που αφορούν και τους δύο διαδρόμους

(αυτός άλλωστε είναι και ο σκοπός ενός καθολικού συστήματος). Ο άξονας Y_{bar} είναι παράλληλος με τον άξονα του Borρά.

Όλα τα συστήματα συντεταγμένων έχουν άξονα Z ο οποίος ορίζεται κατά την σύμβαση θετικής φοράς : προχωράμε από $X \rightarrow Y \rightarrow Z$ με γωνίες 90° και φορά αντίθετη της κίνησης των δεικτών του ρολογιού. Ο Z άξονας δεν επηρεάζεται από τους οριζόντιους περιστροφικούς μετασχηματισμούς.

Με την παραπάνω δήλωση συστημάτων συντεταγμένων μπορούμε να προγραμματίσουμε τις επιφάνειες αποκλεισμού εμποδίων στο τοπικό τους σύστημα (επί του οποίου και έχουμε γνώση των σχετιζομένων γεωμετρικών παραμέτρων και μεταβλητών) και με χρήση των γεωμετρικών μετασχηματισμών περιστροφή και μετατόπιση (βλ. κεφ. 7 «ΓΕΩΜΕΤΡΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ») να καταλήγουμε στο καθολικό σύστημα (επί του οποίου δηλώνεται και το σημείο του οποίου το επιτρεπτό υψόμετρο υπολογίζουμε). Ακόμα πετυχαίνουμε η γωνία των αξόνων X και Y_2 και το αζιμούθιο (Az) να

συμπίπτουν πάντα κατά μέτρο: $|Az| = \left| X \hat{O} Y_2 \right|$

ενώ η γωνία στροφής $(X, Y) \rightarrow (X_2, Y_2)$ έστω faxis ικανοποιεί πάντα τη σχέση $afaxis = -(Azim+90)$, με $Azim = -Az$

6.3 – ΟΝΟΜΑΤΟΛΟΓΙΑ ΜΕΤΑΒΛΗΤΩΝ

Στον κώδικα χρησιμοποιούνται κατάλληλα προθέματα, καταλήξεις και συντομογραφίες για την αποτελεσματική δήλωση των μεταβλητών ώστε ο σκοπός και η πληροφορία της μεταβλητής να γίνεται διαισθητικά αναγνωρίσιμη και ο κώδικας εύκολα επισκέψιμος και διορθώσιμος.

Τα προθέματα a, b δηλώνουν πως η μεταβλητή αναφέρεται στους διαδρόμους 1 & 2 αντίστοιχα. Για παράδειγμα η δήλωση της επιφάνειας προσέγγισης στους δύο διαδρόμους είναι :

aapproach[6] (RUNWAY 1) bapproach[6] (RUNWAY 2)

Ανάλογα με το σύστημα συντεταγμένων στο οποίο αναφέρεται η μεταβλητή χρησιμοποιούνται κατάλληλες καταλήξεις, για παράδειγμα το κέντρο του πρώτου διαδρόμου θα ναι

aRW_M \rightarrow στο τοπικό σύστημα

aRW_M2 \rightarrow στο τοπικό περιστραμμένο σύστημα

aRW_Mbar \rightarrow στο καθολικό σύστημα (στοιχείο που εισάγεται στο GUI - Graphical user interface - του προγράμματος)

Χρησιμοποιείται επίσης η κατάληξη 180 με την ακόλουθη σημασία:

aRW_M180 → συμμετρικό ως προς την αρχή των αξόνων του aRW_M (στο συγκεκριμένο σημείο δεν έχει αξία η λήψη του συμμετρικού, αλλά θα δούμε αργότερα πως αυτό λύνει το πρόβλημα της λειτουργίας διαδρόμου και στις δύο κατευθύνσεις για απογείωση-προσγείωση)

Τέλος γίνεται χρήση των ακόλουθων συντομεύσεων

RW → Διάδρομος (Runway)

AP → Approach (Προσέγγιση)

TO → Take Off (Απογείωση)

TR → Transitional (Μεταβατική)

_s → slope (Κλίση)

Παρατήρηση: Θα φάνηκε από τα ως τώρα γραφόμενα, πως στην ονοματολογία δεδομένων και μεταβλητών χρησιμοποιούμε δηλώσεις ελαφρώς διαφορετικές από τις προφανείς, έτσι για παράδειγμα το σημείο (point) αντί “point”, δηλώνεται ως “mypoint”. Αυτό γίνεται επειδή στα CLR (Common Language Runtime) προγράμματα, όπως αυτό που κατασκευάζουμε, υπάρχει μεγάλος αριθμός δεσμευμένων λέξεων με αποτέλεσμα τον κίνδυνο δημιουργίας ασάφειας κατά τη μεταγλώττιση (compile). Το περιβάλλον της Visual C++ παρέχει μεγάλο αριθμό τύπων δεδομένων, το κόστος αυτής της ευκολίας όμως είναι να απαιτείται από τον χρήστη-προγραμματιστή μια γνώση επαρκούς αριθμού αυτών ώστε να αποφεύγονται τέτοιοι σκόπελοι. Βέβαια θα παρατηρούσε κανείς πως τα δεδομένα αυτά είναι αποθηκευμένα σε τάξεις (classes) ενώ εμείς χρησιμοποιούμε δομές (structures), όμως και αυτό γίνεται σκόπιμα για να μην καταλήξουμε στην ίδια ασάφεια που περιγράψαμε πιο πάνω· να αναφέρουμε πάντως πως η ασάφεια δεν απειλεί αποκλειστικά τον μεταγλωττιστή αλλά και τον προγραμματιστή αφού στις περιπτώσεις που η χρήση των προκαθορισμένων από το περιβάλλον μεταβλητών γίνει επιτακτική (πχ για τη δημιουργία γραφικών απαιτούνται οι τύποι point, line κοκ) παρότι η αναφορά σε διαφορετικούς τύπους δεδομένων επιτρέπει την μεταγλώττιση, η ύπαρξη διπλών δηλώσεων δυσχεραίνει την εποπτεία επί του προγράμματος (πχ «παρακολούθηση» μεταβλητών κατά το debug).

ΚΕΦΑΛΑΙΟ 7

ΓΕΩΜΕΤΡΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Για την κατασκευή του προγράμματος rPORT χρειάστηκε να δοθούν λύσεις σε γεωμετρικά προβλήματα, τα οποία σε προγραμματιστικό περιβάλλον αποκτούν σημαντικές διαστάσεις. Για παράδειγμα το πρόβλημα περίληψης σημείου σε πολύγωνο («δοθέντος σημείου και πολυγώνου, βρείτε αν το σημείο βρίσκεται εντός του πολυγώνου») ενώ από ένα άτομο μπορεί να λυθεί εποπτικά, απαιτεί υπολογίσιμους προγραμματιστικούς πόρους για την αντίστοιχη υπολογιστική του υλοποίηση. Θα δούμε εδώ ότι για πολλά ζητήματα η ενδεικνύμενη υπολογιστική υλοποίηση είναι αρκετά διαφορετική από την διαισθητική λύση που θα έδινε ένα άτομο.

Η επιστημονική περιοχή της πληροφορικής που μελετά γεωμετρικά προβλήματα και τις αλγοριθμικές τους λύσεις καλείται υπολογιστική γεωμετρία. Περιλαμβάνει τον σχεδιασμό μεθόδων επίλυσης αλλά και τεχνικές για την σθεναρή υλοποίηση τους. Σημειώνουμε εδώ πως μέθοδο (method) καλούμε την ιδέα και τις μαθηματικές έννοιες πίσω από μια επίλυση και την διακρίνουμε από την υλοποίηση (implementation) που αποτελεί την έκφραση της ιδέας σε γλώσσα προγραμματισμού, κάτι που πολλές φορές μπορεί να αποδειχθεί αρκετά επίπονο (ειδικά αν η μέθοδος δεν είναι εξαντλητικά προνοητική).

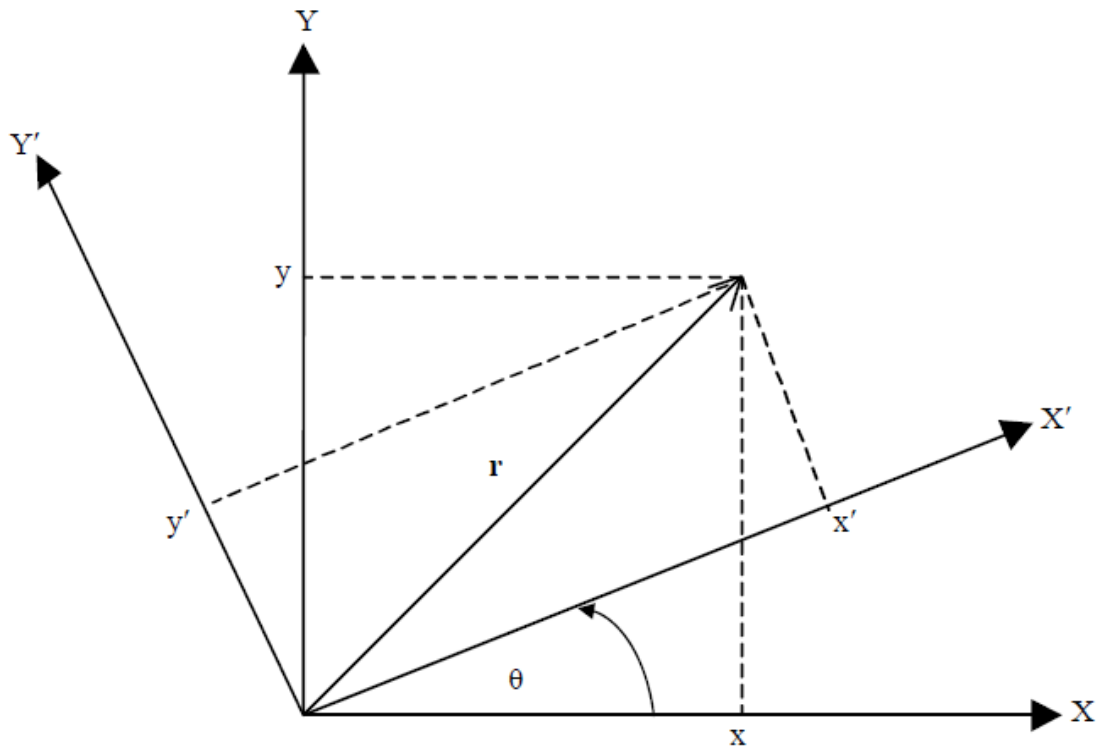
Στο παρόν κεφάλαιο καταγράφονται οι γεωμετρικοί αλγόριθμοι που σχεδιάστηκαν κατά την ανάπτυξη του προγράμματος rPORT από τους βασικούς μέχρι τους πιο πολύπλοκους, οι οποίοι αποτελούν μια σύνθεση των προηγούμενων σε μια λειτουργική ολότητα. Το κεφάλαιο αποτελεί το ερευνητικό κομμάτι της εργασίας για την περάτωσή του οποίου εξετάστηκαν διάφορες λύσεις και προσεγγίσεις και επιλέχθηκαν οι πιο αποτελεσματικές και προγραμματιστικά «οικονομικές». Σημειώνουμε τέλος πως τα κομμάτια κώδικα που παρατίθενται είναι ελαφρώς διαφορετικά από του παραρτήματος (κώδικας που χρησιμοποιείται) αφού σκοπό έχουν την κατανόηση των αλγορίθμων από τον αναγνώστη σε αντίθεση από τον σε χρήση κώδικα που προνοεί για την εποπτική του αναγνώριση και άμεσο έλεγχο-επιδιόρθωση (debug) από τον προγραμματιστή.

7.1– ΓΕΩΜΕΤΡΙΚΟΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ

Με τον όρο γεωμετρικοί μετασχηματισμοί αναφερόμαστε στις μεθόδους που μας επιτρέπουν να απεικονίσουμε ένα σημείο (και κατ' επέκταση ένα σχήμα) σε διαφορετικά συστήματα συντεταγμένων τα οποία έχουν συγκεκριμένες τοπολογικές σχέσεις μεταξύ τους. Συνηθέστεροι μετασχηματισμοί είναι η περιστροφή, η μετατόπιση, η μεγέθυνση-σμίκρυνση, η κύρτωση και ο αντίστροφος μετασχηματισμός. Χρησιμοποιούμε και αναλύουμε τους πρώτους δύο.

- Περιστροφή (rotation)

Όταν ένα σύστημα συντεταγμένων (X', Y') προκύπτει από την περιστροφή συστήματος (X, Y) κατά γωνία θ τότε τυχαίο σημείο $T(x, y)$ θα απεικονίζεται στο νέο σύστημα με το σημείο $T'(x', y')$ όπως εικονίζεται παρακάτω:



ΣΧΗΜΑ 7.1

Το διάνυσμα θέσεως παραμένει σταθερό, επομένως οι σχέσεις που συνδέουν τις συντεταγμένες του διανύσματος \mathbf{r} ως προς τα δυο αυτά συστήματα συντεταγμένων είναι:

$$x' = x \cos \theta + y \sin \theta \quad , \quad y' = -x \sin \theta + y \cos \theta \quad (1)$$

Για την αναπαράσταση γεωμετρικών μετασχηματισμών στα συστήματα γραφικών χρησιμοποιείται η αναπαράσταση σημείων και διανυσμάτων σε ομογενείς συντεταγμένες. Σε ένα σύστημα ομογενών συντεταγμένων στο διδιάστατο επίπεδο για την αναπαράσταση ενός σημείου P (x_p, y_p) χρησιμοποιείται μια τριάδα αριθμών $(x_p, y_p, 1)$ και οι μετασχηματισμοί έχουν τη μορφή:

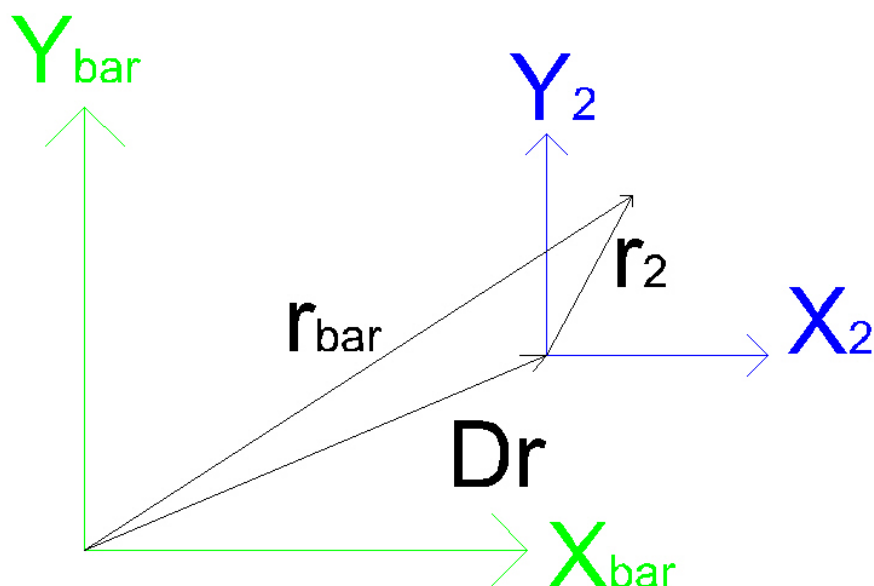
$$(x', y', 1) = (x, y, 1) * M \quad (2)$$

όπου ο M είναι πίνακας 3×3 που στην περίπτωση της περιστροφής γίνεται:

$$M = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Μετατόπιση (translation)

Χρησιμοποιείται για να απεικονίσουμε σημείο $T(x_2, y_2)$ του συστήματος (X_2, Y_2) σε σύστημα (X_{bar}, Y_{bar}) που έχει σχετική θέση ως προς το πρώτο η οποίο ορίζεται από το διάνυσμα \vec{Dr} :



ΣΧΗΜΑ 7.2

Οι συντεταγμένες στο νέο σύστημα υπολογίζονται από την διανυσματική ισότητα **(3)**:

$$\vec{r}_{bar} = \vec{Dr} + \vec{r}_2$$

ενώ ο πίνακας M παίρνει τη μορφή

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & dy & 1 \end{bmatrix}$$

Στο πρόγραμμα το προς έρευνα επιτρεπτού υψομέτρου σημείο, εισάγεται σε καθολικές συντεταγμένες και στη συνέχεια μεταφέρεται στο τοπικό σύστημα κάθε διαδρόμου. Όταν αναφέρεται στον πρώτο διάδρομο καλείται

“aScoutbar” (αντίστοιχα bScoutbar για τον δεύτερο) και οι μετασχηματισμοί στους οποίους υποβάλλεται είναι :

aScoutbar → aScout2 → aScout → aScout180

καθολικό → τοπικό περιστραμμένο → τοπικό → τοπικό

μετατόπιση – περιστροφή – λήψη αντισυμμετρικού

(για εξήγηση καταλήξεων βλ. κεφ. 6 «ΑΠΟΘΗΚΕΥΣΗ ΚΑΙ ΧΡΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ» παρ.3 «ΟΝΟΜΑΤΟΛΟΓΙΑ ΜΕΤΑΒΛΗΤΩΝ»)

Αυτό γίνεται επειδή οι επιφάνειες αποκλεισμού εμποδίων εκφράζονται στο τοπικό τους σύστημα και έτσι θεωρήθηκε υπολογιστικά οικονομικότερο να μεταφέρουμε το σημείο από το καθολικό στο τοπικό τους σύστημα παρά όλες τις συντεταγμένες των επιφανειών από το τοπικό τους στο καθολικό σύστημα. Για το λόγο αυτό η γωνία στροφής faxis (γωνία $(X,Y) \rightarrow (X2,Y2)$) λαμβάνεται με την αντίθετη της τιμή, επειδή θεωρούμε την αντίστροφη περιστροφή.

Θυμίζουμε επίσης ότι προς μελέτη και των δύο διευθύνσεων ως χρησιμοποιούμενων για απογείωση και προσγείωση, δεν περιστρέφονται οι επιφάνειες (όπως ορίζονται στο κεφ. 8 «ΟΙ ΕΠΙΦΑΝΕΙΕΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ ΣΤΟ ΠΡΟΓΡΑΜΜΑ») κατά 180° αλλά το προς ερευνα σημείο, κάτι το οποίο μπορεί να γίνει με απλή θεώρηση του συμμετρικού του ως προς την αρχή των αξόνων. Ο αλγόριθμος που περιγράφει τις παραπάνω ενέργειες επί του σημείου aScoutbar (για bScoutbar έχουμε αντίστοιχα το πρόθεμα b) είναι :

```
void asetScout()
{
    //Actual point manipulation algorithm
    double PI = 3.14159265 ;
    double a = - (afaxis*PI) / (180.0); //faxis is angle
    //local->2 so angle
    //2->local = - aaxis(radians)
    // translation
    aScout2.x = aScoutbar.x - aRW_Mbar.x ;
    aScout2.y = aScoutbar.y - aRW_Mbar.y ;
    // rotation
    aScout.x = (aScout2.x*cos(a))+(aScout2.y*sin(a)) ;
    aScout.y = (aScout2.y*cos(a))-(aScout2.x*sin(a)) ;
    // reflection of point aScout
    aScout180.x = -aScout.x ;
    aScout180.y = -aScout.y ;
}
```

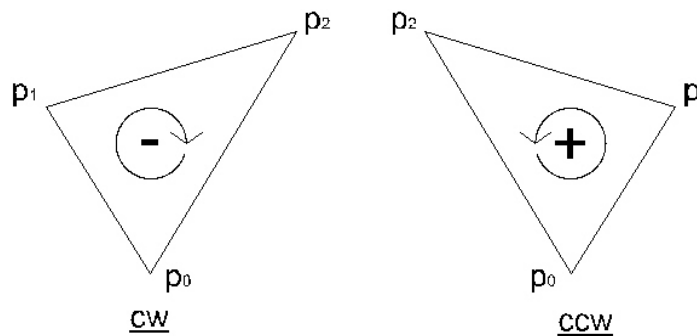
(παράτίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

Τέλος αναφέρουμε πως ο χειρισμός του άξονα Z, περιγράφεται ως προς τον τρόπο και την οπτική του ξεχωριστά στην 5^η παράγραφο.

7.2 – ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΣ ΤΡΙΩΝ ΣΗΜΕΙΩΝ

Το κατηγορημα προσανατολισμού *ccw* είναι από τα πιο βασικά και χρήσιμα εργαλεία της υπολογιστικής γεωμετρίας. Κατηγορημα (predicate) καλείται ο έλεγχος μιας ιδιότητας πάνω σε ορισμένα γεωμετρικά δεδομένα. Η έξοδος του κατηγορήματος παίρνει συνήθως δύο ή τρεις διακριτές τιμές· η εκδοχή τριπλής εξόδου υιοθετείται και εδώ.

Το συγκεκριμένο κατηγορημα έχει ως όρισμα (δεδομένα εισόδου) τρία σημεία στο επίπεδο, τα οποία συμβολίζουμε με p_0, p_1, p_2 , και υπολογίζει τον προσανατολισμό τους (orientation), ο οποίος δηλώνει την φορά της στροφής που ορίζουν τα σημεία p_0, p_1, p_2 σε αυτή τη σειρά. Με άλλα λόγια το κατηγορημα φανερώνει σε πια πλευρά της ευθείας που ορίζουν τα p_0, p_1 κείται το p_2 . Όταν η στροφή είναι σύμφωνη με τους δείκτες τους ρολογιού (CW – Clock wise) θεωρείται αρνητική ενώ όταν είναι αντίθετη (CCW – Counter Clock wise) θεωρείται θετική.



ΣΧΗΜΑ 7.3

Ο αλγόριθμος που υλοποιεί το κατηγορημα (συνάρτηση *ccw*) έχει ως εξής:

```
int ccw(my point p0, my point p1, my point p2)
{
    double dx1, dx2, dy1, dy2 ;

    dx1 = p1.x - p0.x ; // Δx των σημείων pa & p0
    dy1 = p1.y - p0.y ; // Δy των σημείων pa & p0
    dx2 = p2.x - p0.x ; // Δx των σημείων pb & p0
    dy2 = p2.y - p0.y ; // Δy των σημείων pb & p0

    if (dx1*dy2 > dy1*dx2) return +1; // CCW
    if (dx1*dy2 < dy1*dx2) return -1; // CW
    if ( ( dx1*dx2 < 0 ) || ( dy1*dy2 < 0 ) ) return -1; // p0 ∈ (p2p1)
    if ( ( dx1*dx1 + dy1*dy1 ) < ( dx2*dx2 + dy2*dy2 ) ) return +1 ;
    // p1 ∈ (p0p2)
    return 0; // p2 ∈ (p0p1)
```

}

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

Για να καταλάβουμε πως λειτουργεί ως υποθέσουμε αρχικά πως όλες οι ποσότητες $dx1, dy1, dx2, dy2$ είναι θετικές. Έπειτα προσέξτε ότι η κλίση της ευθείας που συνδέει τα $p0$ και $p1$ είναι $dy1/dx1$ και η κλίση της ευθείας $p0p2$ είναι $dy2/dx2$. Αν η κλίση της δεύτερης ευθείας είναι μεγαλύτερη από αυτή της πρώτης, απαιτείται μια «αριστερή» (CCW) στροφή στη διαδρομή $p0 \rightarrow p1 \rightarrow p2$. Στην αντίθετη περίπτωση απαιτείται μια «δεξιά» (CW) στροφή. Το να συγκρίνουμε κλίσεις είναι κάπως άβολο για το πρόγραμμα καθώς οι γραμμές μπορεί να είναι κατακόρυφες ($dx1$ ή $dx2$ ίσο με 0) και γι' αυτό τις πολλαπλασιάζουμε με $dx1 * dx2$. Αποδεικνύεται πως οι κλίσεις δεν είναι ανάγκη να ναι θετικές για να λειτουργήσει αυτός ο έλεγχος.

Υπάρχει όμως μια ακόμα καθοριστική παράλειψη στην παραπάνω λογική: αγνοεί την περίπτωση όπου όλες οι κλίσεις είναι ίδιες (περίπτωση συνευθειακών σημείων). Επιλέγουμε να υπερκεράσουμε το εμπόδιο αυτό με τον εξής τρόπο: Το κατηγορήμα γίνεται τριπλής εξόδου, δηλαδή αντί να επιστρέφει 0 ή 1 (για CCW ή CW), χρησιμοποιούμε έξοδο 1 και -1, κρατώντας την τιμή 0 για την περίπτωση όπου το $p2$ είναι επί του ευθύγραμμου τμήματος ($p0 p1$). Οι περιπτώσεις και η αντίστοιχη έξοδος που παράγεται είναι:

CCW $\rightarrow +1$
 CW $\rightarrow -1$
 $p2 \in (p0p1) \rightarrow 0$
 $p0 \in (p2p1) \rightarrow -1$
 $p1 \in (p0p2) \rightarrow 1$

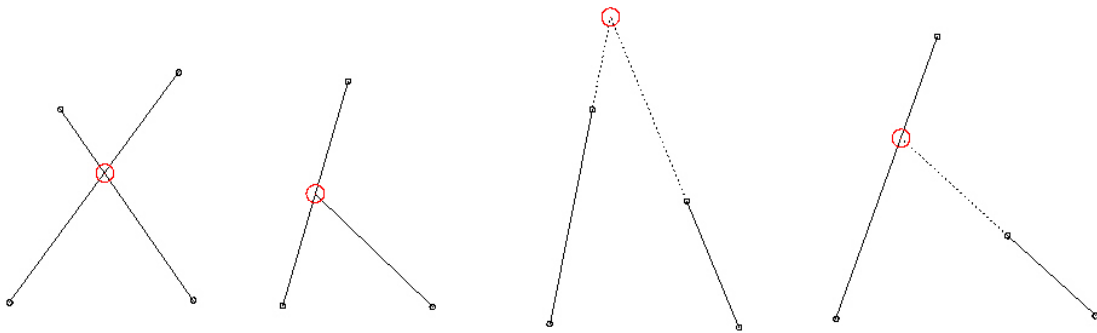
Η κατηγοριοποίηση αυτή προστέθηκε και στα σχόλια (//) του κώδικα πιο πάνω. Θα δούμε ότι αυτή η σύμβαση απλοποιεί τον κώδικα συναρτήσεων που χρησιμοποιούν την ccw

7.3 – ΔΙΑΤΑΞΕΙΣ ΕΥΘΥΓΡΑΜΜΩΝ ΤΜΗΜΑΤΩΝ – ΕΛΕΓΧΟΣ ΤΟΜΗΣ

Το κεφάλαιο των διατάξεων (ή διευθετήσεων, arrangements) είναι ευρύτατο. Με τον όρο διατάξεις εννοούμε σύνολα αντικειμένων κάποιου είδους, τα οποία θέλουμε να αποτυπώσουμε με αποτελεσματικό τρόπο ώστε να επιτρέπουμε ορισμένες ταχείες λειτουργίες πάνω σε αυτά. Για παράδειγμα η διάταξη ευθυγράμμων τμημάτων στο επίπεδο συνίσταται στον υπολογισμό όλων των

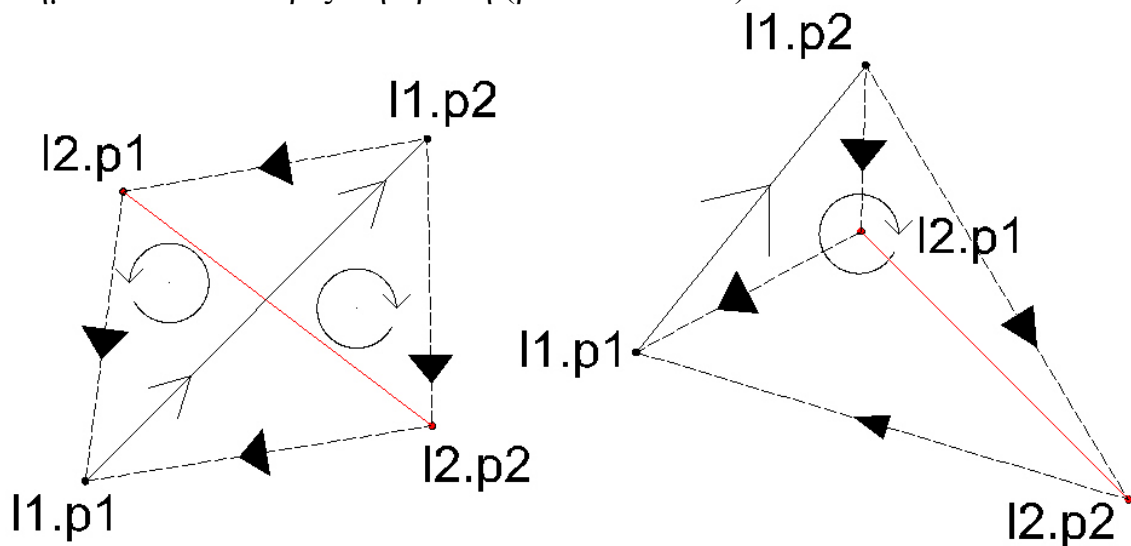
τομών μεταξύ τους και στον υπολογισμό της σχετικής θέσης των αντικειμένων της διάταξης.

Για τις ανάγκες της εργασίας η επιθυμητή λειτουργία επί της διάταξης ευθυγράμμων τμημάτων είναι ο έλεγχος τομής : δοσμένων δύο ευθυγράμμων τμημάτων, αποφανθείτε αν τέμνονται ή όχι. Ο ευθύς τρόπος επίλυσης αυτού είναι να βρούμε (από εξισώσεις διαφορικής γεωμετρίας) το σημείο τομής των δοσμένων τμημάτων και να ελέγξουμε αν το σημείο αυτό κείται επί των τμημάτων (ικανοποιεί την εξίσωση ευθείας τουλάχιστον ενός από τα δύο)



ΣΧΗΜΑ 7.4

Ωστόσο οι μέθοδοι διαφορικής γεωμετρίας αποδεικνύονται μη ικανοποιητικοί και χρονοβόροι στις προγραμματιστικές υλοποιήσεις τους. Αντί αυτών χρησιμοποιούμε το κατηγορημα προσανατολισμού με τον ακόλουθο τρόπο: Θεωρούμε ότι δύο γραμμές l_1 και l_2 με άκρα $l_1.p_1$, $l_1.p_2$ και $l_2.p_1$, $l_2.p_2$ τέμνονται όταν τα άκρα της δεύτερης γραμμής βρίσκονται σε διαφορετικό από τα ημιεπίπεδα που ορίζει η πρώτη (βλ. ΣΧΗΜΑ 5).



ΣΧΗΜΑ 7.5

Επομένως το κατηγορημα προσανατολισμού ccw θα παράγει εξόδους αντιθέτου προσήμου (+1 & -1) στην περίπτωση ύπαρξης τομής (διαφορετικές φορές διαγραφής) και ίδιου προσήμου στην περίπτωση μη ύπαρξης τομής για όρισμα $p0, p1$ τα άκρα της πρώτης ευθείας (πχ $l1.p1, l1.p2$) και $p2$ τα άκρα της δεύτερης (ο έλεγχος γίνεται δύο φορές για καθένα από τα άκρα της δεύτερης πχ πρώτα για $l2.p1$ και έπειτα για $l2.p2$). Καθώς το ccw είναι τριπλής εξόδου, χειριζόμαστε την έξοδο «0» με χρήση του \leq για την περίπτωση που το σημείο τομής κείται επί ενός εκ των δύο τμημάτων (πχ $l1.p1, l1.p2$ και $l2.p1$ συνευθειακά). Η μέθοδος αυτή υλοποιείται με την συνάρτηση `intersect`:

```
int intersect(myline la, myline lb)
{
    return(
        (ccw(la.p1, la.p2, lb.p1)*ccw(la.p1, la.p2, lb.p2))<=0)
        &&
        (ccw(lb.p1, lb.p2, la.p1)*ccw(lb.p1, lb.p2, la.p2) )<=0)
    );
}
```

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

Η `intersect` παράγει αληθή έξοδο όταν υπάρχει τομή και ψευδή (0) όταν δεν υπάρχει. Αυτή η λύση δείχνει να απαιτεί αρκετούς υπολογισμούς αναλογικά με την έκταση του προβλήματος ωστόσο προνοεί για όλες τις πιθανές περιπτώσεις. Για παράδειγμα, αν και τα τέσσερα άκρα των ευθειών είναι συγγραμικά υπάρχουν έξι διαφορετικές περιπτώσεις (συν τις περιπτώσεις όπου έχουμε συμπίπτοντα άκρα), τέσσερις μόνο εκ των οποίων δίνουν τομή. Τέτοιες ειδικές περιπτώσεις είναι η αχίλλειος πτέρνα των γεωμετρικών αλγορίθμων· δεν μπορούμε να τις αποφύγουμε, μπορούμε ωστόσο να τις χειριστούμε με στοιχειώδη κατηγορήματα όπως το ccw .

7.4 – ΠΕΡΙΛΗΨΗ ΣΕ ΕΠΙΦΑΝΕΙΑ

Το πρόβλημα περίληψης σημείου σε επιφάνεια μελετάται σε οριζόντια προβολή και μεταπίπτει έτσι στο πρόβλημα περίληψης σημείου σε επίπεδο σχήμα. Απαιτείται η εξέταση των δύο κάτωθι περιπτώσεων.

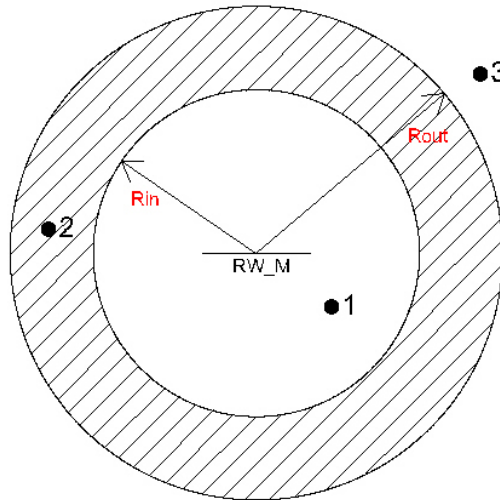
I. ΠΕΡΙΛΗΨΗ ΣΕ ΚΥΚΛΟ

Χρησιμεύει για να αποφανθούμε περί της περίληψης του σημείου στις επιφάνειες εσωτερική-εξωτερική οριζόντια και κωνική. Καθώς ο κύκλος αποτελεί το γεωμετρικό τόπο των σημείων που απέχουν σταθερή απόσταση

από κάποιο σημείο (κέντρο) υπολογίζουμε αρχικά την απόσταση του προς έρευνα σημείου από το κέντρο του κύκλου (κέντρο διαδρόμου = (0,0) στο τοπικό σύστημα) έστω distance και διακρίνουμε τις περιπτώσεις:

(βλ. ΣΧΗΜΑ 6)

1. $distance \leq R_{in} \rightarrow$ σημείο στην εσωτερική οριζόντια επιφάνεια
2. $R_{in} \leq distance \leq R_{out} \rightarrow$ σημείο στην κωνική επιφάνεια
3. $distance \geq R_{out} \rightarrow$ σημείο στην εξωτερική οριζόντια επιφάνεια



ΣΧΗΜΑ 7.6

Τα παραπάνω υλοποιούνται με τη συνάρτηση cyclic η οποία μεριμνεί και για την περίπτωση που το σημείο βρίσκεται στην μεταβατική επιφάνεια και δίνει μη περίληψη στην εσωτερική οριζόντια παρότι $distance \leq R_{in}$. Παραθέτουμε τον αλγόριθμο της cyclic για τα σχήματα του 1^{ου} διαδρόμου (πρόθεμα α):

```
void cyclic(mypoint t, int rin, int rout)
{
    aisINconical=false ;
    aisINinner=false;
    aisINouter=false;
    double distance = (t.x*t.x)+(t.y*t.y) ;
    distance = sqrt(distance) ;

    if(distance <= rin &&
        !(aisINsafetyzone||aisINtransitional)) {
        aisINinner=true ; return;}

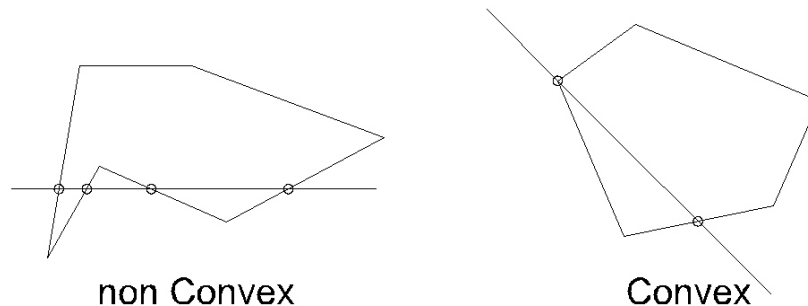
    if(distance>= rout){ aisINouter=true ; return;}
    if(distance<rout && distance>rin) aisINconical=true ;
}
```

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

II. ΠΕΡΙΛΗΨΗ ΣΕ ΠΟΛΥΓΩΝΟ

Οι υπόλοιπες επιφάνειες είναι πολυγωνικές, στην περίπτωση τους όμως το πρόβλημα είναι πιο πολύπλοκο αφού τυχαίο πολύγωνο δεν αποτελεί γεωμετρικό τόπο, επομένως δεν υπάρχουν μετρικές σχέσεις για τον έλεγχο της περίληψης. Ο όρος «τυχαίο πολύγωνο» χρησιμοποιείται ως προς τις διαστάσεις και όχι ως προς τις ιδιότητες των σχημάτων, αφού οι επιφάνειες περιορισμού εμποδίων (και οι οριζόντιες προβολές αυτών) μοιράζονται ένα πολύ σημαντικό γνώρισμα, την κυρτότητα.

Κυρτό (convex) καλείται ένα πολύγωνο, όταν δύναται να έχει το πολύ δύο τομές με οποιαδήποτε γραμμή. Σε ένα κυρτό πολύγωνο κάθε σημείο του είναι ορατό από κάθε άλλο. Αντίστοιχα ορίζεται το μη κυρτό (non convex) πολύγωνο, όπως φαίνεται στο σχήμα 7:

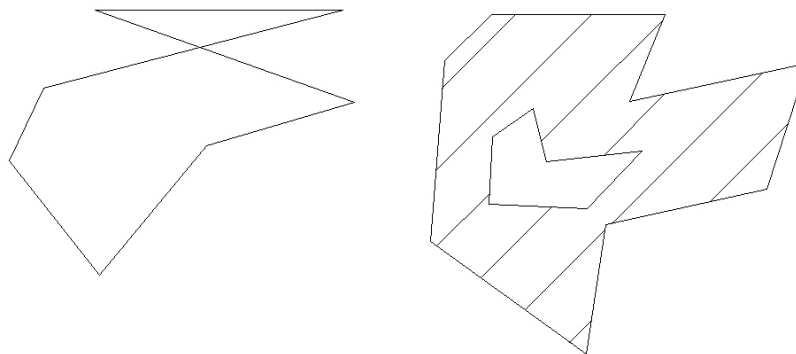


ΣΧΗΜΑ 7.7

Ο αλγόριθμος που θα περιγράψουμε εδώ δεν περιορίζεται σε κυρτά πολύγωνα αλλά γενικεύει τον έλεγχο τομής στα απλά. Ένα πολύγωνο είναι απλό όταν:

- Η πολυγωνική γραμμή που το ορίζει περιέχει γραμμές που τέμνονται μόνο στις κορυφές τους
- Δεν περιέχει αίθρια ή τρύπες, δηλαδή κάθε κλειστή καμπύλη μπορεί να συρρικνωθεί τοπολογικά σε ένα σημείο παραμένοντας στο εσωτερικό του πολυγώνου.

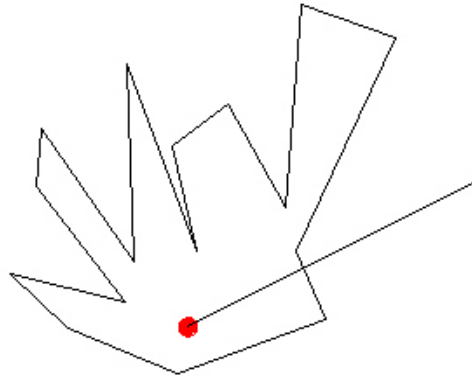
Δύο μη απλά πολύγωνα τα οποία παραβιάζουν τις παραπάνω συνθήκες, φαίνονται στο σχήμα 8



ΣΧΗΜΑ 7.8

Δοσμένου σημείου και (απλού) πολυγώνου, ακολουθούμε την παρακάτω λογική για να αποφανθούμε αν το σημείο βρίσκεται εντός του πολυγώνου:

Φέρουμε γραμμή «απείρου» μήκους από το σημείο προς οποιαδήποτε διεύθυνση (ουσιαστικά απαιτείται γραμμή αρκετά μεγάλη, ώστε να εξασφαλίζεται πως το σημείο τέλους της βρίσκεται εκτός του πολυγώνου) και μετράμε τον αριθμό των τομών της γραμμής με το πολύγωνο. Αν ο αριθμός αυτός είναι περιττός, το σημείο είναι εντός, αν είναι άρτιος είναι εκτός του πολυγώνου. (βλ. σχήμα 9)



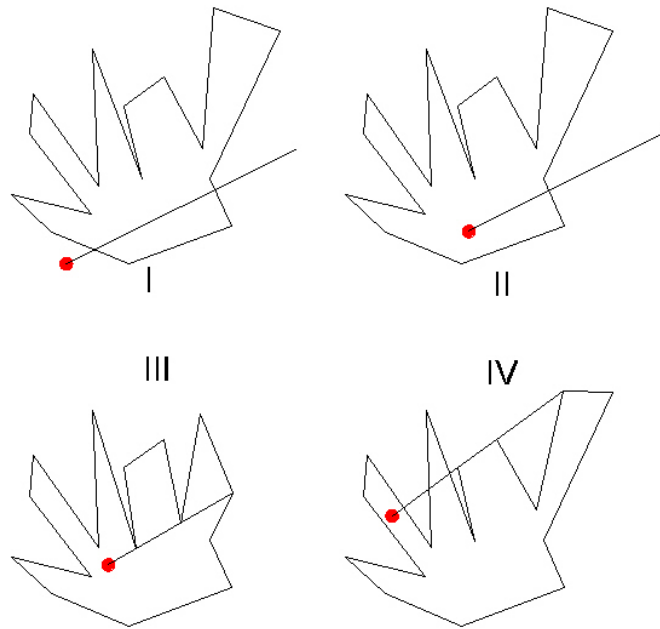
ΣΧΗΜΑ 7.9

Η εγκυρότητα της λογικής αυτής φαίνεται καταγράφοντας τι συμβαίνει καθώς διαγράφουμε την δοκιμαστική γραμμή από το τέλος προς την αρχή της: μετά το πρώτο σημείο τομής είμαστε εντός του πολυγώνου, μετά το δεύτερο είμαστε εκτός και ούτω καθ' εξής. Για περιττό αριθμό τομών, το (προς έλεγχο) σημείο αρχής θα βρίσκεται αναγκαστικά εντός του πολυγώνου.

Η κατάσταση ωστόσο δεν είναι τόσο απλή, καθώς μερικές τομές μπορεί να συμβούν στις κορυφές των πολυγώνων. Το σχήμα 10 παρουσιάζει κάποιες από τις καταστάσεις που πρέπει να προβλεφθούν και έχουν ως εξής:

- i. Ευθέως «εντός»
- ii. Ευθέως «εκτός»
- iii. Η δοκιμαστική γραμμή λήγει επί κορυφής του πολυγώνου αφού έχει αγγίξει άλλες δύο κορυφές
- iv. Η δοκιμαστική γραμμή συμπίπτει με μία πλευρά του πολυγώνου πρωτού λήξει.

Σε κάποιες από τις περιπτώσεις όπου η δοκιμαστική γραμμή τέμνει μια κορυφή θα πρέπει να προσμετράται ως μια τομή με το πολύγωνο, σε άλλες πάλι ως δύο (ή και καμία). Η ανάγκη να χειριστούμε περιπτώσεις όπου κορυφή κείται επί της δοκιμαστικής γραμμής μας αναγκάζει να μην αρκεστούμε στην αρίθμηση τομών. Ο ακόλουθος αλγόριθμος χειρίζεται όλες τις καταστάσεις του σχήματος 10.



ΣΧΗΜΑ 7.10

```

int inside(mypoint t, mypoint *p, int N)
{
    int i, count = 0, j=0;
    myline lt, lp ;
    p[0] = p[N] ; p[N+1] = p[1] ; // sentinel values

    lt.p1 = t; lt.p2 = t; lt.p2.x = 5000000; // test line is
horizontal,
    //traveling towards the infinite positive x
    //the alternative lt.p2.x = 300*INT_MAX wouldn't work

    for(i=1; i<=N; i++)
    {
        lp.p1=p[i] ; lp.p2=p[i]; // degenerate a polygon
segment into a
polygon vertex
        if(!intersect(lp,lt)){
            lp.p2 = p[j] ; j = i;
            if( intersect(lp,lt) ) count ++ ;
        }
    }

    return count & 1 ;
}

```

(παράτιθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

Η συνάρτηση `inside` δέχεται ορίσματα, στις παραμέτρους :

- `mypoint t` → σημείο που ερευνούμε
- `mypoint *p` → δείκτης στον πίνακα `mypoint` που αποθηκεύει τα δεδομένα του πολυγώνου που ερευνούμε. Χρησιμοποιούμε παράμετρο δείκτη η οποία δεικτοδοτείται με αγκύλες `[]` στο σώμα της `inside` και χρησιμοποιείται ως πίνακας. Αυτή είναι η ασφαλέστερη μέθοδος μεταβίβασης πίνακα σε συνάρτηση και η συνηθέστερα χρησιμοποιούμενη .
- `N` → αριθμός κορυφών του πολυγώνου(χρησιμοποιεί στη ροή των βρόγχων `for`).

Ο αλγόριθμος είναι μια σύνθεση των `ccw` και `intersect` που είδαμε προηγουμένως και χρησιμοποιεί οριζόντια δοκιμαστική ευθεία προς απλοποίηση των υπολογισμών. Η τετμημένη της ευθείας είναι 5000000 ώστε το πέρας της να είναι εξασφαλισμένα εκτός του εκάστοτε πολυγώνου (κανένα τοπογραφικό σχέδιο δεν θα έπαιρνε καθολικές συντεταγμένες ως προς σύστημα που απέχει 5000000 m από την περιοχή, ωστόσο θα μπορούσαμε να δώσουμε και μεγαλύτερη τιμή, ανάλογα με τα όρια του τύπου δεδομένων `double` στο περιβάλλον που προγραμματίζουμε). Στην μεταβλητή `j` αποθηκεύουμε την τελευταία κορυφή του πολυγώνου που δεν κείται επί της ευθείας. Αν το επόμενο σημείο του πολυγώνου που δεν είναι επί της δοκιμαστικής ευθείας είναι στην ίδια πλευρά της ευθείας με το `j`-στο σημείο (`j` & `j-1` στο ίδιο ημιεπίπεδο → μη ύπαρξη τομής), τότε δεν αυξάνουμε τον μετρητή τομών (μεταβλητή `count`)· σε διαφορετική περίπτωση έχουμε τομή και ο μετρητής αυξάνεται κατά 1 (`count++`).

Ο αλγόριθμος επιστρέφει αληθή τιμή (1→το σημείο είναι εντός του πολυγώνου) όταν ο μετρητής τομών (`count`) είναι περιττός και ψευδή (0→το σημείο είναι εκτός του πολυγώνου) όταν ο `count` είναι άρτιος. Αυτό είναι ένα συνηθισμένο τέχνασμα διαχωρισμού ζυγών-μονών που γίνεται με την γραμμή `return count & 1;` για να το εξηγήσουμε όμως πρέπει να αναφερθούμε στα ακόλουθα:

- Δυαδικό σύστημα : Η δυαδική αναπαράσταση εκφράζει κάθε αριθμό ως πρόσθεση n δυνάμεων του 2, για n -bito περιβάλλον. Για παράδειγμα σε 8-bito περιβάλλον οι διαθέσιμοι προσθετέοι είναι :

ΔΥΑΔΙΚΟ ΣΥΣΤΗΜΑ	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Αντίστοιχος αριθμός	128	64	32	16	8	4	2	1
Παράδειγμα ο αριθμός 11	0	0	0	0	1	0	1	1
	0*128	0*64	0*32	0*16	1*8	0*4	1*2	1*1

Όταν ο εκάστοτε προσθετέος απαιτείται, η αντίστοιχη θέση παίρνει την τιμή 1 ενώ σε διαφορετική περίπτωση γίνεται 0

- Τελεστής & : αποτελεί τελεστή χειρισμού bit, αντίστοιχο της λογικής πράξης AND του τελεστή δηλαδή &&, μόνο που ο συγκεκριμένος εκτελεί την πράξη σε μεμονωμένα bit. Η πράξη AND επιστρέφει αληθή τιμή όταν και οι δύο τελεστέοι είναι αληθείς και ψευδή τιμή σε κάθε άλλη περίπτωση. Ας θεωρήσουμε για παράδειγμα την πράξη 1&2 σε επίπεδο bit (για 8-bitο περιβάλλον)

$$\begin{array}{r} 0000\ 0001\ (1) \\ \&\ 0000\ 0010\ (2) \\ \hline 0000\ 0000 = 0\ (\Psi\text{ΕΥΔΗΣ}) \end{array}$$

Επειδή λοιπόν στο δυαδικό σύστημα όλοι οι προσθετέοι είναι άρτιοι πλην του τελευταίου ($2^0=1$), το αν ένας δυαδικός αριθμός είναι άρτιος ή όχι καθορίζεται απ' το αν το τελευταίο του ψηφίο είναι 0 (άρτιος) ή 1 (περιττός), επομένως η πράξη «count &1» στα bit ενός αριθμού επιστρέφει 0 (ΨΕΥΔΗΣ) όταν ο count είναι άρτιος και 1 (ΑΛΗΘΗΣ) όταν ο count είναι περιττός :

$$\begin{array}{r} \text{???? ????}\ (ΠΕΡΙΤΤΟΣ) \\ \&\ 0000\ 0001\ (1) \\ \hline 0000\ 0001 = 1\ (ΑΛΗΘΗΣ) \end{array} \qquad \begin{array}{r} \text{???? ????}\ (ΑΡΤΙΟΣ) \\ \&\ 0000\ 0001\ (1) \\ \hline 0000\ 0000 = 0\ (\Psi\text{ΕΥΔΗΣ}) \end{array}$$

Συμπληρωματικά με την συνάρτηση inside, χρησιμοποιείται η ακόλουθη συνάρτηση outline, η οποία αποφαινεται για το αν σημείο ανήκει στο περίγραμμα σχήματος ελέγχοντας

- Αν το σημείο συμπίπτει με κορυφή
- Αν το σημείο ανήκει στις πλευρές, κάνοντας χρήση της τιμής επιστροφής «0» της ccw, για συνευθεία σημεία (βλ. §2)

```
int outline(mypoint t, mypoint *p,int N)
{
    mypoint pa,pb; // variables that call the ccw
    p[0]=p[N];p[N+1]=p[1]; //sentinel values

    for(int i=0; i<=N; i++){
        pa=p[i];pb=p[i+1];
        if( ((pa.x==t.x)&&(pa.y==t.y)) ||
        ((pb.x==t.x)&&(pb.y==t.y)) )
        {
            //"Point is a vertex"
            return 1; }
        if(!ccw(pa,pb,t)) { // "Point on outline"
            return 1;}
    } return 0;}

```

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

Συγκεντρωτικά, σημείο ανήκει σε πολύγωνο όταν έστω μια από τις `inside` και `outline` επιστρέψει αληθή τιμή, κάτι που ελέγχει η συνάρτηση `inPolygon`:

```
bool inPolygon(myPoint t, myPoint *p, int N)
{
    if( inside(t, p, N) || outline(t, p, N) ){return true ;}
    return false ;
}
```

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

7.5 – ΥΠΟΛΟΓΙΣΜΟΣ ΥΨΟΜΕΤΡΩΝ – ΧΕΙΡΙΣΜΟΣ ΤΟΥ ΑΞΟΝΑ Z

Ο άξονας Z, παραλήφθηκε από την παράγραφο των γεωμετρικών μετασχηματισμών, αφενός επειδή η περιστροφή του επιπέδου (X,Y) τον αφήνει αμετάβλητο, αφετέρου επειδή τα σχήματα μελετώνται για όσο το δυνατόν περισσότερο χρόνο στην οριζόντια προβολή τους. Έτσι, το υψόμετρο Z, αντιμετωπίζεται ως βαθμωτό μέγεθος παρότι είναι άρρηκτα συνδεδεμένο με το σύστημα όπου αναφέρεται.

Η πορεία υπολογισμού του επιτρεπτού υψομέτρου κατασκευάζεται έτσι ώστε μόλις αποφανθούμε για την επιφάνεια στην οποία περιλαμβάνεται το προς έλεγχο σημείο, να χρησιμοποιηθεί ο κατάλληλος αλγόριθμος για την εύρεση του Z. Οι μεταβλητές που αποθηκεύουν την κατάσταση περίληψης ή μη είναι τύπου `bool` (`TRUE` ή `FALSE`) και για τον πρώτο διάδρομο ονομάζονται:

Μεταβλητή	Επιφάνεια επί της οποίας ελέγχει την περίληψη
<code>aisINapproach</code>	ΠΡΟΣΕΓΓΙΣΗΣ
<code>aisINTakeoff</code>	ΑΠΟΓΕΙΩΣΗΣ
<code>aisINSafetyzone</code>	ΖΩΝΗ ΑΣΦΑΛΕΙΑΣ
<code>aisINtransitional</code>	ΜΕΤΑΒΑΤΙΚΗ
<code>aisINconical</code>	ΚΩΝΙΚΗ
<code>aisINinner</code>	ΕΣΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ
<code>aisINouter</code>	ΕΞΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ

Η αντίστοιχη ονοματολογία για τον δεύτερο διάδρομο, έχει το πρόθεμα `b`

Ουσιαστικά, ο αλγόριθμος εύρεσης υψομέτρου χρησιμοποιεί τον ορισμό και τα χαρακτηριστικά των επιφανειών για τον υπολογισμό των επιτρεπτών υψομέτρων. Κάθε προσωρινός υπολογισμός αποθηκεύεται στην μεταβλητή `tempHeight` και με λογικές πράξεις (`if`) εξασφαλίζεται πως παίρνουμε το δυσμενέστερο υψόμετρο

```

void acomputeHeight(mypoint &H)
{
    double distance = (H.x*H.x)+(H.y*H.y) ;
    distance = sqrt(distance) ;
    double tempHeight, s=0.05 ;
    if(aisINsafetyzone){H.z = 0; H.description = "SAFETY
ZONE";
        return ;}
    if(aisINinner){H.z = 45 ; H.description="INNER
HORIZONTAL" ;}
    if(aisINouter){
        tempHeight = 45 + s*(-aRin+aRout);
        if(tempHeight<H.z){H.z=tempHeight;
H.description="OUTER
HORIZONTAL";}
    }
    if(aisINconical){tempHeight = 45 + s*(distance-aRin);
        if(tempHeight<H.z){H.z=tempHeight;
H.description="CONICAL";}
    }
    if(aisINapproach){double tempX = -H.x ;
        if(tempX>=(aAP_Xthreshold+aAP_l1+aAP_l2)){tempHeight=(aAP
_l1*
aAP_s1)+(aAP_s2*aAP_l2);}
        else
if(tempX>=(aAP_Xthreshold+aAP_l1)){tempHeight=(aAP_l1*aAP_s1)+
(aAP_s2*(tempX-aAP_l1-aAP_Xthreshold));}
        else tempHeight = aAP_s1*(tempX-aAP_Xthreshold);
        if(tempHeight<H.z){H.z=tempHeight;
H.description="APPROACH";}
    }
    if(aisINtakeoff){
        tempHeight = (H.x-aTO_X)*aTO_s;
        if(tempHeight<H.z){H.z=tempHeight;
H.description="TAKE OFF";}
    }
    if(aisINtransitional){
        double tempY = H.y ;
        if(H.y<0)tempY = -H.y ;
        tempHeight = (tempY - aSezorY)*aTR_s;
        if(tempHeight<H.z){H.z=tempHeight;
H.description="TRANSITIONAL";}
    }
}

```

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

7.6 – ΒΙΒΛΙΟΘΗΚΕΣ ΓΕΩΜΕΤΡΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Μια διαφορετική (και πιο σύγχρονη) προσέγγιση στον γεωμετρικό προγραμματισμό είναι οι βιβλιοθήκες γεωμετρικών αλγορίθμων. Με τον όρο αυτό εννοούμε μια συγκέντρωση συναρτήσεων και αντίστοιχων δεδομένων που υποβοηθούμενα από κατάλληλη τεκμηρίωση (documentation) παρέχουν στον προγραμματιστή τα απαραίτητα εργαλεία και τον αντίστοιχο «τρόπο εργασίας» για την επίλυση γεωμετρικών προβλημάτων. Υπάρχουν διάφορες βιβλιοθήκες που αναπτύσσονται από ανεξάρτητους μεταξύ τους ερευνητικούς ή εμπορικούς φορείς ωστόσο κατά το γενικό πρότυπο των βιβλιοθηκών της C++, διέπονται (σε διαφορετικό βαθμό η καθεμιά) από τα ακόλουθα χαρακτηριστικά:

- i. Διατίθενται σε διαφορετικές μορφές ανάλογα το αν η χρήση τους γίνεται για εμπορικούς, ερευνητικούς ή ατομικούς σκοπούς (διαφορετικές άδειες χρήσης - licenses)
- ii. Είναι λειτουργικές σε διαφορετικά προγραμματιστικά περιβάλλοντα ανά γλώσσα (πχ Visual C++, Borland Turbo C++, Dev C++) ωστόσο απαιτούν διαφορετικές ρυθμίσεις (ή και εκδόσεις) για να τρέξουν στο καθένα.
- iii. Συνεργάζονται με άλλες βιβλιοθήκες (πχ βιβλιοθήκες γραφικών) και παράγουν λειτουργικά αποτελέσματα ακόμα γρηγορότερα.
- iv. Περιέχουν δικούς τους τύπους δεδομένων για την αποθήκευση και χρήση πληροφοριών (όπως ορίσαμε και εμείς για την ανάπτυξη «γηγενών» αλγορίθμων) οι οποίοι είναι φτιαγμένοι έτσι ώστε να προσδίδουν μεγάλη φορητότητα (λειτουργικότητα σε διαφορετικά συστήματα) στον κώδικα που αναπτύσσεται βάση αυτών.
- v. Παρουσιάζουν μεγάλη οργάνωση και ταξινόμηση των αλγορίθμων η οποία γίνεται ανάλογα με την αντίστοιχη επιστημονική περιοχή (πχ βασική γεωμετρία, κυρτά περιβλήματα, διαγράμματα Voronoi, πολύεδρα κοκ) με σκοπό την μετρίαση της δυσκολίας χειρισμού ενός μεγάλου όγκου προγραμματιστικών πόρων.
- vi. Πολλές από αυτές όχι μόνο δέχονται αλλά και παροτρύνουν την συνεισφορά στον κώδικα από τους χρήστες (ανοιχτό λογισμικό – open source). Έτσι γίνονται ταχέως αναπτυσσόμενες και ελέγχονται διαρκώς για λάθη και παραλείψεις.

Η ενσωμάτωση τέτοιων βιβλιοθηκών απαιτεί πολύ καλή γνώση της γλώσσας προγραμματισμού και η χρήση τους αποδεικνύεται κάθε άλλο παρά απλή. Ενδεικτικά αναφέρουμε πως το εγχειρίδιο χρήσης της βιβλιοθήκης CGAL (Computational Geometry Algorithm) έχει έκταση 3500 σελίδων, προς υπεράσπισή της όμως παραθέτουμε κάποια από τα ισχυρά χαρακτηριστικά της:

- 600,000 γραμμές κώδικα

- 90 συστατικά λογισμικού (software components)
- Υποστήριξη διαφόρων συστημάτων (Cross platform support)
- Ετήσιες εκδόσεις με 12000 κατά μέσο όρο λήψεις (downloads)
- Η CGAL χρησιμοποιείται σε πολλά πεδία εφαρμογών από εταιρίες όπως Total (Oil&gas), British Telecom (Telecom), Cadence (VLSI), Leica Geosystems (GIS), Dassault Systèmes (CAD), The Moving Picture Company (Visual effects).
- Κάλυψη του συνόλου σχεδόν της υπολογιστικής γεωμετρίας (περικλείων όγκος, πολυεδρικές επιφάνειες, λογικές πράξεις μεταξύ σχημάτων, τριγωνισμοί, διαγράμματα Voronoi, δημιουργία πλέγματος(mesh), υποδιαιρέσεις, παραμετροποιήσεις και απλοποίηση σχημάτων, έρευνα γειτονίας, διατάξεις, τομές, άθροισμα Minkowski, περιβλήματα κα)

Παρόμοιες βιβλιοθήκες είναι οι GEOS και Wykobi, ευκολότερες στη χρήση αλλά υποδεέστερες της CGAL ενώ γεωμετρικά κατηγορήματα (predicates) παρέχονται και σε μη εξειδικευμένες βιβλιοθήκες. Τέτοιες βιβλιοθήκες αυξάνουν την παραγωγικότητα καθώς μας επιτρέπουν να επικεντρωθούμε στην εφαρμογή που αναπτύσσουμε. Για τους σκοπούς της εργασίας ωστόσο, δεν κρίθηκε σκόπιμη η χρήση τους αφενός γιατί θα απέτρεπαν από την κατανόηση της γεωμετρικής (μαθηματικής) φύσης του προβλήματος που ερευνούμε (προσδιορισμός επιτρεπτών υψομέτρων) και των λύσεων που επιδέχεται, αφετέρου επειδή θα απαιτούσαν εμβάθυνση στον προγραμματισμό η οποία ξεφεύγει από την «δικαιοδοσία» του προπτυχιακού πολιτικού μηχανικού. Σε κάθε περίπτωση η επέκταση του αναπτυσσόμενου προγράμματος θα μπορούσε να περιλαμβάνει την χρήση τέτοιων βιβλιοθηκών αφού η λύση σε επιπλέον πεδία προβλημάτων μπορεί να επιδέχεται ανεξάρτητο προγραμματισμό· άλλωστε η θεωρητική βάση τους είναι η υπολογιστική γεωμετρία, βάση της οποίας αναπτύχθηκαν και οι γηγενείς αλγόριθμοι στο παρόν κεφάλαιο, επομένως δεν θα υπάρχουν συγκρούσεις «λογικής» με τον υφιστάμενο κώδικα.

ΚΕΦΑΛΑΙΟ 8

ΟΙ ΕΠΙΦΑΝΕΙΕΣ ΠΕΡΙΟΡΙΣΜΟΥ ΕΜΠΟΔΙΩΝ ΣΤΟ ΠΡΟΓΡΑΜΜΑ

Το παρόν κεφάλαιο περιγράφει την παραμετροποίηση των επιφανειών, ουσιαστικά δηλαδή τον τρόπο με τον οποίο «αντιλαμβάνεται» τις επιφάνειες περιορισμού εμποδίων το πρόγραμμα και δίνει μια έννοια των αλγορίθμων που τρέχουν μόλις ο χρήστης πατήσει το κουμπί “Generate Surfaces” στο γραφικό περιβάλλον αλληλεπίδρασης (GUI) του προγράμματος.

Κατά τη δημιουργία του κώδικα του προγράμματος γίνεται προσπάθεια να υλοποιηθεί μια απεικόνιση των γεωμετρικών στοιχείων αεροδρομίου με τη χρήση όσο το δυνατόν λιγότερων μεταβλητών. Ο λόγος που συμβαίνει κάτι τέτοιο είναι η μείωση της έκτασης του κώδικα, κάτι που τον καθιστά ευκολότερα επισκέψιμο και κάνει τις αιτίες πιθανών λαθών κατά το δυνατόν ορατές. Έτσι η κάθε επιφάνεια που μελετάται «σπάει» στα δομικά της στοιχεία, δηλαδή στις αναγκαστικά ανεξάρτητες μεταβλητές που ορίζονται από τον ICAO (και τώρα γίνονται μεταβλητές του προγράμματος) και στις παράγωγες μεταβλητές που μπορούν να οριστούν από γεωμετρικές σχέσεις και συναρτήσεις μεταξύ των πρώτων μεταβλητών.

Οι συντεταγμένες των κορυφών των επιφανειών περιορισμού εμποδίων δεν αποθηκεύονται σε βάση δεδομένων ώστε να γίνει ανάθεση ανάλογα με την επιλογή τύπου αεροδρομίου, καθώς η μεταβλητότητα του μήκους του διαδρόμου (εισάγεται ως στοιχείο στο Graphical User Interface του προγράμματος) θα απαιτούσε αναπροσαρμογή αυτών. Αντ’ αυτού μπορούμε να ορίσουμε κάποια σημεία αναφοράς επί του διαδρόμου και έπειτα με χρήση γεωμετρικών σχέσεων και των κατάλληλων μεταβλητών να παράγουμε κάθε φορά το επιθυμητό αποτέλεσμα. Ο τρόπος με τον οποίο ανατίθενται τιμές στις συντεταγμένες των κορυφών είναι προσθετικός, δηλαδή προχωράμε από μέσα (άξονας διαδρόμου) προς τα έξω, ορίζοντας κάθε φορά νέα σημεία αναφοράς ώστε να άρουμε την μεταβλητότητα ενδιάμεσων στοιχείων και να χρησιμοποιούμε δεδομένα σημεία (και δεδομένες αποστάσεις ως προς αυτά) ως άγκυρα (anchor) για τον υπολογισμό των υπολοίπων. Η ιδέα της άγκυρας, ως μεθόδου υπολογισμού γεωμετρικών οντοτήτων, είναι ισχυρό εργαλείο της υπολογιστικής γεωμετρίας και βρίσκει συχνή εφαρμογή στους αλγορίθμους εύρεσης κυρτού περιβλήματος σε 2 και τρεις διαστάσεις.

Για τους σκοπούς της διπλωματικής εργασίας δεν μελετάται το σύνολο των επιφανειών περιορισμού εμποδίων αλλά οι κυριότερες εξ αυτών καθώς κύριο μέλημα είναι να οριστούν οι μέθοδοι και τα γνωστικά πεδία (μαθηματικά εργαλεία) που μας επιτρέπουν να καταπιαστούμε με γεωμετρικά (στη βάση τους) προβλήματα σε προγραμματιστικά περιβάλλοντα. Οποιοδήποτε επέκταση επί των δευτερευόντων επιφανειών θα αποτελούσε εργασία επαναληπτική και χρονοβόρα που θα αποσπούσε από την εμβάθυνση και κατανόηση του τρόπου εργασίας σε παρόμοια θέματα.

Στο πρόγραμμα ορίζονται και μελετώνται τα παρακάτω στοιχεία:

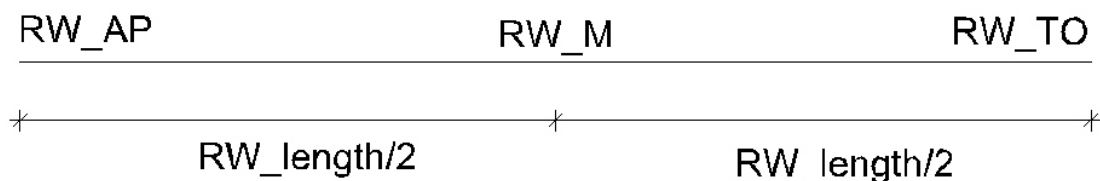
- * Άξονας διαδρόμου
- 1. Ζώνη ασφαλείας
- 2. Εσωτερική οριζόντια επιφάνεια
- 3. Κωνική επιφάνεια
- 4. Εξωτερική οριζόντια επιφάνεια
- 5. Επιφάνεια Προσέγγισης
- 6. Επιφάνεια Απογείωσης
- 7. Μεταβατική επιφάνεια

• ΑΞΟΝΑΣ ΔΙΑΔΡΟΜΟΥ

Ο άξονας του διαδρόμου δεν συγκαταλέγεται στις επιφάνειες απομάκρυνσης εμποδίων, εισαγάγουμε ωστόσο μια προγραμματιστική (ως ακολούθως περιγραφόμενη) έκφραση αυτού, επειδή το μήκος του (RW_length) δεν είναι δεδομένο (εισάγεται ως μεταβλητή στο GUI του προγράμματος).

Η προγραμματιστική έκφραση του άξονα περιλαμβάνει τα άκρα (RW_TO & RW_AP) και το μέσο του (RW_M). Οι τοπικές συντεταγμένες των επιφανειών απομάκρυνσης εμποδίων υπολογίζονται με σημεία αναφοράς τα παραπάνω, τα οποία στο τοπικό σύστημα έχουν μηδενική τεταγμένη και τετμημένη ως εξής :

$$x = x_0 \pm \frac{l}{2}, \text{ για } \begin{aligned} x &= RW_AP, RW_TO \\ l &= \text{Μήκος}(RW_length) \text{ διαδρόμου} \\ x_0 &= RW_M \end{aligned}$$

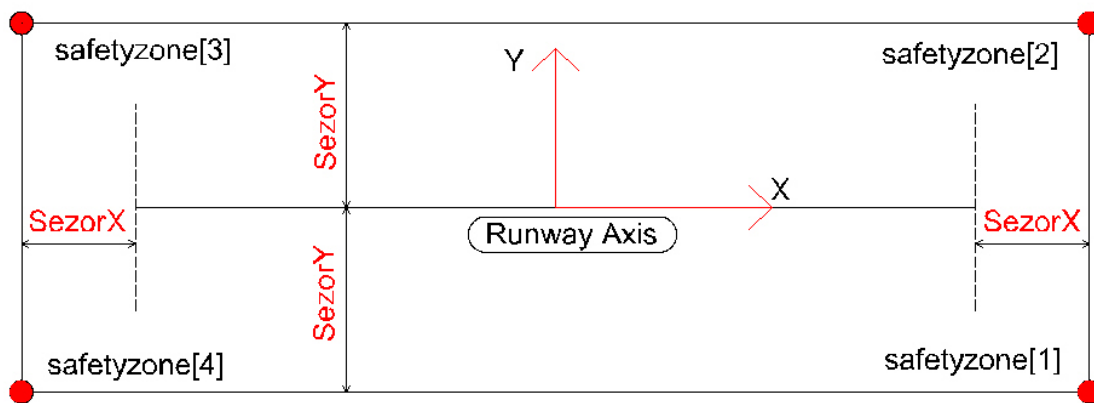


ΣΧΗΜΑ 8.1

8.1 – ΖΩΝΗ ΑΣΦΑΛΕΙΑΣ

Η ζώνη ασφαλείας αποτελεί (σε οριζόντια προβολή) ένα τετράπλευρο τα δεδομένα του οποίου θα αποθηκευτούν σε πίνακα struct mypoint (βλ. κεφ. 6.1 «Δόμηση δεδομένων») με όνομα safetyzone[6] (η χρήση πίνακα 6 στοιχείων για την αποθήκευση σχήματος 4 κορυφών εξηγήθηκε στο κεφάλαιο 6).

Ο πίνακας θα οριστεί με τη χρήση των μεταβλητών SezorX, SezorY οι οποίες αντιπροσωπεύουν τις αποστάσεις κατά x & y αντίστοιχα των κορυφών του τετραπλεύρου από τα σημεία αναφοράς επί του άξονα του διαδρόμου, όπως εικονίζεται παρακάτω:



ΣΧΗΜΑ 8.2

Οι τιμές των μεταβλητών SezorX, & SezorY για τα διάφορα είδη προσέγγισης και κωδικούς αριθμούς διαδρόμου δίδονται στο τέλος του κεφαλαίου.

Ο αλγόριθμος που θα ορίζει τις συντεταγμένες του σχήματος στο τοπικό σύστημα κάθε διαδρόμου έχει ως εξής:

```
safetyzone[1].x = (RW_lenght/2) + SezorX ;
safetyzone[2].x = safetyzone[1].x ;
safetyzone[3].x = -safetyzone[2].x ;
safetyzone[4].x = safetyzone[3].x ;
safetyzone[1].y = - SezorY ;
safetyzone[2].y = SezorY ;
safetyzone[3].y = SezorY ;
safetyzone[4].y = -SezorY ;
```

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

Χρησιμοποιούμε τα προθέματα a & b για τους διαδρόμους 1 & 2 αντίστοιχα

8.2 – ΕΣΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ ΕΠΙΦΑΝΕΙΑ

Το κεφάλαιο 4 του Annex 14 (έκδοση 2004) επιτρέπει τη λήψη εσωτερικής οριζόντιας επιφάνειας με ένα (κυκλική) ή δύο («αμφιθεατρική») σημεία αναφοράς. Υιοθετείται η πρώτη προσέγγιση για τους σκοπούς της εργασίας.

Για τη «δημιουργία» της επιφάνειας στο πρόγραμμα δεν θα ορίσουμε νέο αντικείμενο (πχ τύπου “circle”) αλλά μια νέα μεταβλητή, την R_{in} , που αντιπροσωπεύει την ακτίνα της ενώ κάνουμε παράλληλα χρήση του σημείου RW_M ως σημείο αναφοράς (κέντρο). Η μορφή της επιφάνειας φαίνεται στο ΣΧΗΜΑ 4 ενώ οι τιμές της μεταβλητής R_{in} για τα διάφορα είδη προσέγγισης και κωδικούς αριθμούς αεροδρομίου δίδονται στο τέλος του κεφαλαίου.

8.3 – ΚΩΝΙΚΗ ΕΠΙΦΑΝΕΙΑ

Όμοια με προηγούμενα η κωνική επιφάνεια έχει κυκλικά όρια (παίρνει μορφή δακτυλίου). Εσωτερικό όριο της κωνικής επιφάνειας είναι ο κύκλος $C(RW_M, R_{in})$ ενώ εξωτερικό ο $C'(RW_M, R_{out})$.

Η ακτίνα R_{out} ορίζεται ως

$$R_{out} = R_{in} + DR$$

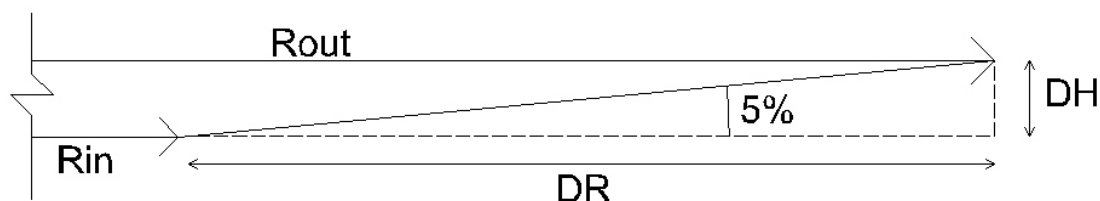
όπου DR η (ακτινική) απόσταση μέχρι επίτευξης υψομετρικής διαφοράς (DH) ίσης με

- 100 μέτρα για κατηγορίες 3 & 4
- 60 μέτρα για κατηγορίες 1 & 2

για κλίση 5%, δηλαδή

$$DR = DH / 5\% \Rightarrow \Delta R =$$

- 2000 μέτρα για κατηγορίες 3 & 4
- 1200 μέτρα για κατηγορίες 1 & 2

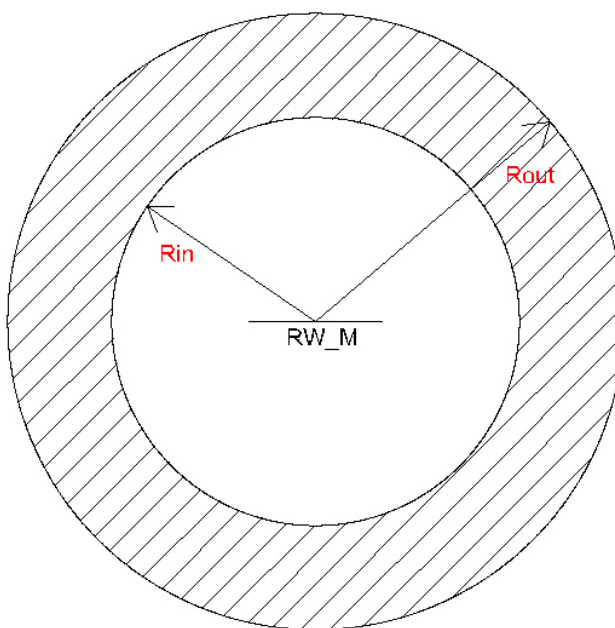


ΣΧΗΜΑ 8.3

Η μορφή της επιφάνειας φαίνεται (σκιαγραφημένη), σε οριζόντια προβολή, στο ΣΧΗΜΑ 4 ενώ οι τιμές της μεταβλητής R_{out} για τα διάφορα είδη προσέγγισης και κωδικούς αριθμούς αεροδρομίου δίδονται στο τέλος του κεφαλαίου.

8.4 – ΕΞΩΤΕΡΙΚΗ ΟΡΙΖΟΝΤΙΑ ΕΠΙΦΑΝΕΙΑ

Ορίζεται εκτός του κύκλου $C'(RW_M, R_{out})$ και δεν απαιτεί την εισαγωγή νέας μεταβλητής για την περιγραφή της (βλ. ΣΧΗΜΑ 4)



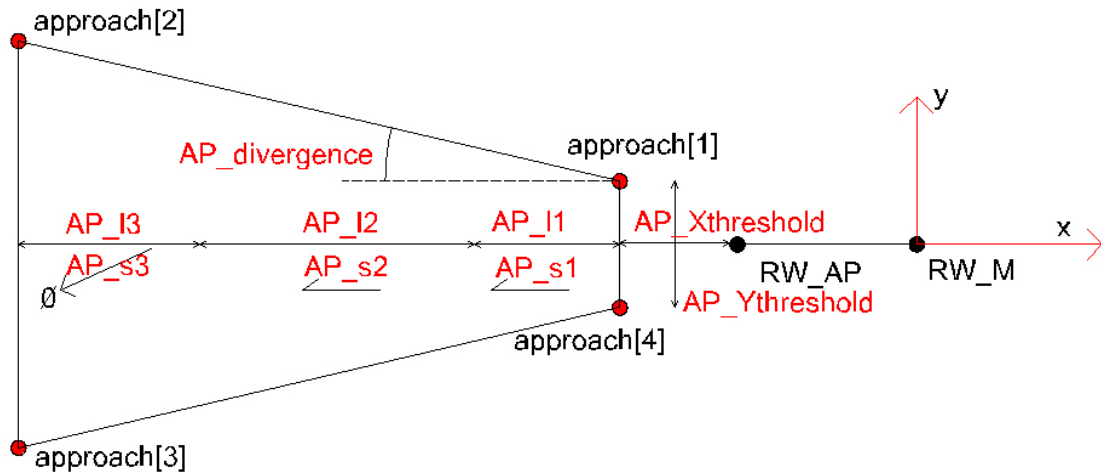
ΣΧΗΜΑ 8.4

8.5 – ΕΠΙΦΑΝΕΙΑ ΠΡΟΣΕΓΓΙΣΗΣ

Η επιφάνεια προσέγγισης αποτελεί (σε οριζόντια προβολή) ένα τετράπλευρο τα δεδομένα του οποίου θα αποθηκευτούν σε πίνακα struct mypoint με όνομα approach[6] (βλ. κεφ. 6 «ΑΠΟΘΗΚΕΥΣΗ & ΧΡΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ» παρ.1 «ΔΟΜΗΣΗ ΔΕΔΟΜΕΝΩΝ»).

Ο πίνακας θα οριστεί με τη χρήση των ακόλουθων μεταβλητών (βλ. ΣΧΗΜΑ 5) :

- AP_l1, AP_l2, AP_l3, AP_Xthreshold, AP_Ythreshold → εικονιζόμενα μήκη
- AP_s1, AP_s2, AP_s3 → κλίσεις (επί κατακόρυφου επιπέδου)
- AP_divergence → εικονιζόμενη (οριζοντιογραφική) κλίση



ΣΧΗΜΑ 8.5

Οι τιμές των παραπάνω μεταβλητών για τα διάφορα είδη προσέγγισης και κωδικούς αριθμούς αεροδρομίων δίδονται στο τέλος του κεφαλαίου. Ο αλγόριθμος που θα ορίζει τις συντεταγμένες του σχήματος στο τοπικό σύστημα κάθε διαδρόμου έχει ως εξής:

```
aapproach[1].x = aRW_AP.x - aAP_Xthreshold ;
//έχει ήδη οριστεί RW_AP.x = -(RW_length)/2 & RW_AP.y = 0
aapproach[1].y = aAP_Ythreshold/2 ;
aapproach[2].x = aapproach[1].x - (aAP_l1 + aAP_l2 +
aAP_l3) ;
aapproach[2].y = aapproach[1].y +
((aAP_l1 + aAP_l2 + aAP_l3) * aAP_divergence) ;
aapproach[3].x = aapproach[2].x ;
aapproach[3].y = -aapproach[2].y ;
aapproach[4].x = aapproach[1].x ;
aapproach[4].y = -aapproach[1].y ;
```

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

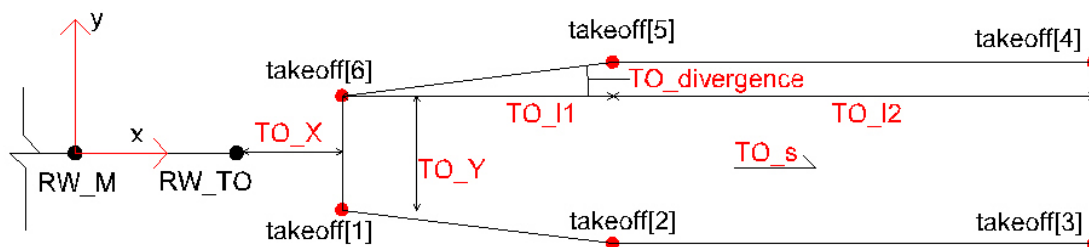
→Θυμίζουμε ότι το *a* δηλώνει ότι ο κώδικας αναφέρεται στον πρώτο διάδρομο (RUNWAY 1). Ο αντίστοιχος κώδικας για τον 2^ο διάδρομο έχει πρόθεμα *b*

8.6 – ΕΠΙΦΑΝΕΙΑ ΑΠΟΓΕΙΩΣΗΣ

Η επιφάνεια απογείωσης αποτελεί (σε κάτοψη) ένα εξάπλευρο τα δεδομένα του οποίου θα αποθηκευτούν σε πίνακα struct mypoint (βλ. κεφ. 6.1) με όνομα takeoff[8] .

Ο πίνακας θα οριστεί με τη χρήση των ακόλουθων μεταβλητών (βλ. ΣΧΗΜΑ 6) :

- TO_X, TO_Y, TO_I1, TO_I2 → εικονιζόμενα μήκη
- TO_divergence → εικονιζόμενη (οριζοντιογραφική) κλίση
- TO_s → κλίση (επί κατακόρυφου επιπέδου)



ΣΧΗΜΑ 8.6

Οι τιμές των παραπάνω μεταβλητών για τους διάφορους κωδικούς αριθμούς αεροδρομίων δίδονται στο τέλος του κεφαλαίου .

Στην παρακάτω φόρμα φαίνεται η πορεία υπολογισμού των μη δοσμένων (παράγωγων) μεταβλητών TO_I1 & TO_I2

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Κατηγορία Αεροδρομίου	TO_Y	Τελικό Εύρος	TO_divergence (%)	$\text{TO_I1} = \frac{(3)-(2)}{2*(4)}$	Μήκος	$\text{TO_I2} = (6)-(5)$
1	60	380	10	1600	1600	0
2	80	580	10	2500	2500	0
3,4	180	1200	12,5	4080	15000	10920

Ο αλγόριθμος που θα ορίζει τις συντεταγμένες του σχήματος στο τοπικό σύστημα κάθε διαδρόμου έχει ως εξής:

```

atkeoff[1].x = aRW_TO.x + aTO_X ;
//έχει ήδη οριστεί RW_TO.x = (RW_length)/2 & RW_AP.y = 0
atkeoff[1].y = -(aTO_Y/2);
atkeoff[2].x = atkeoff[1].x + aTO_l1 ;
atkeoff[2].y = atkeoff[1].y -
                ((aTO_l1)*(aTO_divergence));
atkeoff[3].x = atkeoff[2].x + aTO_l2 ;
atkeoff[3].y = atkeoff[2].y ;
atkeoff[4].x = atkeoff[3].x ;
atkeoff[4].y = -atkeoff[3].y ;
atkeoff[5].x = atkeoff[2].x ;
atkeoff[5].y = -atkeoff[2].y ;
atkeoff[6].x = atkeoff[1].x ;
atkeoff[6].y = -atkeoff[1].y ;

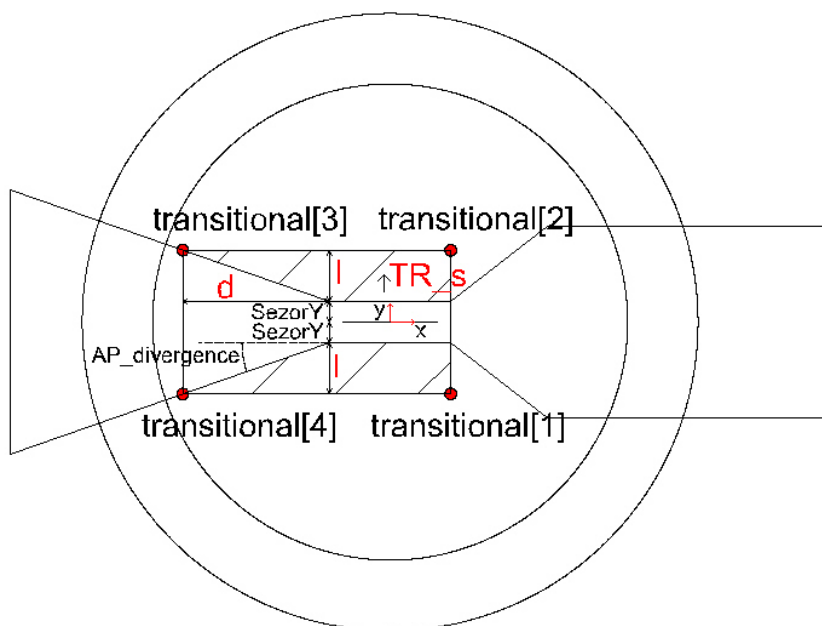
```

(παράτίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

→ Θυμίζουμε ότι το *a* δηλώνει ότι ο κώδικας αναφέρεται στον πρώτο διάδρομο (RUNWAY 1). Ο αντίστοιχος κώδικας για τον 2^ο διάδρομο έχει πρόθεμα *b*

8.7 – ΜΕΤΑΒΑΤΙΚΗ ΕΠΙΦΑΝΕΙΑ

Ορίζουμε το εικονιζόμενο τετράπλευρο για λόγους που θα εξηγήσουμε παρακάτω



ΣΧΗΜΑ 8.7

Η μεταβατική επιφάνεια (σε κάτοψη) αποτελείται από τα 2 σκιαγραφημένα τετράπλευρα. Το περικλείον τετράπλευρο που ορίζεται από τις κορυφές transitional[1] έως transitional[4] (φορά διαγραφής ΑΔΩ) ορίζεται με τη βοήθεια των ακολούθων μεταβλητών :

- l : το κατά y μήκος που απαιτείται ώστε να επιτευχθεί υψομετρική διαφορά 45 μέτρων (από την ζώνη ασφαλείας ως την εσωτερική οριζόντια επιφάνεια) με κλίση TR_s

$$l = \frac{45}{TR_s}, \text{ βλ. ΣΧΗΜΑ 7}$$

- d : το κατά x μήκος πέραν της ζώνης ασφαλείας που απαιτείται ώστε οι οριζόντιες πλευρές της μεταβατικής επιφάνειας να τμήσουν την επιφάνεια προσέγγισης

$$d = \frac{45}{TR_s * AP_divergence}, \text{ βλ. ΣΧΗΜΑ 7}$$

Καθώς l & d είναι μονοσήμαντα ορισμένα ως προς TR_s & $AP_divergence$, η μόνη μεταβλητή που προστίθεται στο πρόγραμμα εξαιτίας της επιφάνειας είναι η TR_s , οι τιμές της οποίας δίδονται στο τέλος του κεφαλαίου για τους διάφορους κωδικούς αριθμούς αεροδρομίων.

Ο λόγος που δεν ορίζουμε ξεχωριστά τα σκιαγραφημένα τετράπλευρα είναι η οικονομία υπολογισμών. Αντί να προσθέσουμε 2 ξεχωριστά σχήματα στο πρόγραμμα (8 κορυφές) που θα απαιτούσαν να τρέχουν εκ νέου οι αλγόριθμοι περίκλεισης και υπολογισμού ύψους προτιμάται η ακόλουθη προσέγγιση: Ορίζεται (με λογικές πράξεις (if) μεταξύ των συναρτήσεων περίκλεισης) πως ένα σημείο ανήκει στη μεταβατική επιφάνεια όταν και μόνο όταν ανήκει στο περικλείον τετράπλευρο χωρίς να ανήκει σε καμία από τις «επιφάνεια απογείωσης» και «ζώνη ασφαλείας». Επιπλέον αν ανήκει στον κάτω τομέα της μεταβατικής επιφάνειας (αρνητικά τοπικά y) χρησιμοποιείται ο ίδιος αλγόριθμος υπολογισμού ύψους με τον πάνω τομέα, μόνο που λαμβάνεται το $|y|$ ως τεταγμένη στους υπολογισμούς. Παρότι η οικονομία υπολογισμών που επιτυγχάνεται δεν είναι εμφανής κατά τον χρόνο εκτέλεσης του προγράμματος (εξαιτίας της μικρής έκτασης του προγράμματος συγκριτικά με τη μέση σύγχρονη υπολογιστική ισχύ), λαμβάνουμε την παραπάνω μέριμνα σε αυτό και πολλά άλλα βήματα του προγραμματισμού, προκαταλαμβάνοντας αντίστοιχες ενέργειες που θα ήταν απαραίτητες σε μελλοντικές επεκτάσεις του προγράμματος ή σε περίπτωση ενσωμάτωσής του σε μεγαλύτερα πακέτα λογισμικού (η συγκεκριμένη «μηχανή» μπορεί να χρησιμοποιηθεί πολύ εύκολα ανεξάρτητα από το γραφικό της περιβάλλον αφού οι αλγόριθμοι και οι μεταβλητές ορίζονται εκτός της κύριας εκ του CLR παραγόμενης τάξης Form1.h – βλ. κεφ. 9 «ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ H/Y»)

Ο αλγόριθμος που θα καθορίζει τις συντεταγμένες του σχήματος στο τοπικό σύστημα κάθε διαδρόμου έχει ως εξής:

```

atransitional[1].x = asafetyzone[1].x ;
atransitional[1].y = -(aSezorY + (45/aTR_s)) ;
atransitional[2].x = atransitional[1].x ;
atransitional[2].y = -atransitional[1].y ;
atransitional[3].x = asafetyzone[3].x -
                    (45/(aTR_s*aAP_divergence)) ;
atransitional[3].y = atransitional[2].y ;
atransitional[4].x = atransitional[3].x ;
atransitional[4].y = atransitional[1].y ;

```

(παρατίθεται στο «Παράρτημα Α – Κώδικας προγράμματος»)

→ Θυμίζουμε ότι το a δηλώνει ότι ο κώδικας αναφέρεται στον πρώτο διάδρομο (RUNWAY 1). Ο αντίστοιχος κώδικας για τον 2^ο διάδρομο έχει πρόθεμα b

ΕΠΙΦΑΝΕΙΑ			ΠΡΟΣΕΓΓΙΣΗΣ									ΚΩΝΙΚΗ		ΑΠΟΓΕΙΩΣΗΣ							ΖΩΝΗ ΑΣΦΑΛΕΙΑΣ		ΜΕΤΑΒΑΤΙΚΗ
ΕΙΔΟΣ ΠΡΟΣΕΓΓΙΣΗΣ	ΚΩΔΙΚΟΣ	PortCode	AP_Xthreshold	AP_Ythreshold	AP_I1	AP_s1 (%)	AP_I2	AP_s2 (%)	AP_I3	AP_s3 (%)	AP_divergence (%)	Rin	Rout	TO_X	TO_Y	TO_I1	TO_I2	TO_s (%)	TO_divergence (%)	SezorX	SezorY	TR_s (%)	
ΜΗ ΕΝΟΡΓΑΝΗ	1	101	30	60	1600	5	∅	∅	∅	∅	10	2000	2700	30	60	1600	0	5	10	30	30	20	
	2	102	60	80	2500	4					10	2500	3600	60	80	2500	0	4	10	60	40	20	
	3	103	60	150	3000	3,33					10	4000	5500	60	180	4080	10920	2	12,5	60	75	14,3	
	4	104	60	150	3000	2,5					10	4000	6000	60	180	4080	10920	2	12,5	60	75	14,3	
ΜΗ ΑΚΡΙΒΗΣ (ΕΝΟΡΓΑΝΗ)	1	201	60	150	2500	3,3	0	0	0	∅	15	3500	4700	30	60	1600	0	5	10	60	75	20	
	2	202	60	150	2500	3,3	0	0	0		15	3500	4700	60	80	2500	0	4	10	60	75	20	
	3	203	60	300	3000	2	3600	2,5	8400		15	4000	5500	60	180	4080	10920	2	12,5	60	150	14,3	
	4	204	60	300	3000	2	3600	2,5	8400		15	4000	6000	60	180	4080	10920	2	12,5	60	150	14,3	
ΑΚΡΙΒΗΣ (ΕΝΟΡΓΑΝΗ)	1	301	60	150	3000	2,5	12000	3	0	∅	15	3500	4700	30	60	1600	0	5	10	60	75	20	
	2	302	60	150	3000	2,5	12000	3	0		15	3500	4700	60	80	2500	0	4	10	60	75	20	
	3	303	60	300	3000	2,5	3600	2,5	8400		15	4000	6000	60	180	4080	10920	2	12,5	60	150	14,3	
	4	304	60	300	3000	2,5	3600	2,5	8400		15	4000	6000	60	180	4080	10920	2	12,5	60	150	14,3	

ΠΙΝΑΚΑΣ 8.1

ΚΕΦΑΛΑΙΟ 9

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ Η/Υ

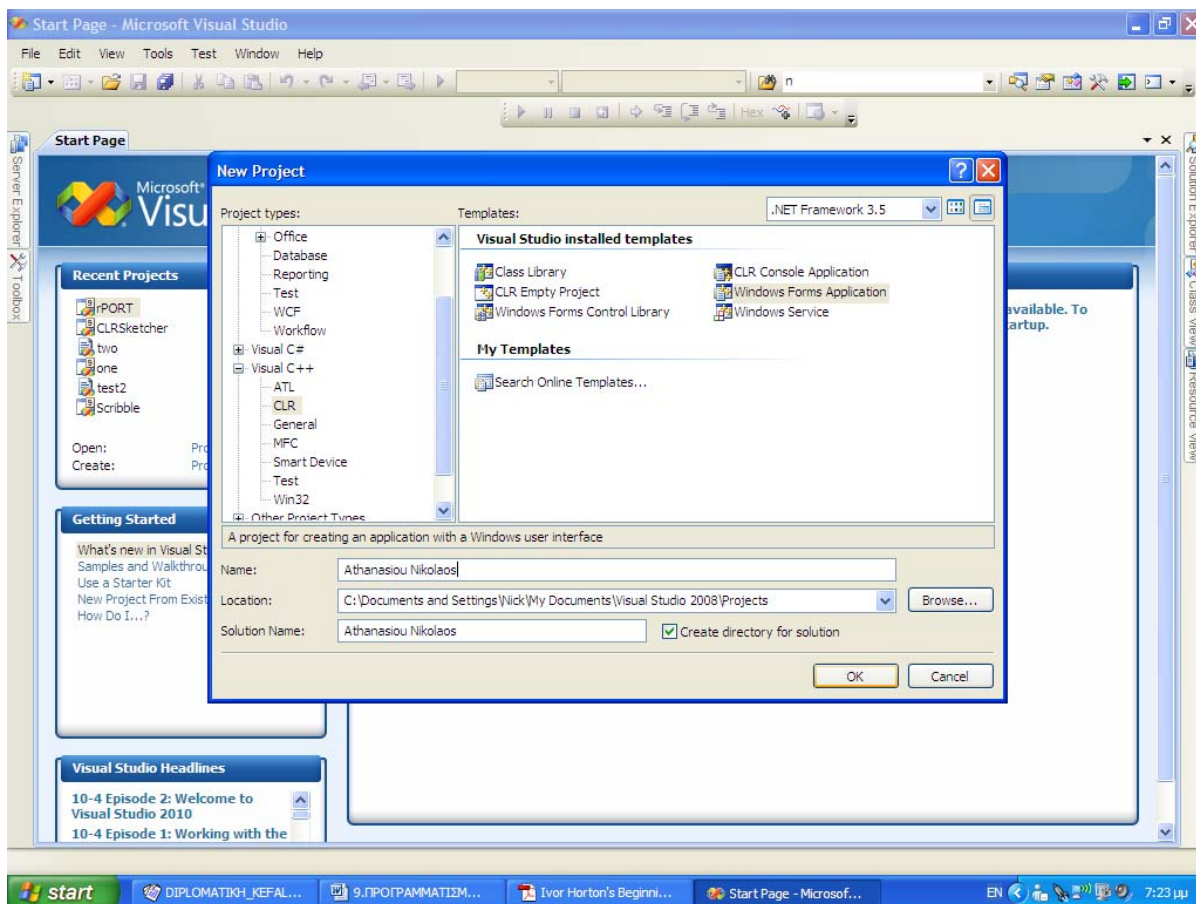
Στο παρόν κεφάλαιο γίνεται μια σύντομη παρουσίαση του προγραμματιστικού περιβάλλοντος της Visual C++ 2008 και ακολουθεί μια βήμα προς βήμα εκτέλεση του προγράμματος όπου επεξηγείται ο κώδικας που τρέχει στο «παρασκήνιο» κάθε φορά· τέλος παρατίθεται μια δοκιμαστική εκτέλεση του προγράμματος όπου τυπώνονται τα δεδομένα εισόδου και τα παραγόμενα αποτελέσματα. Οι δύο πρώτες ενότητες θα μπορούσαν να αποτελέσουν ξεχωριστά κεφάλαια, ωστόσο

- κρίθηκε μη σκόπιμη η διεξοδική παρουσίαση ενός προγράμματος του οποίου δεν χρησιμοποιούμε το σύνολο των χαρακτηριστικών και δυνατοτήτων (ενδεικτικά αναφέρουμε πως ένα μέσο εισαγωγικό εγχειρίδιο της Visual C++ έχει έκταση περί τις 1400 σελίδες)
- επιχειρείται να δοθούν απλώς τα λογικά βήματα με τα οποία οι αλγόριθμοι συντίθενται σε ένα λειτουργικό σύνολο

Έτσι ο αναγνώστης αποκτά εποπτεία επί των χαρακτηριστικών του προγραμματιστικού περιβάλλοντος που αφορούν τη δημιουργία του προγράμματος rPORT ενώ έχοντας εξασφαλίσει κατανόηση της λογικής του προγράμματος μπορεί, μετά τη μελέτη του παραρτήματος Α όπου παρατίθεται ο κώδικας, να επέμβει με αλλαγές και βελτιώσεις στην περίπτωση μελλοντικής αναθεώρησής του.

Οι επικρατέστερες εναλλακτικές για τη δημιουργία του rPORT (πάντα με χρήση Visual C++) ήταν το MFC (Microsoft Foundation Classes) και το WFA (Windows Forms Application) και εξαιτίας ευελιξίας και ευκολίας χρήσης προτιμήθηκε η δεύτερη. Παρακάτω περιγράφονται και εξηγούνται αποκλειστικά τα των προγραμμάτων WFA, τα οποία δημιουργούνται στο πλαίσιο διαλόγου που φαίνεται στην εικόνα 9.1 και έχουν αρχική μορφή όπως στην εικόνα 9.2 .

9.1 – ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ VISUAL C++ 2008



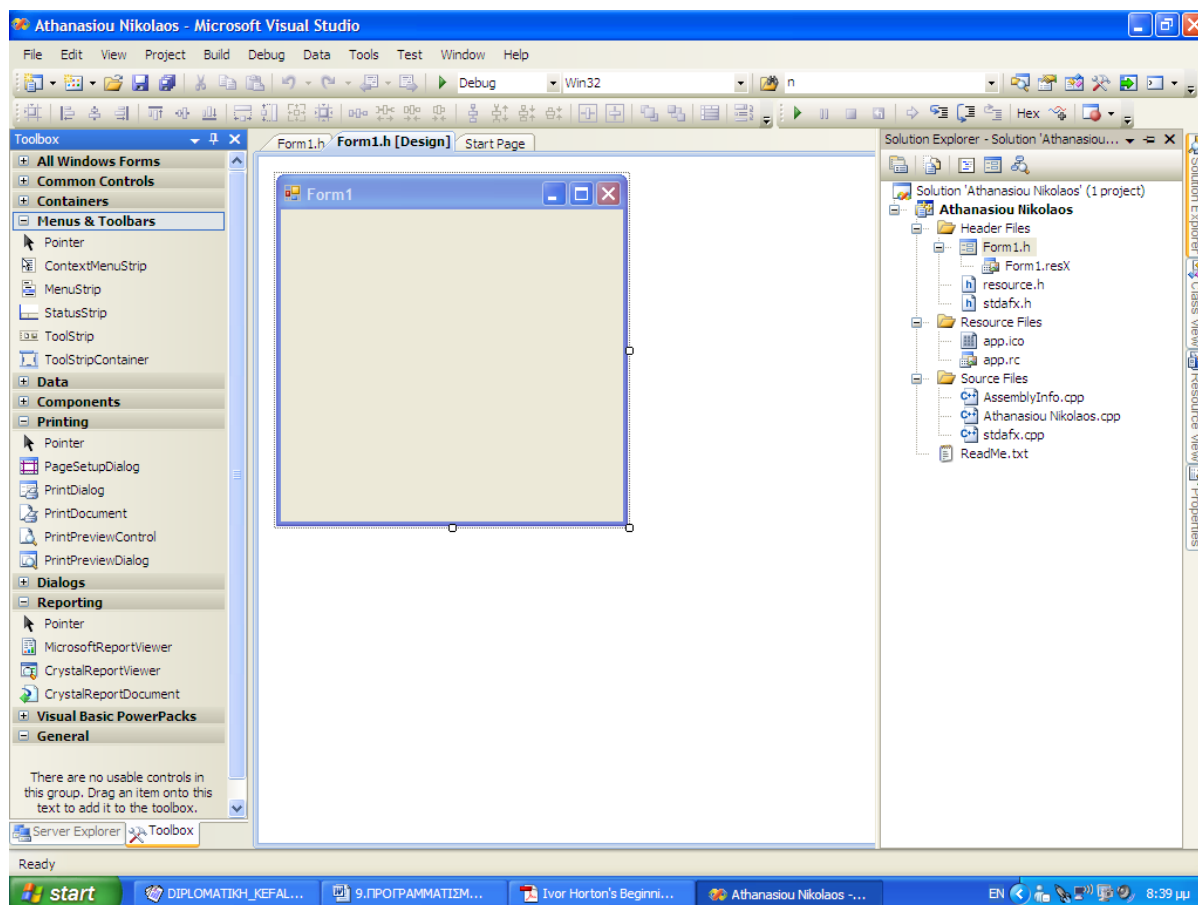
ΕΙΚΟΝΑ 9.1

ΓΕΝΙΚΑ

Το Visual studio παρέχει διάφορες επιλογές ως προς τη γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε. Η έκδοση της C++ που υποστηρίζεται στο Visual Studio καλείται Visual C++ (αντίστοιχα έχουμε Visual Basic και Visual C#) και παρότι παρουσιάζει αρκετές διαφορές από την C++ όπως ορίζεται στον ISO/ANSI (International Standards Organization /American National Standards Institute) διατηρεί τα δομικά χαρακτηριστικά και την εμφάνιση της αρχικής γλώσσας· μάλιστα είναι δυνατή και ανάπτυξη προγραμμάτων σε γηγενή (native) κώδικα, με την προτυποποιημένη μορφή, ωστόσο με αυτόν τον τρόπο δεν εκμεταλλευόμαστε κανένα από τα εξελιγμένα χαρακτηριστικά και τις λειτουργίες του εν λόγω IDE (Integrated Design Environment – Ενσωματωμένο περιβάλλον σχεδίασης).

Όπως φαίνεται στην εικόνα 9.1 επιλέγουμε να εργαστούμε σε CLR κατασκευάζοντας Windows Forms Application. Το CLR (Common Language

Runtime) είναι ένα προτυποποιημένο (standardized) περιβάλλον για την εκτέλεση προγραμμάτων γραμμένων σε ένα ευρύ φάσμα γλωσσών υψηλού επιπέδου και αποτελεί στοιχείο του .NET framework, του γνωστού «πλαίσιου» εργασίας που κάνει πιο γρήγορη την ανάπτυξη προγραμμάτων για Windows. Ως προς την επιλογή κατασκευής «Windows Forms Application» θα αναφέρουμε απλά πως παρουσιάζει πολλά πλεονεκτήματα όσον αφορά τη δημιουργία προγράμματος με γραφικό περιβάλλον αλληλεπίδρασης με τον χρήστη (GUI). Η κεντρική ιδέα σε μια WFA (εφαρμογή φόρμας των Windows) είναι πως κάθε συστατικό του προγράμματος αποτελεί μια προκατασκευασμένη ως προς τη μορφή και τις ιδιότητες φόρμα, στην οποία μπορεί να ενσωματωθεί ο κατάλληλος κώδικας, ώστε κατά την εκτέλεση το κάθε στοιχείο να επιτελεί την επιθυμητή λειτουργία. Έτσι για παράδειγμα, το μενού όπου βρίσκονται οι επιλογές «Αρχείο-Επεξεργασία-Προβολή...» είναι φόρμα τύπου menu strip (λωρίδα μενού) που την βρίσκουμε στα εργαλεία (Toolbox) στα αριστερά της οθόνης (βλ. ΕΙΚΟΝΑ 9.2) · μετά την εισαγωγή της μπορούμε να επιλέξουμε τον αριθμό και το είδος των στοιχείων που θα περιέχει καθώς και την λειτουργία του καθενός



ΕΙΚΟΝΑ 9.2

ΕΙΣΑΓΩΓΗ ΝΕΩΝ ΔΕΔΟΜΕΝΩΝ

Για να υπάρξει ωστόσο νέο πρόγραμμα απαιτούνται δεδομένα, δηλαδή μεταβλητές και συναρτήσεις. Η εισαγωγή των δεδομένων γίνεται σε δύο τύπους αρχείων, σε αρχεία κεφαλίδας (.h) και αρχεία πηγαίου κώδικα (.cpp) τα οποία προστίθενται στο παράθυρο «Solution Explorer» όπως φαίνεται στα δεξιά της εικόνας 9.2 . Χονδρικά (παραλείπονται οι εξηγήσεις επί της C++) για αρχεία έστω Neo.h και Neo.cpp μια μέθοδος εισαγωγής δεδομένων προβλέπει τα εξής:

Βάζουμε στο Neo.h

- Δηλώσεις μεταβλητών με πρόθεμα extern
- Πρωτότυπα συναρτήσεων
- Προσθέτουμε την εντολή “#include Neo.h” στο Form1.h (κάτω από την εντολή #pragma once, εντολή που εξασφαλίζει την περίληψη του αρχείου μια μόνο φορά κατά τη μεταγλώττιση)

Βάζουμε στο Neo.cpp

- Δηλώσεις μεταβλητών χωρίς πρόθεμα extern
- Υλοποιήσεις συναρτήσεων
- Την εντολή #include “stdafx.h”

Στην περίπτωση που έχω περισσότερα του ενός αρχεία πηγαίου κώδικα ορίζω στο καθένα τις μεταβλητές που χρησιμοποιώ. Τέλος να αναφέρουμε πως η εισαγωγή δεδομένων τύπου “resources” (πχ εικόνες, ήχοι κλπ) γίνεται στο παράθυρο του «Solution Explorer», στον φάκελο «Resource Files».

ΧΕΙΡΙΣΤΕΣ ΓΕΓΟΝΟΤΩΝ (Event handlers)

Η αλληλεπίδραση μεταξύ χρήστη και προγράμματος επιτυγχάνεται μέσω των γεγονότων (events). Καλούμε γεγονός οποιαδήποτε ενέργεια του χρήστη, επί του προγράμματος, η οποία έχει προβλεφθεί από το προγραμματιστικό περιβάλλον. Ένα γεγονός «ενεργοποιείται» από τον προγραμματιστή, ώστε κατά την πραγματοποίησή του να δίνει μια σειρά από δράσεις, ουσιαστικά δηλαδή να προκαλεί την εκτέλεση ενός μπλοκ κώδικα.

Έτσι το πάτημα ενός κουμπιού, η πληκτρολόγηση δεδομένων σε ένα πλαίσιο διαλόγου ακόμα και η αιώρηση του δείκτη του ποντικιού πάνω από ένα στοιχείο αποτελούν παραδείγματα γεγονότων τα οποία προγραμματίζουμε ανάλογα με τις ανάγκες μας. Η συνάρτηση που καλείται από ένα γεγονός λέμε ότι είναι απεσταλμένη συνάρτηση του γεγονότος (delegate function), ωστόσο επειδή στα CLR προγράμματα μπορεί να δημιουργηθεί αυτόματα (όχι φυσικά το αλγοριθμικό περιεχόμενο αλλά ο σκελετός) δεν είναι ανάγκη να

ασχοληθούμε περαιτέρω με την «αποστολή» συναρτήσεων, δηλαδή την ανάθεση συγκεκριμένης συνάρτησης σε συγκεκριμένο γεγονός. Όταν έχει γίνει η ανάθεση, η αντίστοιχη συνάρτηση λειτουργεί ως event handler (χειριστής γεγονότος) και έχει την μορφή του ακόλουθου παραδείγματος:

```
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e)
{
    //ACTUAL INCLUSION COMPUTATION
    aisINapproach = inPolygon(aScout, aapproach, 4) ;
    aisINTakeoff = inPolygon(aScout, atakeoff, 6) ;

    aisINsafetyzone = inPolygon(aScout, asafetyzone, 4) ;
    transitional(aScout, atransitional) ;
    cyclic(aScout, aRin, aRout) ;
    //DISPLAY OF THE RESULTS
    textBox33->
Text=System::Convert::ToString(aisINapproach);
    textBox34->Text=System::Convert::ToString(aisINTakeoff);
    textBox35->
Text=System::Convert::ToString(aisINsafetyzone);
    textBox36->
Text=System::Convert::ToString(aisINtransitional);
    textBox37->Text=System::Convert::ToString(aisINinner);
    textBox38->Text=System::Convert::ToString(aisINouter);
    textBox39->
Text=System::Convert::ToString(aisINconical);
}
```

Εδώ παρουσιάστηκε ο χειριστής του γεγονότος «πάτημα του κουμπιού 2», το οποίο στο rPORT είναι το πρώτο από τα κουμπιά “Compute Inclusion”. Το όνομα της συνάρτησης (System::Void button2_Click) αποδίδεται αυτόματα ανάλογα με τον αποστολέα του γεγονότος (button 2) και το γεγονός που χειριζόμαστε (μονό κλικ επί του κουμπιού). Η πρώτη παράμετρος της συνάρτησης (System::Object^ sender) αναγνωρίζει την πηγή του γεγονότος (τον αποστολέα – κουμπί 2) και η δεύτερη (System::EventArgs^ e) παρέχει πληροφορίες σχετικά με το γεγονός. Στο σώμα της συνάρτησης γράφουμε τον επιθυμητό κώδικα. Στο δοσμένο παράδειγμα, μόλις πατηθεί το κουμπί 2, θα τρέξει κώδικας που ερευνά την περίληψη σημείου σε καθεμιά από τις επιφάνειες περιορισμού εμποδίων του πρώτου διαδρόμου, για επιφάνεια προσέγγισης αριστερά της επιφάνειας απογείωσης, και εμφανίζει τα αποτελέσματα, ως TRUE ή FALSE για περίληψη ή μη περίληψη, σε κουτιά κειμένου (text Box).

ΦΟΡΜΕΣ (Forms)

Ολόκληρο το πλαίσιο εκτέλεσης του προγράμματος (παράθυρο) αποτελεί μια φόρμα (Form1) και φόρμα δεν είναι παρά μια τάξη της Visual C++ .

Εκμεταλλευόμενα στο έπακρο τον αντικειμενοστραφή προγραμματισμό τα CLR προγράμματα ορίζουν τάξεις (και μέσω πολύπλοκης διαδικασίας κληρονομικότητας και άλλες παράγωγες τάξεις, ώστε όλες σχεδόν οι φόρμες να συνδέονται μεταξύ τους) που αντικατοπτρίζουν στοιχεία του γραφικού περιβάλλοντος χρήστη και αντικείμενα της καθημιάς δύνανται να εκτελέσουν διάφορες λειτουργίες. Αυτή η προκατασκευή στοιχείων εξηγεί και την ομοιότητα στην εμφάνιση των εμπορικών προγραμμάτων αφού στοιχεία όπως κουμπιά, μπάρες αναμονής, αναδιπλούμενα μενού και άλλα, συντίθενται σχεδόν αυτόματα σε αυτό και παρόμοια περιβάλλοντα προγραμματισμού.

Στο πρόγραμμα rPORT που κατασκευάστηκε, οι φόρμες που περιλαμβάνονται είναι ανά κατηγορία:

- Common Controls (κοινοί ελεγκτές) Χρησιμοποιήθηκαν οι κάτωθι
 - Button (κουμπί) : Στοιχείο που εγείρει ένα γεγονός όταν ο χρήστης το πατήσει («κλικάρει»)
 - Combo Box (συνδυαστικό κουτί) : Εμφανίζει ένα μορφοποιούμενο πλαίσιο κειμένου με αναδιπλούμενη (drop down) λίστα επιτρεπόμενων τιμών
 - Label (ετικέτα) : Παρέχει πληροφορίες χρόνου εκτέλεσης ή επεξηγηματικό κείμενο για ένα στοιχείο ελέγχου
 - Text Box (κουτί κειμένου) : Χρησιμοποιείται για είσοδο (input) ή απεικόνιση (display) δεδομένων. Δέχεται (και αποδίδει) μόνο δεδομένα τύπου text (κείμενο) ωστόσο με κατάλληλες μετατροπές τύπων μπορούμε να το χρησιμοποιήσουμε και για αριθμητικά, λογικά ή αλφαριθμητικά δεδομένα.
 - Tool Tip (πληροφορία εργαλείου) : Εμφανίζει πληροφορίες όταν ο χρήστης τοποθετεί τον δείκτη του ποντικιού πάνω στο σχετιζόμενο στοιχείο ελέγχου.
- Containers (περιέκτες) : Αφορούν στοιχεία που αναλαμβάνουν τον διαχωρισμό των δεδομένων επί της οθόνης ώστε να γίνει ευκολότερη η χρήση του προγράμματος. Χρησιμοποιήθηκαν οι κάτωθι
 - Tab Control (διαχωριστής ετικέτας) : Χωρίζει το παράθυρο εκτέλεσης του προγράμματος σε διακεκριμένα φύλλα (όμοια γίνεται και ο διαχωρισμός σε ΦΥΛΛΟ 1, ΦΥΛΛΟ 2 κλπ στο Excel) ένα εκ των οποίων μπορεί να προβάλλεται κάθε στιγμή στην οθόνη.
 - Group Box (κουτί ομάδας) : Χωρίζει τα στοιχεία κάθε οθόνης (δηλαδή κάθε ξεχωριστής προβολής) σε ομάδες με χρήση ενός διακριτού (και μορφοποιούμενου) περιγράμματος.
- Menus & Toolbars (Μενού και γραμμές εργαλείων) Χρησιμοποιήθηκαν οι κάτωθι
 - Menu Strip (Γραμμή μενού) : Προβάλλει εντολές και επιλογές του

προγράμματος που ομαδοποιούνται κατά λειτουργία (όπως πχ στο μενού του Word κάτω από την ομάδα «Επεξεργασία» υπάρχουν οι εντολές «Αναίρεση» «Αποκοπή» κλπ)

- Tool Strip (Γραμμή εργαλείων) : Παρέχει γραμμές εργαλείων και άλλα στοιχεία αλληλεπίδρασης με τον χρήστη, που υποστηρίζουν πολλές επιλογές εμφάνισης καθώς και υπερφόρτωση καλούμενων συναρτήσεων και επανάκληση αντικειμένων κατά τον χρόνο εκτέλεσης. Για παράδειγμα στο Word η εντολή «Αναίρεση» της ομάδας «Επεξεργασία» υπάρχει και ως κουμπί σε γραμμή εργαλείων (εικονίζεται με ανεστραμμένο βέλος) το οποίο καλεί την ίδια συνάρτηση με το στοιχείο της γραμμής μενού.

- Dialogs (πλαίσια διαλόγου) Χρησιμοποιήθηκαν οι κάτωθι

- Save File Dialog (πλαίσιο διαλόγου για αποθήκευση) : Προβάλλει ένα πλαίσιο διαλόγου που προτρέπει τον χρήστη να επιλέξει όνομα και τοποθεσία για το αποθηκευόμενο αρχείο.

9.2 – ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ



ΕΙΚΟΝΑ 9.3

Καθώς η ανάπτυξη του προγράμματος έγινε σε πολλά βήματα, με συνεχείς αναθεωρήσεις, βελτιώσεις και διορθώσεις θα ήταν χρονοβόρα η αναλυτική παρουσίασή της· αντ' αυτής παραθέτουμε μια ενδεικτική εκτέλεση του προγράμματος όπου εποπτικά αναφέρεται ο κώδικας που τρέχει έπειτα από κάθε ενέργεια (γεγονός). Τα κυριολεκτικά αλγοριθμικά βήματα δίδονται στο «Παράρτημα Α – Κώδικας προγράμματος». Με παρόμοιο τρόπο θα μπορούσαμε να παρακολουθήσουμε τον υπολογισμό των τιμών των μεταβλητών κατά το debug (εκσφαλμάτωση) του προγράμματος, θέτοντας σημεία ελέγχου (breakpoints) στην αρχή των χειριστών γεγονότων (event handlers), ωστόσο περιγραφή αυτού είναι προτιμότερη επί της οθόνης παρά σε αρχείο κειμένου.

Μόλις ανοίξουμε το rPORT βλέπουμε την οθόνη της εικόνας 9.3. Αναζητούμε το επιτρεπτό υψόμετρο σημείου, σε περιοχή αεροδρομίου δεδομένων χαρακτηριστικών, όπως ορίζεται από τις απαιτήσεις περιορισμού εμποδίων.

Στα text Box δεξιά της ετικέτας Determine Point, συμπληρώνουμε τις συντεταγμένες X και Y του επιθυμητού σημείου σε καθολικό σύστημα συντεταγμένων. Κατά την εκτέλεση, οι τιμές X,Y αποδίδονται στις μεταβλητές aScoutbar.x και aScoutbar.y για τον πρώτο διάδρομο και bScoutbar.x, bScoutbar.y για τον δεύτερο (οι καθολικές συντεταγμένες του σημείου δεν επηρεάζονται από τα τοπικά των διαδρόμων συστήματα συντεταγμένων ωστόσο χρησιμοποιούμε δύο ζεύγη τιμών για ευκολότερη εποπτεία των αλγορίθμων που ακολουθούν). Να θυμίσουμε πως τα text Box δέχονται μόνο κείμενο έτσι απαιτείται μετατροπή τύπων από text σε double για ανάθεση αριθμητικών τιμών στις μεταβλητές· για το λόγο αυτό η εισαγωγή του αρνητικού συμβόλου πρέπει να γίνει αφού έχουμε γράψει τον αριθμό (στην αρχή του αριθμού αλλά χρονικά τελευταίο), διαφορετικά ο υπολογιστής αδυνατεί να μετατρέψει το κείμενο παύλα (-) σε αριθμητικό δεδομένο.

Περιγράφουμε τώρα τα του διαδρόμου 1 (RUNWAY 1) ενώ τα ίδια ισχύουν και για τον 2: Επιλέγουμε τύπο προσέγγισης (Approach Type) και κωδικό αεροδρομίου (Code Number) από τα αντίστοιχα Combo Box με το πρώτο να δίνει μια τιμή εκ των 100, 200, 300 στη μεταβλητή atype και το δεύτερο να δίνει μια τιμή εκ των 1, 2, 3, 4 στη μεταβλητή acode. Η πρόσθεση atype+acode δίνει τη μεταβλητή aPortCode που είδαμε στον πίνακα 8.1 και είναι ο τρόπος με τον οποίο ξεχωρίζουμε στο πρόγραμμα αεροδρόμια διαφορετικών χαρακτηριστικών με έναν και μόνο αριθμό.

Δεξιά της ετικέτας Centre συμπληρώνουμε τις συντεταγμένες του κέντρου του διαδρόμου στο καθολικό σύστημα και αυτές ανατίθενται ως τιμές στις μεταβλητές aRW_Mbar.x , aRW_Mbar.y , aRW_Mbar.z . Παράλληλα (μόλις εισάγουμε τιμές σε αυτά τα κουτιά) ορίζονται ως μηδενικές οι οριζόντιες συντεταγμένες του κέντρου του διαδρόμου στα αντίστοιχα τοπικά συστήματα.

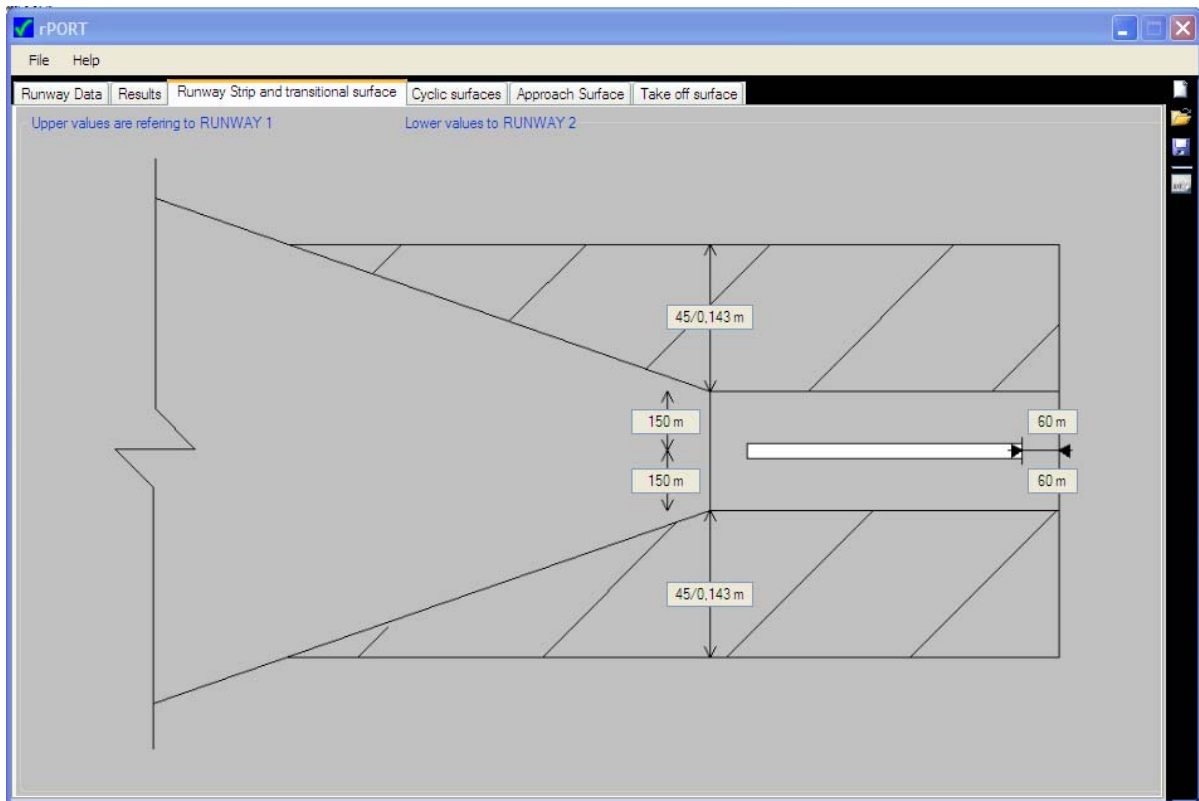
Να σημειώσουμε πως δεν γίνεται πρόβλεψη για μετατοπισμένα κατώφλια και το μήκος διαδρόμου λαμβάνεται ως η απόσταση ανάμεσα στα δύο άκρα του, ενώ ως κέντρο νοείται το μέσο της απόστασης αυτής επί του άξονα του διαδρόμου.

Δεξιά της ετικέτας Length συμπληρώνουμε το μήκος του διαδρόμου, που ανατίθεται στη μεταβλητή aRW_lenght. Μόλις αυτό δοθεί υπολογίζονται οι συντεταγμένες των άκρων του διαδρόμου aRW_AP.x , aRW_AP.y και aRW_TO.x , aRW_TO.y (στο τοπικό σύστημα) για επιφάνεια προσέγγισης αριστερά της επιφάνειας απογείωσης.

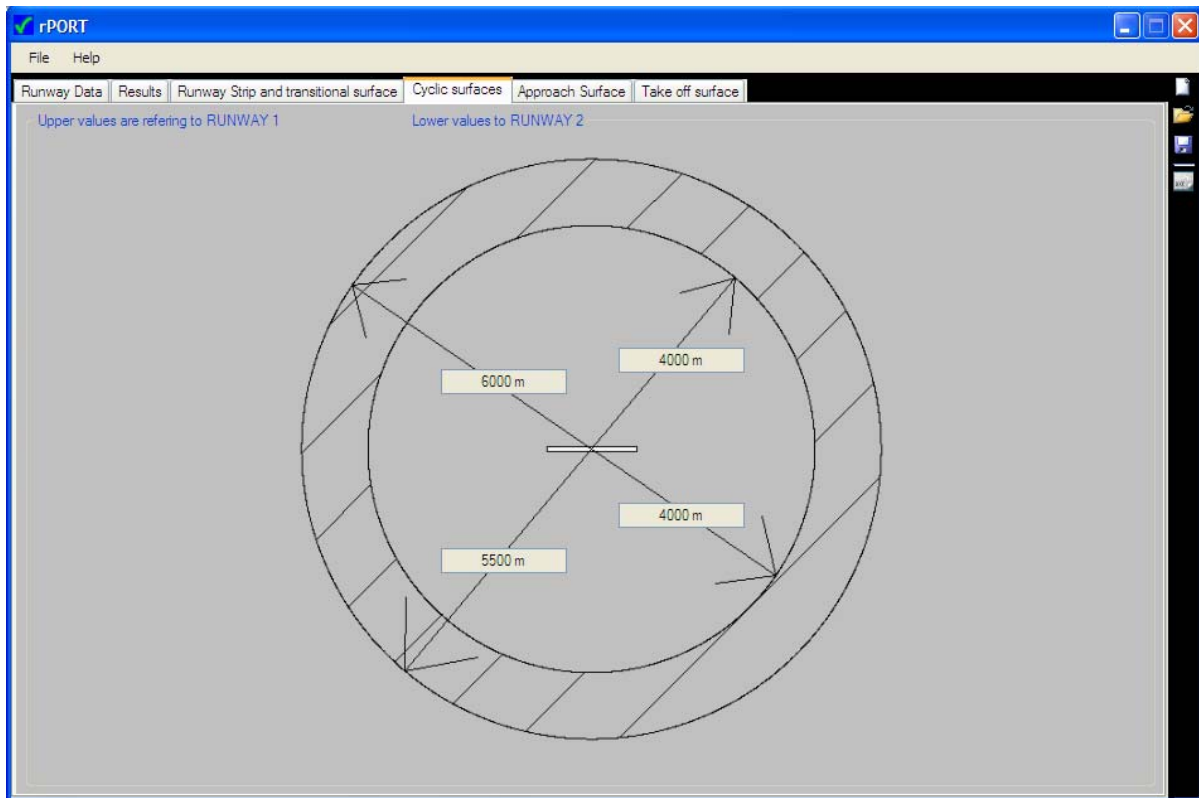
Δεξιά της ετικέτας Azimuth συμπληρώνουμε το αζιμούθιο (σε μοίρες) του διαδρόμου, τη γωνία δηλαδή του βορρά με τον άξονα του διαδρόμου. Επειδή το αζιμούθιο αναφέρεται σε σύμφωνες με τη φορά κίνησης του ρολογιού μοίρες, η αντίστοιχη μεταβλητή aAzim του προγράμματος παίρνει την αντίθετη τιμή, ώστε να είναι εκφρασμένη κατά τη μαθηματική θετική φορά. Ακόμα ορίζεται η γωνία στροφής afaaxis που διαγράφει το τοπικό σύστημα μέχρι να συναντήσει το τοπικό στραμμένο ως : $afaaxis = -(aAzim + 90)$.

Έπειτα πατάμε το κουμπί “Generate surfaces” και αρχικά (με χρήση της συνάρτησης asetDimensions) αποδίδονται τιμές στις μεταβλητές που αναφέρονται στον πίνακα 8.1 ανάλογα με την τιμή της μεταβλητής aPortCode, γίνεται δηλαδή διαστασιολόγηση των επιφανειών περιορισμού εμποδίων. Αυτό ενεργοποιεί (κάνει ορατή) την πρώτη στήλη κουμπιών, όπου ο χρήστης μπορεί να επιλέξει την επιθυμητή διεύθυνση απογείωσης – προσγείωσης. Ουσιαστικά, αν το αζιμούθιο που εισάγαμε αναφέρεται σε γωνία με τον άξονα X να δείχνει προς την μεριά της απογείωσης πατάμε το πάνω κουμπί, αν ο X δείχνει προς την προσγείωση το κάτω. Σε περίπτωση που ο διάδρομος χρησιμοποιείται και για τις δύο λειτουργίες σε κάθε διεύθυνση θα πατήσουμε και τα δύο κουμπιά. Το αποτέλεσμα της διαστασιολόγησης δίδεται στις τέσσερις τελευταίες καρτέλες και έχει την μορφή των εικόνων 9.4 - 9.7.

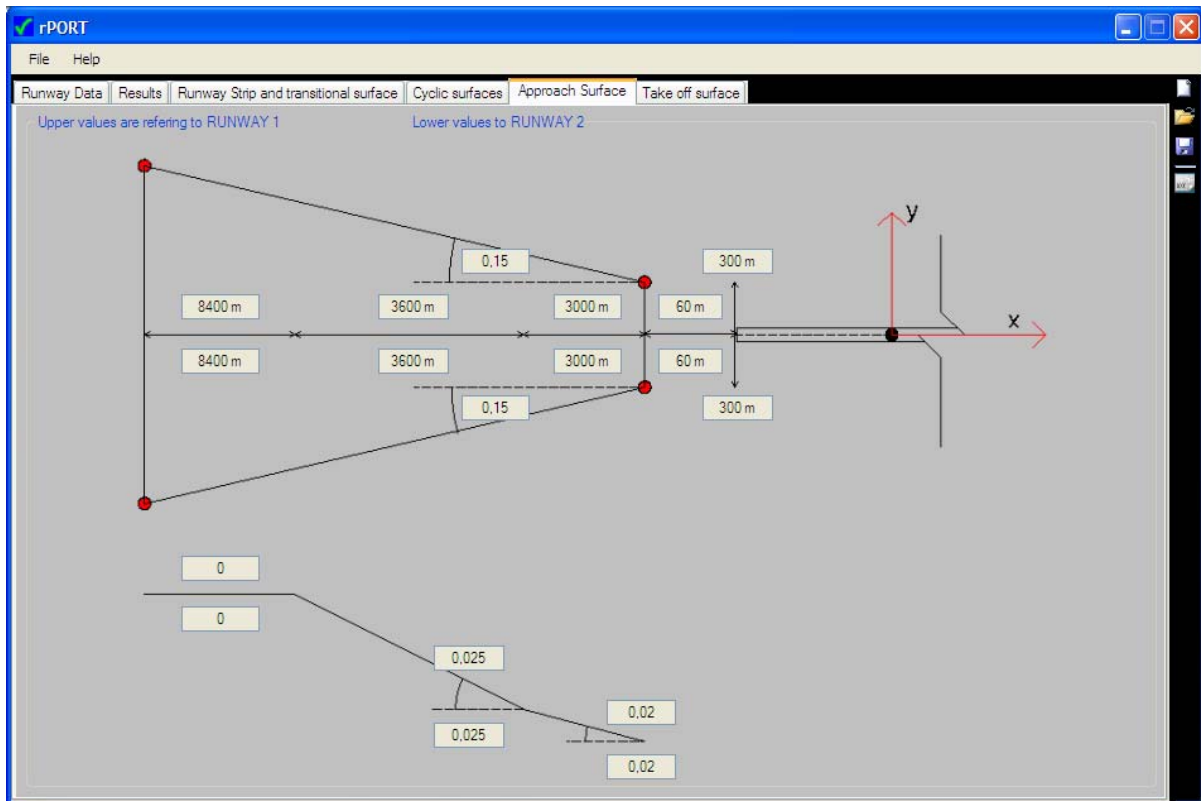
Από τις μεταβλητές (διαστάσεις) αυτές, με χρήση της συνάρτησης acreateSurfaces, παράγονται οι επιφάνειες περιορισμού εμποδίων στο τοπικό τους σύστημα σύμφωνα με τα όσα αναφέρθηκαν στο κεφάλαιο 8. Εντός του ιδίου γεγονότος (πάτημα του κουμπιού “Generate surfaces”) γίνεται και ο χειρισμός του σημείου που προσδιορίζουμε με τη συνάρτηση asetScout: το σημείο μεταφέρεται από το καθολικό σύστημα (Xbar , Ybar), στο τοπικό στραμμένο (X2 , Y2) και έπειτα στο τοπικό (X , Y) όπου είναι εκφρασμένες και οι επιφάνειες περιορισμού εμποδίων.



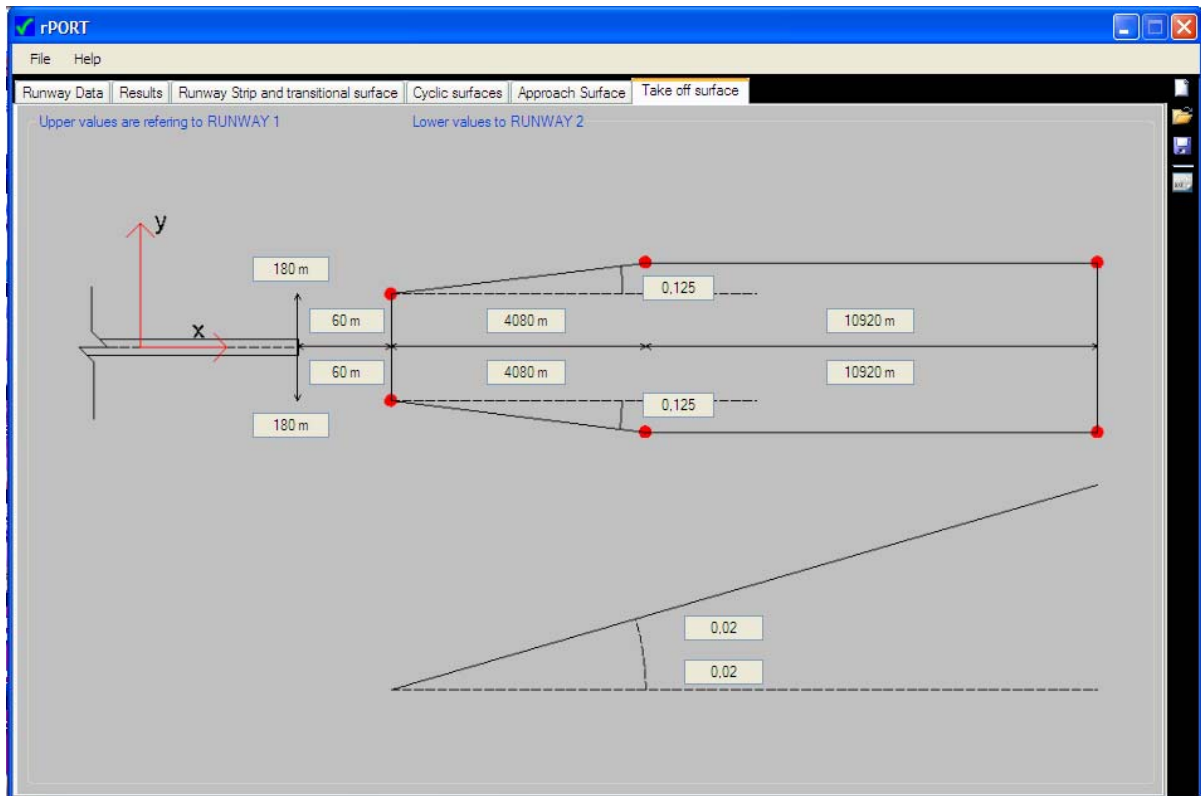
ΕΙΚΟΝΑ 9.4



ΕΙΚΟΝΑ 9.5

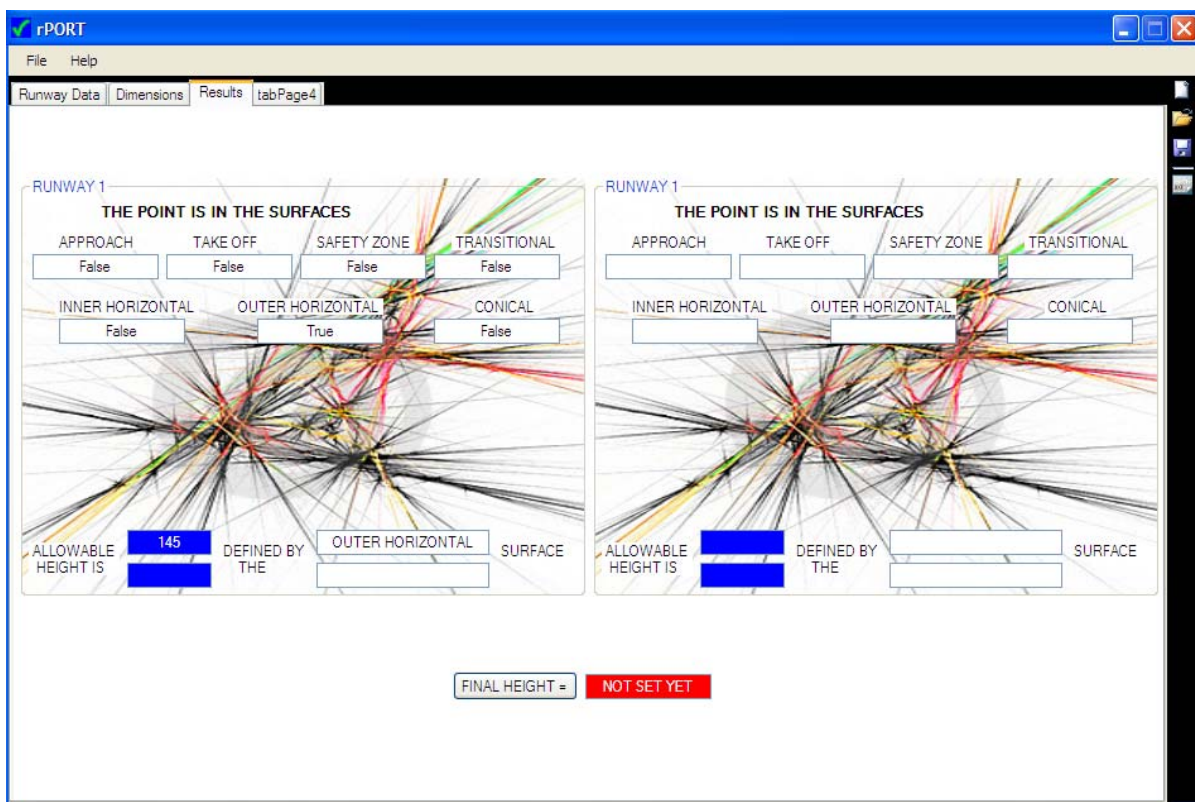


ΕΙΚΟΝΑ 9.6



ΕΙΚΟΝΑ 9.7

Ακολουθεί το πάτημα των κουμπιών της πρώτης στήλης για υπολογισμό επιτρεπτών υψομέτρων με τη δεδομένη διεύθυνση διαδρόμων. Πατώντας το “Compute Inclusion” τρέχουν οι αλγόριθμοι περίληψης του σημείου σε καθεμιά από τις επιφάνειες περιορισμού εμποδίων που περιγράφηκαν στο κεφάλαιο 7 («ΓΕΩΜΕΤΡΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ») και ανάλογα με το αποτέλεσμα οι μεταβλητές *aisINapproach*, *aisINtakeoff*, *aisINsafetyzone*, *aisINtransitional*, *aisINconical*, *aisINinner*, *aisINouter* παίρνουν τιμές *true* ή *false* (για περίληψη ή μη)· η τελική σύνθεση των αλγορίθμων φαίνεται στο παράρτημα παράθεσης του κώδικα ενώ η οπτικοποίηση των αποτελεσμάτων δίνεται στο πάνω μέρος της καρτέλας “Results” όπως φαίνεται στην εικόνα 9.5. Το κάθε κουμπί «Compute Inclusion» ενεργοποιεί και το αντίστοιχό του Compute Height».



ΕΙΚΟΝΑ 9.8

Πατώντας το κουμπί «Compute Height» εκτελείται η συνάρτηση *acomputeHeight* η οποία υπολογίζει το επιτρεπτό υψόμετρο ανάλογα με την επιφάνεια στην οποία βρίσκεται το σημείο. Το αριθμητικό αποτέλεσμα φαίνεται στο κάτω μέρος της καρτέλας «Results» δεξιά της ετικέτας

“ALLOABLE HEIGHT IS” ενώ παρέχεται και η πληροφορία ως προς το ποια ήταν η κρίσιμη επιφάνεια που όρισε το υψόμετρο.

Πανομοιότυπες ενέργειες εκτελούνται με το πάτημα των κουμπιών της κάτω σειράς, για αντίθετη διάταξη απογείωσης-προσγείωσης (σε περίπτωση χρήσης του διαδρόμου για απογείωση και προσγείωση και στις δύο διευθύνσεις πατάμε και τις δύο σειρές κουμπιών). Τα αποτελέσματα για την περίληψη ή μη του σημείου αντικαθιστούν τα παλιά ενώ τα αριθμητικά αποτελέσματα επιτρεπτού ύψους αναγράφονται κάτω από τα προηγούμενα.

Αν υπάρχει και δεύτερος διάδρομος εκτελούνται οι ίδιες ενέργειες, για τα στοιχεία ελέγχου της ομάδας RUNAWAY 2, και παρόμοιος κώδικας τρέχει στο παρασκήνιο. Τελικά με το πάτημα του κουμπιού «FINAL HEIGHT» επιλέγεται το κρισιμότερο υψόμετρο που ορίζουν όλοι οι πιθανοί συνδυασμοί, αποδίδεται στη μεταβλητή NikosAthanasίου και εμφανίζεται στο διπλανό κουτί κειμένου (σε κόκκινο φόντο).

9.3 – ΔΟΚΙΜΑΣΤΙΚΗ ΕΞΟΔΟΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Μετά από εισαγωγή δεδομένων και εκτέλεση του προγράμματος επιλέγουμε από το μενού «File» την εντολή «Export Printable Text». Η σχετιζόμενη με την εντολή συνάρτηση παράγει ένα αρχείο κειμένου, το όνομα και την θέση αποθήκευσης του οποίου μπορούμε να μορφοποιήσουμε στο αντίστοιχο πλαίσιο διαλόγου. Το είδος της εξόδου είναι πάντοτε .txt και δεν απαιτείται η εισαγωγή της κατάληξης αυτής στο όνομα του αρχείου (προστίθεται με συνάρτηση χειρισμού αλφαριθμητικών)

Μια δοκιμαστική έξοδος του προγράμματος (όπου φαίνονται και τα δεδομένα εισόδου) είναι η παρακάτω:

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
School of Civil Engineering - Department of Transportation Planning and
Engineering

ATHANASIOU NIKOLAOS cod 01102069 - DIPLOMATIKH ERGASIA

SURVEY OF ALLOABLE ALTITUDE DUE TO RESTRICTION BY THE
OBSTACLE LIMITATION SURFACES

***** DETERMINE ALTITUDE FOR THE POINT *****

X=4622.000000 Y=2346.000000

The local coordinate system for each runway is set to its centre with
the X axis at the runway's axis

RUNWAY 1 - DATA

POINT in local coordinate system is: X=5143.678212 Y=-639.667451

Approach Type -> Non instrument

Code number -> 3

Runway Centre: X = 0.000000 Y = 0.000000 Altitude = 0.000000

Length(meters) : 1800.000000

Azimuth(counter-clockwise degrees) : 56.000000

RUNWAY 1 - RESULTS

1st alloable height = 104.165005

2nd alloable height = 104.165005

RUNWAY 2 - DATA

POINT in local coordinate system is: X=-2516.358274 Y=-5629.567837

Approach Type -> No presicion

Code number -> 3

Runway Centre: X = 345.000000 Y = 6788.000000 Altitude = 55.000000

Length(meters) : 2000.000000
 Azimuth(counter-clockwise degrees) : 22.000000

RUNWAY 2 - RESULTS

1st alloable height = 175.000000
 2nd alloable height = 175.000000

F I N A L A L T I T U D E = 104.165005 meters

COORDINATES OF THE OBSTACLE LIMITATION SURFACES IN EACH RUNWAY'S LOCAL SYSTEM

RUNWAY 1

SAFETY ZONE

X=960.000000 Y=-75.000000
 X=960.000000 Y=75.000000
 X=-960.000000 Y=75.000000
 X=-960.000000 Y=-75.000000

APPROACH SURFACE

X=-960.000000 Y=75.000000
 X=-3960.000000 Y=375.000000
 X=-3960.000000 Y=-375.000000
 X=-960.000000 Y=-75.000000

TAKE OFF SURFACE

X=960.000000 Y=-90.000000
 X=5040.000000 Y=-600.000000
 X=15960.000000 Y=-600.000000
 X=15960.000000 Y=600.000000
 X=5040.000000 Y=600.000000
 X=960.000000 Y=90.000000

CONICAL SURFACE

Rin=4000 Rout=5500

TRANSITIONAL

X=960.000000 Y=-389.685315
 X=960.000000 Y=389.685315
 X=-4106.853147 Y=389.685315
 X=-4106.853147 Y=-389.685315

RUNWAY 2

SAFETY ZONE

X=1060.000000 Y=-150.000000
X=1060.000000 Y=150.000000
X=-1060.000000 Y=150.000000
X=-1060.000000 Y=-150.000000

APPROACH SURFACE

X=-1060.000000 Y=150.000000
X=-16060.000000 Y=2400.000000
X=-16060.000000 Y=-2400.000000
X=-1060.000000 Y=-150.000000

TAKE OFF SURFACE

X=1060.000000 Y=-90.000000
X=5140.000000 Y=-600.000000
X=16060.000000 Y=-600.000000
X=16060.000000 Y=600.000000
X=5140.000000 Y=600.000000
X=1060.000000 Y=90.000000

CONICAL SURFACE

Rin=4000 Rout=5500

TRANSITIONAL

X=1060.000000 Y=-464.685315
X=1060.000000 Y=464.685315
X=-3157.902098 Y=464.685315
X=-3157.902098 Y=-464.685315

9.4 – ΟΠΤΙΚΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Είναι γεγονός ότι μεγάλο ρόλο στην χρησιμότητα και πρακτικότητα ενός προγράμματος, καθώς και στον έλεγχο της ορθότητας των υπολογισμών παίζει η δυνατότητα κάποιας οπτικοποίησης των αποτελεσμάτων. Στην περίπτωση μας, επιθυμητό θα ήταν να παράγεται η προβολή σε οριζόντιο επίπεδο των επιφανειών περιορισμού εμποδίων και του ερευνούμενου σημείου.


Στο περιβάλλον CLR υπάρχει η δυνατότητα δημιουργίας γραφικών, ωστόσο επειδή βασική μονάδα μέτρησης είναι το pixel και εμείς απαιτούμε σχέδια αντί σκίτσων, θα χρειαστούν πολύπλοκες μεταφορές σε διάφορα συστήματα ώστε να καταλήξουμε στο (επιθυμητής κλίμακας) τελικό σχέδιο. Επίσης είναι γεγονός πως η όποια οπτικοποίηση επί της οθόνης γινόταν με αυτό τον τρόπο, θα απαιτούσε επιπλέον προγραμματισμό για την παραγωγή εκτύπωσης.

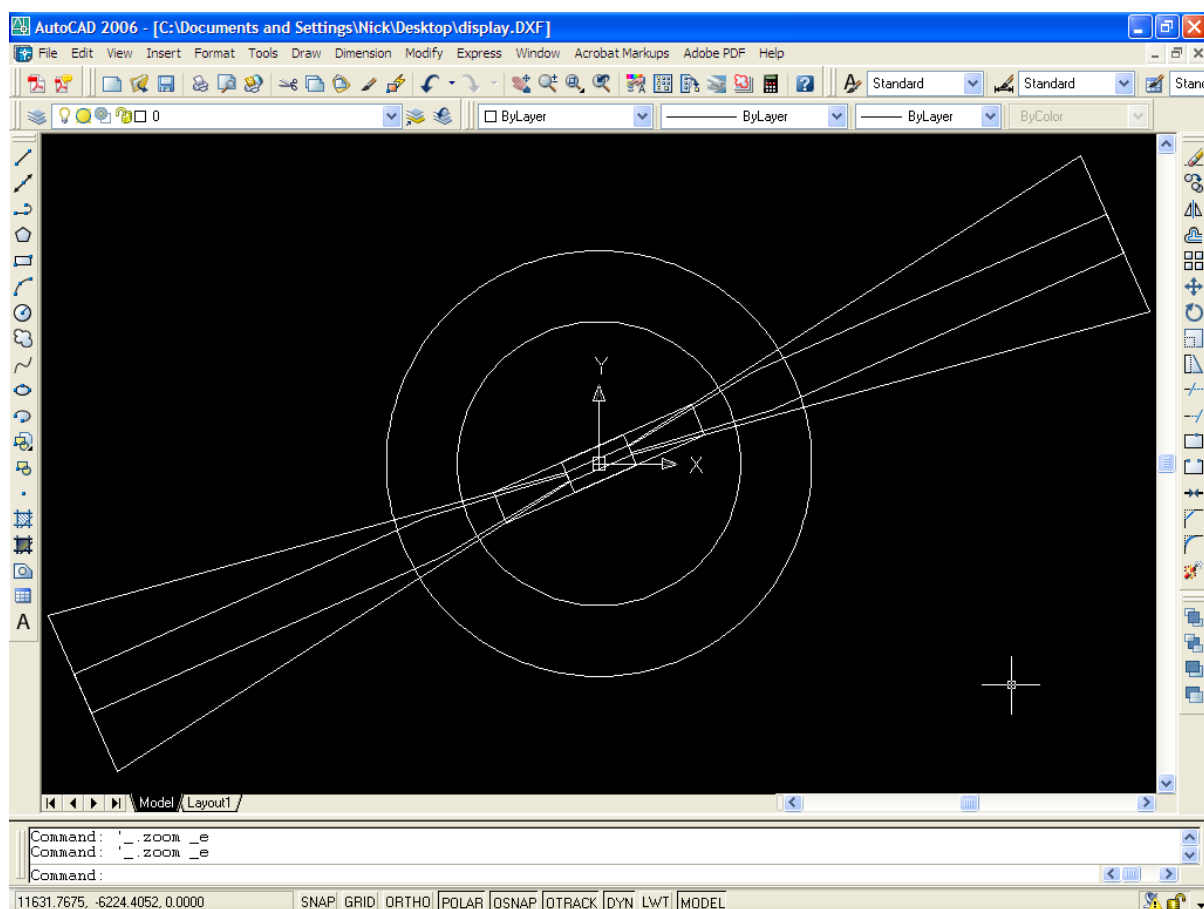
Η αντιπρόταση έρχεται από τα CAD περιβάλλοντα και από το πιο γνωστό εξ αυτών, το AutoCAD. Η Autodesk δημιούργησε το DXF, μια επισημασμένη (tagged) μορφή αναπαράστασης πληροφοριών των αρχείων σχεδίου (.dwg). Το DXF είναι δεδομένα σε μορφή κώδικα ASCII, δηλαδή μπορεί να δημιουργηθεί όπως τυπώνουμε δεδομένα σε αρχείο .txt, και όλες οι πιθανές μορφές αρχείου σχεδίου .dwg μπορούν να αποθηκευτούν σε αυτό και να διαβαστούν είτε από επεξεργαστή κειμένου είτε από το AutoCAD. Η μορφή που έχει ένα DXF αρχείο έχει ενδεικτικά ως εξής :

```
(...)
2
DIMSTYLE
5
A
330
0
100
AcDbSymbolTable
70
1
100
AcDbDimStyleTable
71
0
0
DIMSTYLE
105
27
330
A
(...)
```

Το DXF έχει το πλεονέκτημα πως δημιουργούμε απευθείας εκτυπώσιμο σχέδιο με τα χαρακτηριστικά, την κλίμακα και τις επαγγελματικές

προδιαγραφές που επιθυμούμε, χωρίς πολύπλοκα προγραμματιστικά μηχανεύματα, ωστόσο ένα μειονέκτημα είναι η απαιτούμενη έκταση κώδικα που για ένα κενό αρχείο AutoCAD φτάνει περί τις 50 σελίδες.

Στο rPORT υπάρχει το κουμπί  (Export AutoCAD data), πάτημα του οποίου ανοίγει πλαίσιο όπου δηλώνουμε το όνομα του αρχείου στο οποίο θέλουμε να αποθηκεύσουμε το σχετικό σχέδιο. Σημειώνουμε πως κάθε φορά εξάγονται σε σχέδιο μόνο οι επιφάνειες που έχουμε δηλώσει (για τον αντίστοιχο διάδρομο και διεύθυνση). Η αποθήκευση σε DXF παράγεται όπως και το κείμενο .txt του προγράμματος, με τύπωση δεδομένων σε αρχείο (βλ. Παράρτημα Α – Κώδικας προγράμματος, export2AutoCad.cpp) και ενδεικτική έξοδος είναι η παρακάτω, όπου εικονίζονται οι επιφάνειες περιορισμού εμποδίων σε κάτοψη, για έναν διάδρομο με δυνατότητα εκτέλεσης προσγειοαπογειώσεων και στις δύο διευθύνσεις.



ΕΙΚΟΝΑ 9.9

ΚΕΦΑΛΑΙΟ 10

ΠΡΟΤΑΣΕΙΣ ΕΠΕΚΤΑΣΗΣ – ΧΡΗΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ rPORT

Στο παρόν κεφάλαιο παρουσιάζονται αρχικά προτάσεις επέκτασης και βελτίωσης του προγράμματος rPORT. Εξαιτίας περιορισμένου χρόνου και έκτασης μιας διπλωματικής εργασίας οι ιδέες αυτές δεν ήταν δυνατό να υλοποιηθούν, αν και όπως θα φανεί στη συνέχεια, αρκετή έρευνα έχει γίνει για να επιλυθούν τα προβλήματα με τα οποία καταπιάνονται. Ο τρόπος με τον οποίο έγινε η ανάπτυξη του rPORT, δηλαδή ως (Tabbed) πρόγραμμα με καρτέλες, επιτρέπει την προσθήκη νέων εργαλείων σε ξεχωριστές καρτέλες και έτσι έχει προβλεφθεί η άνετη ενσωμάτωση νέου κώδικα που θα λειτουργεί μαζί ή ξεχωριστά από τον προϋπάρχοντα.

Στη συνέχεια δίδονται περιπτώσεις χρήσης του προγράμματος είτε ως έχει είτε με μικρές τροποποιήσεις. Αν και κατά την κύρια χρήση του υπήρξε ως αντικείμενο μελέτης των συστατικών που το δημιούργησαν (γεωμετρικοί αλγόριθμοι, υπολογιστική γεωμετρία, αντικειμενοστραφής προγραμματισμός) επιχειρείται εδώ μια πιο πρακτική προσέγγισή του.

10.1 – ΒΕΛΤΙΩΣΕΙΣ

Υπόλοιπες επιφάνειες

Όπως αναφέρεται στο κεφάλαιο 8 «Για τους σκοπούς της διπλωματικής εργασίας δεν μελετάται το σύνολο των επιφανειών περιορισμού εμποδίων αλλά οι κυριότερες εξ αυτών καθώς κύριο μέλημα είναι να οριστούν οι μέθοδοι και τα γνωστικά πεδία (μαθηματικά εργαλεία) που μας επιτρέπουν να καταπιαστούμε με γεωμετρικά (στη βάση τους) προβλήματα σε προγραμματιστικά περιβάλλοντα». Επόμενο βήμα θα ήταν λοιπόν η προσθήκη και ο γεωμετρικός έλεγχος και των υπολοίπων επιφανειών, καθώς και των στοιχείων του γραφικού περιβάλλοντος χρήστη που θα επέτρεπαν τον πλήρη (και εύχρηστο) ορισμό και χειρισμό τους.

Συστήματα αναφοράς

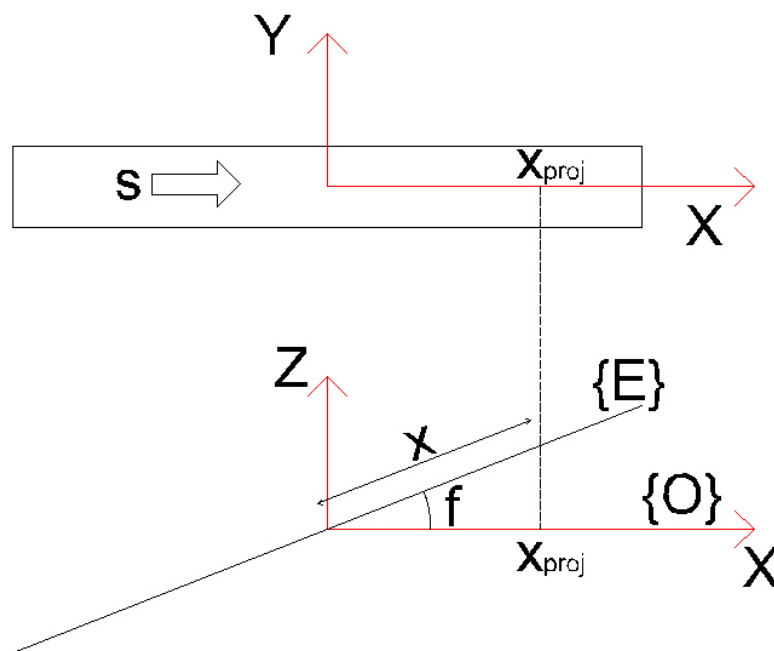
Τα συστήματα συντεταγμένων που χρησιμοποιήσαμε ήταν καρτεσιανά, ωστόσο δεν έγινε ειδική μνεία για τα συστήματα αναφοράς και μετατροπές μεταξύ αυτών. Στο Annex 14 του ICAO ορίζονται τα συνήθη συστήματα αναφοράς (Common reference systems) για οριζόντιες (World Geodetic System 1984 / WGS-84) και κατακόρυφες (Mean sea level – MSL datum) συντεταγμένες και αποτελούν δεδομένο που ένα επαγγελματικό πρόγραμμα πρέπει να λαμβάνει υπόψη όταν συνεργάζεται με άλλα προγράμματα και συσκευές μέτρησης (πχ στην περίπτωση οργάνων GIS).

Μηκοτομή διαδρόμου

Το προφίλ της μέσης γραμμής του διαδρόμου θεωρήθηκε σταθερό και οριζόντιο. Ωστόσο κάτι τέτοιο είναι πρακτικά ανεφάρμοστο. Παρουσιάζεται εδώ μια μέθοδος επίλυσης στην περίπτωση μονής κλίσης (ανύψωσης) του διαδρόμου.

Η κλίση επί του διαδρόμου επηρεάζει (ανυψώνει και βυθίζει) τις επιφάνειες περιορισμού εμποδίων. Ως τώρα η περίληψη σε αυτές ερευνούταν επί της προβολής των· ωστόσο το σύνολο της διαστασιολόγησης δινόταν σε προβολή (από ICAO) επομένως η παραπάνω μέθοδος εργασίας ήταν σχετική προφανής.

Θεωρούμε το επίπεδο προβολής των επιφανειών περιορισμού εμποδίων $\{E\}$ (επί του οποίου δίδονται οι διαστάσεις) και το οριζόντιο επίπεδο $\{O\}$ επί του οποίου έχουμε τις συντεταγμένες του σημείου για το οποίο ερευνούμε το επιτρεπτό υψόμετρο. Θεωρούμε επίσης πως είναι γνωστές (από την ως τώρα εργασία) οι μεταφορές από καθολικό σε τοπικό στραμμένο και τοπικό σύστημα (βλ. κεφ. 7 - 8)



ΣΧΗΜΑ 10.1

Ακολουθώντας την λογική που διέπει ως τώρα το πρόγραμμα, αντί να προβάλλουμε τις επιφάνειες από $\{E\}$ σε $\{O\}$, θα μεταφέρουμε το σημείο από το οριζόντιο επίπεδο στο $\{E\}$. Η έρευνα αναφέρεται σε έρευνα οριζοντιογραφίας καθώς το επιτρεπτό ύψος που βρίσκουμε ανάγεται πολύ εύκολα σε διάδρομο με κλίση, προσθέτοντας την υψομετρική διαφορά που προκύπτει από την ανύψωση (ή βύθιση) του επιπέδου $\{E\}$ σε απόσταση x.

Ισχύει

$$f = \arctan s$$

$$\cos f = \frac{x_{proj}}{x} \Rightarrow x = \frac{x_{proj}}{\cos(\arctan s)}$$

Η τετμημένη X_{proj} θεωρείται η αρχική του σημείου, ενώ η X είναι η αναγόμενη στο επίπεδο $\{E\}$. Η τεταγμένη δεν μεταβάλλεται. Η υπόλοιπη διαδικασία είναι όμοια με την κανονική, με την ερευνά περίληψης να γίνεται για σημείο (X,Y) και στο επιτρεπτό υψόμετρο να προσθαφαιρείται η υψομετρική διαφορά του $\{E\}$ στο σημείο X .

Παρόμοια διαδικασία θα ακολουθείται στην περίπτωση εγκάρσιας κλίσης, που θα επηρεάζει την τεταγμένη του σημείου. Παρότι το ημιπλάτος είναι μικρό σχετικά με το μήκος, οι επιφάνειες περιορισμού εμποδίων επηρεάζονται σε μεγάλες αποστάσεις επομένως οι αποκλίσεις είναι σημαντικές.

Αμφιθεατρικές επιφάνειες

Κατά ICAO προβλέπεται, ως εναλλακτική προσέγγιση, η λήψη εσωτερικής οριζόντιας επιφάνειας με δύο σημεία αναφοράς επί του διαδρόμου. Η πορεία εργασίας σε αυτή την περίπτωση περιγράφεται στο «ΠΑΡΑΡΤΗΜΑ Β - ΕΛΕΓΧΟΣ ΣΕ ΑΜΦΙΘΕΑΤΡΙΚΕΣ ΕΠΙΦΑΝΕΙΕΣ».

Μετατοπισμένο κατώφλι

Η πρόβλεψη για μετατοπισμένα κατώφλια, με τις αντίστοιχες μεταβολές στο GUI, θα συνέβαλε ως βελτίωση στην εξέλιξη του προγράμματος. Πάραυτα ο χρήστης μπορεί να εισάγει ως κέντρο το μέσο της απόστασης των κατωφλιών, η οποία θα εισάγεται και ως μήκος διαδρόμου, και να έχει το επιθυμητό αποτέλεσμα, προσέγγιση η οποία όμως αγνοείται καθώς η εκτέλεση υπολογισμών από το χρήστη είναι αυτό που προσπαθούμε να αποφύγουμε εξ αρχής με τη δημιουργία προγράμματος.

10.2 – ΠΡΟΣΘΗΚΗ ΕΡΓΑΛΕΙΩΝ

Μήκος διαδρόμου

Το μήκος διαδρόμου θεωρείται γνωστό και εισάγεται ως δεδομένο στο GUI του προγράμματος. Ωστόσο θα μπορούσε να υπάρχει ένα εργαλείο (σε ξεχωριστό Tab) όπου θα δηλώνονταν τα ακόλουθα δεδομένα:

- Χαρακτηριστικά αεροσκάφους σχεδιασμού (ή και τύπος του)
- Θερμοκρασίες T1 και T2 (βλ. κεφ. 4.1)

- Υψόμετρο αεροδρομίου
- Κατά μήκος κλίση διαδρόμου

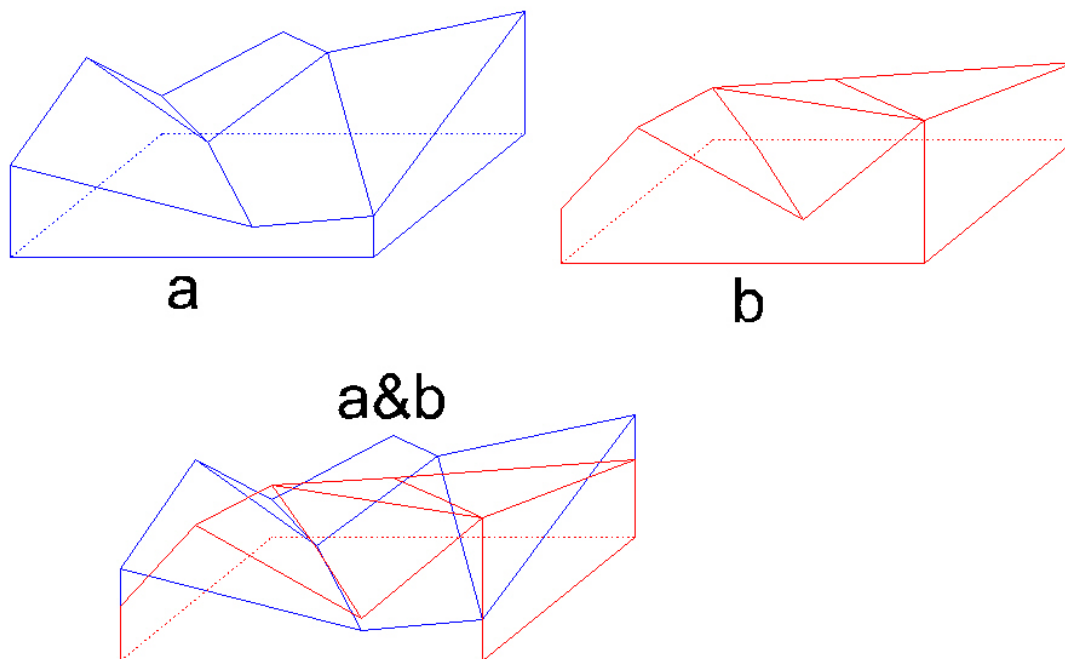
και από τα αλγοριθμικά βήματα που αφορούν αυτά (βλ. κεφ. 4.1) θα εξαγόταν το τελικό μήκος του διαδρόμου. Ωστόσο η επιπλέον εργασία που απαιτούν είναι καθαρά προγραμματιστικής φύσης και πέρα από το πρακτικό δεν έχει μαθηματικό(γεωμετρικό) ενδιαφέρον (ή άλλο ενδιαφέρον για την επιστήμη του πολιτικού μηχανικού)

10.3 – ΧΡΗΣΕΙΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Η προφανής χρήση του προγράμματος είναι για την εύρεση επιτρεπτού ύψους σημείου, όπως ορίζεται από τις επιφάνειες περιορισμού εμποδίων. Η διαδικασία αυτή είναι απαραίτητη κατά την επέκταση ενός αερολιμένα και δημιουργία νέων εγκαταστάσεων που πρέπει να ελεγχθούν ως προς το ύψος. Μια μικρή μετατροπή στο GUI, θα επέτρεπε να ελεγχθούν απευθείας ολόκληρα κτίρια, εισάγοντας τις κορυφές του κυρτού τους περιβλήματος (ουσιαστικά διαδοχική επανάληψη του υφιστάμενου ελέγχου) και παίρνοντας ως αποτέλεσμα έγκριση ή μη (ή ακόμη ποια περιοχή είναι εκτός ορίων, αν και αυτό θα απαιτούσε παραπάνω από μετατροπή του GUI).

Για μία επιπλέον χρήση του προγράμματος στη φάση σχεδιασμού ενός αεροδρομίου, θα έπρεπε να σχετισθεί με τους χωματισμούς. Αν δεδομένο εισόδου γίνουν τα σημεία εδάφους (πχ επί ορθογωνικού κανάβου) είναι απλό με διαδοχική επανάληψη του υφιστάμενου ελέγχου να ευρεθεί για κάθε σημείο το αντίστοιχο επιτρεπτό υψόμετρο. Ο όγκος των χωματισμών προκύπτει ως ο όγκος του στερεού που βρίσκεται ανάμεσα στις δύο επιφάνειες, την επιφάνεια εδάφους και την επιφάνεια που προκύπτει ως κυρτό περίβλημα από τα σημεία επιτρεπτού υψομέτρου, θεωρώντας την πρώτη ως ανώτερη. Ένας τρόπος για την εύρεση του όγκου αυτού είναι η χρήση των βιβλιοθηκών γεωμετρικών αλγορίθμων που περιγράφηκαν στο κεφάλαιο 7.

Με μεθόδους υπολογιστικής γεωμετρίας είναι δυνατό να βρεθεί οποιοσδήποτε όγκος μεταξύ δύο επιφανειών. Ίσως το δυσκολότερο κομμάτι είναι η κατασκευή αλγορίθμων που δίνουν το κυρτό περίβλημα σε τρεις διαστάσεις, αν και κάτι τέτοιο περιγράφεται στο βιβλίο του R. Sedgwick “Algorithms in C++” ως απλή επέκταση του αντίστοιχου δισδιάστατου προβλήματος (ουσιαστικά ως ταξινόμηση πίνακα).



ΕΙΚΟΝΑ 10.1

Μια ακόμη εφαρμογή του rPORT είναι για τη χωροθέτηση εγκαταστάσεων σε αερολιμένα. Στη φάση σχεδιασμού, μπορεί να προκύψουν κάποια στοιχεία διαστασιολόγησης εγκαταστάσεων ως υποχρεωτικά· με τον έλεγχο επιτρεπτών υψομέτρων μπορούμε να τα επανατοποθετήσουμε ώστε να μην παραβαίνουν τους κανονισμούς.

Επίσης η αυτόματη παραγωγή της οριζοντιογραφίας των επιφανειών, σε αξιοποιήσιμη σχεδιαστικά μορφή (.dxf), που παρέχει το πρόγραμμα, μπορεί να θεωρηθεί αρκετά χρήσιμη ως εργαλείο σχεδίασης και εποπτείας των επιφανειών. Η μορφή αυτή επιτρέπει την «συνεργασία» του προγράμματος με πολλά άλλα μέσω του AutoCAD, αυξάνοντας έτσι το πεδίο δράσης του rPORT.

Τέλος το γεγονός πως οι νέοι (custom) τύποι δεδομένων δηλώνονται εκτός των τάξεων του CLR (Form1), όπως και οι νέες συναρτήσεις και μεταβλητές, κάνει τον κώδικα ανεξάρτητο του GUI και του .NET κάτι που είναι εξαιρετικά σημαντικό για την φορητότητά του (αξιοποίηση σε διαφορετικά συστήματα) και την αξιοποίηση μέρους η του συνόλου του σε μεγαλύτερα προγράμματα.

ΚΕΦΑΛΑΙΟ 11

ΣΥΜΠΕΡΑΣΜΑΤΑ

Παρά τη μικρή τους έκταση, τα γραφόμενα διατηρήθηκαν σε ξεχωριστό κεφάλαιο εξαιτίας της σημασιολογικής ανομοιογένειας με τα προηγούμενα. Παραθέτουμε εδώ κάποια συμπεράσματα που προέκυψαν κατά την ανάπτυξη του προγράμματος rPORT και τη μελέτη των σχετικά με αυτό αντικειμένων. Η γενικότερη αυτή εμπειρία λοιπόν καταλήγει στα εξής:

- Η υπολογιστική γεωμετρία αποτελεί μαθηματικό εργαλείο που με τα σύγχρονα δεδομένα μοιάζει να γίνεται απαραίτητο για τον μηχανικό, εξαιτίας της «γεωμετρικής υφής» των προβλημάτων στα οποία καλείται να δώσει λύση.
- Καθώς μεγάλο μέρος του επιστημονικού πεδίου ενός συγκοινωνιολόγου άπτεται (καθαρά) γεωμετρικών (στη βάση τους και όχι σχετικά με τους συνολικούς παράγοντες σχεδιασμού) προβλημάτων (πχ χάραξη οδών και κόμβων, θέματα ορατότητας κλπ), η μέθοδος εργασίας που ακολουθήθηκε στην διπλωματική αυτή είναι ικανή να παράγει λογισμικά πακέτα που θα μελετούν και θα σχεδιάζουν τη λύση σε συγκοινωνιακά - γεωμετρικά προβλήματα. Να αναφέρουμε ως παράδειγμα πως η χάραξη σταγόνας θα μπορούσε να παραμετροποιηθεί με παρόμοιο τρόπο και να έχουμε παραγωγή τελικού σχεδίου σε AutoCAD όπως περιγράφεται στην παράγραφο 9.4 (DXF)
- Η χρήση C++ παρότι ως τώρα αντενδείκνυτο σε μη έμπειρους προγραμματιστές, λόγω πολυπλοκότητας της γλώσσας, με τις τελευταίες προσθήκες του CLR γίνεται δελεαστική για τη δημιουργία γραφικού περιβάλλοντος αλληλεπίδρασης με τον χρήστη, πρωτεύοντος στοιχείου χρηστικότητας ενός προγράμματος
- Ευκόλως εννοείται βέβαια ότι η χρήση προγραμμάτων όπως το rPORT απλοποιεί και επιταχύνει τους εκτελούμενους ελέγχους και υπολογισμούς, ωστόσο είναι η εμπειρία και οι γνώσεις του μηχανικού που θα οδηγήσουν σε τελικές αποφάσεις, εποπτεία επί των αποτελεσμάτων και ορθή χρήση κάθε είδους λογισμικού.
- Κάθε πρόγραμμα που σχεδιάζεται για μηχανικούς παρουσιάζει την ιδιαιτερότητα να αναφέρεται σε ένα επιστημονικό πεδίο που συνεχώς ανανεώνεται και αναθεωρείται. Θα πρέπει να έχει γίνει πρόβλεψη λοιπόν ώστε ο κώδικας να είναι «δεκτικός» απέναντι σε αυτό, αφού όχι μόνο δεδομένες «σταθερές» και συντελεστές μπορούν να αλλάξουν τιμή, αλλά και ολόκληρο το φυσικό υπόβαθρό πίσω από κάποιον υπολογισμό να θεωρηθεί απαρχαιωμένο και να απαιτεί αντικατάσταση.
- Η σύνταξη επαγγελματικών προγραμμάτων για μηχανικούς αποτελεί εργασία που απαιτεί τη συνεργασία περισσότερων ειδικοτήτων. Σαφώς

αποκλειστικά ένας προγραμματιστής αδυνατεί να την εκτελέσει ελλείψει θεωρητικού υποβάθρου, αλλά εξαιτίας αυξημένων προγραμματιστικών απαιτήσεων είναι αναμενόμενο ένας μηχανικός (εξαιρουμένου του ηλεκτρολόγου φυσικά, ο οποίος ωστόσο υποπίπτει στην πάνω κατηγορία) να μην μπορεί να ανταπεξέλθει μόνος του. Είναι θεμιτό επομένως αντίστοιχη συνεργασία να υπάρχει και στην έρευνα, όπου αναπτύσσονται παρόμοια (σαφώς συνθετότερα) προγράμματα.

- Με αυτό (πρόγραμμα rPORT) και παρόμοια υπολογιστικά – σχεδιαστικά εργαλεία να βρίσκονται στην διάθεσή του, ένας μηχανικός είναι σε θέση να διευρύνει τις επιλογές του ως προς τον τρόπο εργασίας, να ελαχιστοποιήσει την πιθανότητα λάθους, να αυξήσει την ακρίβεια και την ταχύτητα των υπολογισμών του και να εντρυφήσει στο πρόβλημα που μελετά χωρίς να τον απασχολεί ο όγκος των υπολογισμών. Επεκτείνει έτσι τις δυνατότητες και δεξιότητές του και είναι σε θέση να ασχοληθεί με θέματα τα οποία στο παρελθόν φάνταζαν δυσπρόσιτα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Κ.Γ. Αμπακούμκιν, Αεροδρόμια, Εκδόσεις Συμμετρία, Αθήνα 1990
- [2] Annex 14, Aerodromes, International Civil Aviation Organization, July 2004
- [3] Γ.Ζ. Εμίρης, Υπολογιστική Γεωμετρία, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Αθήνα 2006
- [4] Ν. Μ. Σγούρος – ΠΜΣ, Διδακτική της Τεχνολογίας & Ψηφιακά Συστήματα (Γεωμετρικοί μετασχηματισμοί Σχημάτων)
- [5] Thomas H. Cormen - Charles E. Leiserson - Ronald L. Rivest - Clifford Stein, Introduction to Algorithms, The MIT Press, 2001
- [6] Robert Sedgewick, Algorithms in C++, Princeton University, 1992
- [7] Joseph O' Rourke, Computational Geometry in C, Cambridge University press
- [8] Herbert Schildt, C++ from the ground up, 2003
- [9] Ivor Horton, Beginning Visual C++ 2008, Wiley Publishing Inc.
- [10] Lars Powers, Mike Snell, Microsoft Visual Studio 2008 Unleashed, Sams Publishing
- [11] Μαλινδρέτος Γεώργιος, Διπλωματική εργασία, Αθήνα 1996
- [12] Μαρούκης Νικόλαος, Διπλωματική εργασία
- [13] CGAL User and Reference Manual: All Parts, August 2007
- [14] DXF reference, Autodesk, 2006

- [15] <http://www.codeproject.com/KB/recipes/Wykobi.aspx>
- [16] <http://traec.osgeo.org/geos/>
- [17] <http://www.cgal.org/>
- [18] <http://www.ceometric.com>
- [19] http://datastructureebook.blogspot.com/2007_04_01_archive.html
- [20] <http://mathworld.wolfram.com/topics/Geometry.html>
- [21] <http://www.mathacademy.com/pr/prime/>
- [22] <http://msdn.microsoft.com/en-us/library/bb384865.aspx>
- [23] <http://www.math.grin.edu/~rebelsky/Courses/152/97F/Readings/student-Binary>
- [24] <http://www.ecocrete.gr/index.php?option=content&task=view&id=6167>
- [25] Edward G. Blankenship, THE AIRPORT, Pall Mall Press, London
- [26] Υπουργική Απόφαση Δ3/Α/20357/4641/2002 «Έγκριση Βασικού Κανονισμού Αδειοδότησης και Λειτουργίας – Εκμετάλλευσης Αεροδρομίων». (ΦΕΚ Β΄ 701/10.6.2002)

ΠΑΡΑΡΤΗΜΑ Α

ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

1

#pragma once

#include "myData.h"

#include "cstring"

namespace rPORT {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::Runtime::InteropServices; //for the use of Marshal in the {save} handler

/// <summary>

/// Summary for Form1

///

/// WARNING: If you change the name of this class, you will need to change the
 /// 'Resource File Name' property for the managed resource compiler tool
 /// associated with all .resx files this class depends on. Otherwise,
 /// the designers will not be able to interact properly with localized
 /// resources associated with this form.

/// </summary>

public ref class Form1 : public System::Windows::Forms::Form

{

public:

Form1(void)

{

InitializeComponent();

//TODO: Add the constructor code here

aScout.z = 999999;

aScout180.z = 999999 ;

bScout.z = 999999 ;

bScout180.z = 999999 ;

NikosAthanasίου = 999999;

}

protected:

/// <summary>

/// Clean up any resources being used.

/// </summary>

~Form1()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::MenuStrip^ menuStrip1;

protected:

private: System::Windows::Forms::ToolStripMenuItem^ fileToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ newToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ openToolStripMenuItem;

private: System::Windows::Forms::ToolStripSeparator^ toolStripSeparator;

private: System::Windows::Forms::ToolStripMenuItem^ saveToolStripMenuItem;


```
private: System::Windows::Forms::ToolStripMenuItem^ helpToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ contentsToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ indexToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ searchToolStripMenuItem;
private: System::Windows::Forms::ToolStripSeparator^ toolStripSeparator5;
private: System::Windows::Forms::ToolStripMenuItem^ aboutToolStripMenuItem;
```

```
private: System::Windows::Forms::TabPage^ tabPage2;
```

```
private: System::Windows::Forms::GroupBox^ groupBox3;
private: System::Windows::Forms::Label^ label43;
private: System::Windows::Forms::TextBox^ textBox47;
private: System::Windows::Forms::TextBox^ textBox46;
private: System::Windows::Forms::Label^ label39;
private: System::Windows::Forms::TextBox^ textBox45;
private: System::Windows::Forms::TextBox^ textBox44;
private: System::Windows::Forms::Label^ label38;
private: System::Windows::Forms::TextBox^ textBox39;
private: System::Windows::Forms::TextBox^ textBox38;
private: System::Windows::Forms::TextBox^ textBox37;
private: System::Windows::Forms::TextBox^ textBox36;
private: System::Windows::Forms::TextBox^ textBox35;
private: System::Windows::Forms::TextBox^ textBox34;
private: System::Windows::Forms::TextBox^ textBox33;
private: System::Windows::Forms::Label^ label35;
private: System::Windows::Forms::Label^ label34;
private: System::Windows::Forms::Label^ label33;
private: System::Windows::Forms::Label^ label32;
private: System::Windows::Forms::Label^ label31;
private: System::Windows::Forms::Label^ label30;
private: System::Windows::Forms::Label^ label29;
private: System::Windows::Forms::Label^ label27;
```

```
private: System::Windows::Forms::TabPage^ tabPage1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::GroupBox^ groupBox1;
```

```
private: System::Windows::Forms::Button^ button8;
```

```
private: System::Windows::Forms::Button^ button7;
```

```
private: System::Windows::Forms::Button^ button3;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::TextBox^ textBox7;
private: System::Windows::Forms::Label^ label11;
private: System::Windows::Forms::TextBox^ textBox6;
private: System::Windows::Forms::Label^ label10;
private: System::Windows::Forms::TextBox^ textBox5;
private: System::Windows::Forms::TextBox^ textBox4;
private: System::Windows::Forms::TextBox^ textBox3;
private: System::Windows::Forms::Label^ label9;
private: System::Windows::Forms::Label^ label8;
private: System::Windows::Forms::Label^ label7;
private: System::Windows::Forms::Label^ label6;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::ComboBox^ comboBox2;
private: System::Windows::Forms::ComboBox^ comboBox1;
private: System::Windows::Forms::GroupBox^ groupBox2;
```

```
private: System::Windows::Forms::Button^ button10;
```

```
private: System::Windows::Forms::Button^ button9;
```

```
private: System::Windows::Forms::Button^ button6;
private: System::Windows::Forms::Button^ button5;
private: System::Windows::Forms::Button^ button4;
private: System::Windows::Forms::Label^ label22;
private: System::Windows::Forms::Label^ label21;
```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

4

```
private: System::Windows::Forms::TextBox^ textBox12;
private: System::Windows::Forms::TextBox^ textBox11;
private: System::Windows::Forms::TextBox^ textBox10;
private: System::Windows::Forms::TextBox^ textBox9;
private: System::Windows::Forms::TextBox^ textBox8;
private: System::Windows::Forms::Label^ label20;
private: System::Windows::Forms::Label^ label19;
private: System::Windows::Forms::Label^ label18;
private: System::Windows::Forms::Label^ label17;
private: System::Windows::Forms::ComboBox^ comboBox4;
private: System::Windows::Forms::Label^ label16;
private: System::Windows::Forms::ComboBox^ comboBox3;
private: System::Windows::Forms::Label^ label15;
private: System::Windows::Forms::TabControl^ tabControl1;
private: System::Windows::Forms::Label^ label44;
private: System::Windows::Forms::GroupBox^ groupBox5;
private: System::Windows::Forms::Label^ label45;
private: System::Windows::Forms::Label^ label46;
private: System::Windows::Forms::TextBox^ textBox40;
private: System::Windows::Forms::TextBox^ textBox41;
private: System::Windows::Forms::Label^ label47;
private: System::Windows::Forms::TextBox^ textBox42;
private: System::Windows::Forms::TextBox^ textBox43;
private: System::Windows::Forms::Label^ label48;
private: System::Windows::Forms::TextBox^ textBox68;
private: System::Windows::Forms::TextBox^ textBox69;
private: System::Windows::Forms::TextBox^ textBox70;
private: System::Windows::Forms::TextBox^ textBox71;
private: System::Windows::Forms::TextBox^ textBox72;
private: System::Windows::Forms::TextBox^ textBox73;
private: System::Windows::Forms::TextBox^ textBox74;
private: System::Windows::Forms::Label^ label49;
private: System::Windows::Forms::Label^ label50;
private: System::Windows::Forms::Label^ label51;
private: System::Windows::Forms::Label^ label52;
private: System::Windows::Forms::Label^ label53;
private: System::Windows::Forms::Label^ label54;
private: System::Windows::Forms::Label^ label55;
private: System::Windows::Forms::Label^ label56;
```

```
private: System::Windows::Forms::SaveFileDialog^ saveFileDialog1;
private: System::Windows::Forms::Button^ button11;
private: System::Windows::Forms::TextBox^ textBox159;
private: System::Windows::Forms::ToolStrip^ toolStrip1;
private: System::Windows::Forms::ToolStripButton^ newToolStripButton;
private: System::Windows::Forms::ToolStripButton^ openToolStripButton;
private: System::Windows::Forms::ToolStripButton^ saveToolStripButton;

private: System::Windows::Forms::ToolStripSeparator^ toolStripSeparator4;
private: System::Windows::Forms::ToolStripButton^ toolStripButton1;
private: System::Windows::Forms::ToolStripMenuItem^ exportAutoCadDataToolStripMenuItem;
private: System::Windows::Forms::ToolTip^ toolTip1;
private: System::Windows::Forms::ToolStripSeparator^ toolStripSeparator1;
private: System::Windows::Forms::TabPage^ tabPage4;
```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

6

```
private: System::Windows::Forms::TabPage^ tabPage5;
private: System::Windows::Forms::TabPage^ tabPage6;
private: System::Windows::Forms::TabPage^ tabPage7;
private: System::Windows::Forms::GroupBox^ groupBox4;
private: System::Windows::Forms::TextBox^ textBox31;
private: System::Windows::Forms::TextBox^ textBox65;
private: System::Windows::Forms::TextBox^ textBox30;
private: System::Windows::Forms::TextBox^ textBox66;
private: System::Windows::Forms::GroupBox^ groupBox6;
private: System::Windows::Forms::TextBox^ textBox58;
private: System::Windows::Forms::TextBox^ textBox57;
private: System::Windows::Forms::TextBox^ textBox23;
private: System::Windows::Forms::TextBox^ textBox22;
private: System::Windows::Forms::GroupBox^ groupBox7;
private: System::Windows::Forms::TextBox^ textBox15;
private: System::Windows::Forms::TextBox^ textBox49;
private: System::Windows::Forms::TextBox^ textBox14;
private: System::Windows::Forms::TextBox^ textBox48;
private: System::Windows::Forms::TextBox^ textBox13;
private: System::Windows::Forms::TextBox^ textBox51;
private: System::Windows::Forms::TextBox^ textBox50;
private: System::Windows::Forms::TextBox^ textBox20;
private: System::Windows::Forms::TextBox^ textBox19;
private: System::Windows::Forms::TextBox^ textBox18;
private: System::Windows::Forms::TextBox^ textBox17;
private: System::Windows::Forms::TextBox^ textBox16;
private: System::Windows::Forms::TextBox^ textBox21;
private: System::Windows::Forms::TextBox^ textBox56;
private: System::Windows::Forms::TextBox^ textBox55;
private: System::Windows::Forms::TextBox^ textBox54;
private: System::Windows::Forms::TextBox^ textBox53;
private: System::Windows::Forms::TextBox^ textBox52;
private: System::Windows::Forms::GroupBox^ groupBox8;
private: System::Windows::Forms::TextBox^ textBox29;
private: System::Windows::Forms::TextBox^ textBox28;
private: System::Windows::Forms::TextBox^ textBox27;
private: System::Windows::Forms::TextBox^ textBox26;
private: System::Windows::Forms::TextBox^ textBox25;
private: System::Windows::Forms::TextBox^ textBox24;
private: System::Windows::Forms::TextBox^ textBox32;
private: System::Windows::Forms::TextBox^ textBox60;
private: System::Windows::Forms::TextBox^ textBox59;
private: System::Windows::Forms::TextBox^ textBox64;
private: System::Windows::Forms::TextBox^ textBox63;
private: System::Windows::Forms::TextBox^ textBox62;
private: System::Windows::Forms::TextBox^ textBox61;
private: System::Windows::Forms::TextBox^ textBox67;
private: System::Windows::Forms::SaveFileDialog^ saveFileDialog2;

private: System::ComponentModel::IContainer^ components;

//private: System::Windows::Forms::ToolStripMenuItem^ newToolStripMenuItem;
```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

7

```

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        this->components = (gcnew System::ComponentModel::Container());
        System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::
ComponentModel::ComponentResourceManager(Form1::typeid));
        this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
        this->fileToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
        this->newToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
        this->openToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
        this->toolStripSeparator = (gcnew System::Windows::Forms::ToolStripSeparator());
        this->saveToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
        this->toolStripSeparator1 = (gcnew System::Windows::Forms::ToolStripSeparator());
        this->exportAutoCadDataToolStripMenuItem = (gcnew System::Windows::Forms::
ToolStripMenuItem());
        this->helpToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());
        this->contentsToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem
    ());
        this->indexToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem())
    ;
        this->searchToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem
    ());
        this->toolStripSeparator5 = (gcnew System::Windows::Forms::ToolStripSeparator());
        this->aboutToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem())
    ;

        this->tabPage2 = (gcnew System::Windows::Forms::TabPage());
        this->textBox159 = (gcnew System::Windows::Forms::TextBox());
        this->button11 = (gcnew System::Windows::Forms::Button());
        this->groupBox5 = (gcnew System::Windows::Forms::GroupBox());
        this->label145 = (gcnew System::Windows::Forms::Label());
        this->label146 = (gcnew System::Windows::Forms::Label());
        this->textBox40 = (gcnew System::Windows::Forms::TextBox());
        this->textBox41 = (gcnew System::Windows::Forms::TextBox());
        this->label147 = (gcnew System::Windows::Forms::Label());
        this->textBox42 = (gcnew System::Windows::Forms::TextBox());
        this->textBox43 = (gcnew System::Windows::Forms::TextBox());
        this->label148 = (gcnew System::Windows::Forms::Label());
        this->textBox68 = (gcnew System::Windows::Forms::TextBox());
        this->textBox69 = (gcnew System::Windows::Forms::TextBox());
        this->textBox70 = (gcnew System::Windows::Forms::TextBox());
        this->textBox71 = (gcnew System::Windows::Forms::TextBox());
        this->textBox72 = (gcnew System::Windows::Forms::TextBox());
        this->textBox73 = (gcnew System::Windows::Forms::TextBox());
        this->textBox74 = (gcnew System::Windows::Forms::TextBox());
        this->label149 = (gcnew System::Windows::Forms::Label());
        this->label150 = (gcnew System::Windows::Forms::Label());
        this->label151 = (gcnew System::Windows::Forms::Label());
        this->label152 = (gcnew System::Windows::Forms::Label());
        this->label153 = (gcnew System::Windows::Forms::Label());
        this->label154 = (gcnew System::Windows::Forms::Label());
        this->label155 = (gcnew System::Windows::Forms::Label());
        this->label156 = (gcnew System::Windows::Forms::Label());
        this->groupBox3 = (gcnew System::Windows::Forms::GroupBox());
        this->label144 = (gcnew System::Windows::Forms::Label());
        this->label143 = (gcnew System::Windows::Forms::Label());
        this->textBox47 = (gcnew System::Windows::Forms::TextBox());
        this->textBox46 = (gcnew System::Windows::Forms::TextBox());

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

8

```

this->label39 = (gcnew System::Windows::Forms::Label());
this->textBox45 = (gcnew System::Windows::Forms::TextBox());
this->textBox44 = (gcnew System::Windows::Forms::TextBox());
this->label38 = (gcnew System::Windows::Forms::Label());
this->textBox39 = (gcnew System::Windows::Forms::TextBox());
this->textBox38 = (gcnew System::Windows::Forms::TextBox());
this->textBox37 = (gcnew System::Windows::Forms::TextBox());
this->textBox36 = (gcnew System::Windows::Forms::TextBox());
this->textBox35 = (gcnew System::Windows::Forms::TextBox());
this->textBox34 = (gcnew System::Windows::Forms::TextBox());
this->textBox33 = (gcnew System::Windows::Forms::TextBox());
this->label35 = (gcnew System::Windows::Forms::Label());
this->label34 = (gcnew System::Windows::Forms::Label());
this->label33 = (gcnew System::Windows::Forms::Label());
this->label32 = (gcnew System::Windows::Forms::Label());
this->label31 = (gcnew System::Windows::Forms::Label());
this->label30 = (gcnew System::Windows::Forms::Label());
this->label29 = (gcnew System::Windows::Forms::Label());
this->label27 = (gcnew System::Windows::Forms::Label());
this->tabPage1 = (gcnew System::Windows::Forms::TabPage());
this->textBox2 = (gcnew System::Windows::Forms::TextBox());
this->textBox1 = (gcnew System::Windows::Forms::TextBox());
this->label3 = (gcnew System::Windows::Forms::Label());
this->label2 = (gcnew System::Windows::Forms::Label());
this->label1 = (gcnew System::Windows::Forms::Label());
this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
this->button8 = (gcnew System::Windows::Forms::Button());
this->button7 = (gcnew System::Windows::Forms::Button());
this->button3 = (gcnew System::Windows::Forms::Button());
this->button2 = (gcnew System::Windows::Forms::Button());
this->button1 = (gcnew System::Windows::Forms::Button());
this->textBox7 = (gcnew System::Windows::Forms::TextBox());
this->label11 = (gcnew System::Windows::Forms::Label());
this->textBox6 = (gcnew System::Windows::Forms::TextBox());
this->label10 = (gcnew System::Windows::Forms::Label());
this->textBox5 = (gcnew System::Windows::Forms::TextBox());
this->textBox4 = (gcnew System::Windows::Forms::TextBox());
this->textBox3 = (gcnew System::Windows::Forms::TextBox());
this->label9 = (gcnew System::Windows::Forms::Label());
this->label8 = (gcnew System::Windows::Forms::Label());
this->label7 = (gcnew System::Windows::Forms::Label());
this->label6 = (gcnew System::Windows::Forms::Label());
this->label5 = (gcnew System::Windows::Forms::Label());
this->label4 = (gcnew System::Windows::Forms::Label());
this->comboBox2 = (gcnew System::Windows::Forms::ComboBox());
this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());
this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());
this->button10 = (gcnew System::Windows::Forms::Button());
this->button9 = (gcnew System::Windows::Forms::Button());
this->button6 = (gcnew System::Windows::Forms::Button());
this->button5 = (gcnew System::Windows::Forms::Button());
this->button4 = (gcnew System::Windows::Forms::Button());
this->label22 = (gcnew System::Windows::Forms::Label());
this->label21 = (gcnew System::Windows::Forms::Label());
this->textBox12 = (gcnew System::Windows::Forms::TextBox());
this->textBox11 = (gcnew System::Windows::Forms::TextBox());
this->textBox10 = (gcnew System::Windows::Forms::TextBox());
this->textBox9 = (gcnew System::Windows::Forms::TextBox());
this->textBox8 = (gcnew System::Windows::Forms::TextBox());
this->label20 = (gcnew System::Windows::Forms::Label());
this->label19 = (gcnew System::Windows::Forms::Label());
this->label18 = (gcnew System::Windows::Forms::Label());
this->label17 = (gcnew System::Windows::Forms::Label());
this->comboBox4 = (gcnew System::Windows::Forms::ComboBox());
this->label16 = (gcnew System::Windows::Forms::Label());
this->comboBox3 = (gcnew System::Windows::Forms::ComboBox());
this->label15 = (gcnew System::Windows::Forms::Label());
this->tabControl1 = (gcnew System::Windows::Forms::TabControl());
this->tabPage4 = (gcnew System::Windows::Forms::TabPage());

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

9

```

this->groupBox4 = (gcnew System::Windows::Forms::GroupBox());
this->textBox67 = (gcnew System::Windows::Forms::TextBox());
this->textBox32 = (gcnew System::Windows::Forms::TextBox());
this->textBox65 = (gcnew System::Windows::Forms::TextBox());
this->textBox30 = (gcnew System::Windows::Forms::TextBox());
this->textBox66 = (gcnew System::Windows::Forms::TextBox());
this->textBox31 = (gcnew System::Windows::Forms::TextBox());
this->tabPage5 = (gcnew System::Windows::Forms::TabPage());
this->groupBox6 = (gcnew System::Windows::Forms::GroupBox());
this->textBox58 = (gcnew System::Windows::Forms::TextBox());
this->textBox57 = (gcnew System::Windows::Forms::TextBox());
this->textBox23 = (gcnew System::Windows::Forms::TextBox());
this->textBox22 = (gcnew System::Windows::Forms::TextBox());
this->tabPage6 = (gcnew System::Windows::Forms::TabPage());
this->groupBox7 = (gcnew System::Windows::Forms::GroupBox());
this->textBox21 = (gcnew System::Windows::Forms::TextBox());
this->textBox56 = (gcnew System::Windows::Forms::TextBox());
this->textBox55 = (gcnew System::Windows::Forms::TextBox());
this->textBox54 = (gcnew System::Windows::Forms::TextBox());
this->textBox53 = (gcnew System::Windows::Forms::TextBox());
this->textBox52 = (gcnew System::Windows::Forms::TextBox());
this->textBox51 = (gcnew System::Windows::Forms::TextBox());
this->textBox50 = (gcnew System::Windows::Forms::TextBox());
this->textBox20 = (gcnew System::Windows::Forms::TextBox());
this->textBox19 = (gcnew System::Windows::Forms::TextBox());
this->textBox18 = (gcnew System::Windows::Forms::TextBox());
this->textBox17 = (gcnew System::Windows::Forms::TextBox());
this->textBox16 = (gcnew System::Windows::Forms::TextBox());
this->textBox15 = (gcnew System::Windows::Forms::TextBox());
this->textBox49 = (gcnew System::Windows::Forms::TextBox());
this->textBox14 = (gcnew System::Windows::Forms::TextBox());
this->textBox48 = (gcnew System::Windows::Forms::TextBox());
this->textBox13 = (gcnew System::Windows::Forms::TextBox());
this->tabPage7 = (gcnew System::Windows::Forms::TabPage());
this->groupBox8 = (gcnew System::Windows::Forms::GroupBox());
this->textBox60 = (gcnew System::Windows::Forms::TextBox());
this->textBox59 = (gcnew System::Windows::Forms::TextBox());
this->textBox64 = (gcnew System::Windows::Forms::TextBox());
this->textBox63 = (gcnew System::Windows::Forms::TextBox());
this->textBox62 = (gcnew System::Windows::Forms::TextBox());
this->textBox61 = (gcnew System::Windows::Forms::TextBox());
this->textBox29 = (gcnew System::Windows::Forms::TextBox());
this->textBox28 = (gcnew System::Windows::Forms::TextBox());
this->textBox27 = (gcnew System::Windows::Forms::TextBox());
this->textBox26 = (gcnew System::Windows::Forms::TextBox());
this->textBox25 = (gcnew System::Windows::Forms::TextBox());
this->textBox24 = (gcnew System::Windows::Forms::TextBox());
this->saveFileDialog1 = (gcnew System::Windows::Forms::SaveFileDialog());
this->toolStrip1 = (gcnew System::Windows::Forms::ToolStrip());
this->newToolStripButton = (gcnew System::Windows::Forms::ToolStripButton());
this->openToolStripButton = (gcnew System::Windows::Forms::ToolStripButton());
this->saveToolStripButton = (gcnew System::Windows::Forms::ToolStripButton());
this->toolStripSeparator4 = (gcnew System::Windows::Forms::ToolStripSeparator());
this->toolStripButton1 = (gcnew System::Windows::Forms::ToolStripButton());
this->toolTip1 = (gcnew System::Windows::Forms::ToolTip(this->components));
this->saveFileDialog2 = (gcnew System::Windows::Forms::SaveFileDialog());
this->menuStrip1->SuspendLayout();
this->tabPage2->SuspendLayout();
this->groupBox5->SuspendLayout();
this->groupBox3->SuspendLayout();
this->tabPage1->SuspendLayout();
this->groupBox1->SuspendLayout();
this->groupBox2->SuspendLayout();
this->tabControl1->SuspendLayout();
this->tabPage4->SuspendLayout();
this->groupBox4->SuspendLayout();
this->tabPage5->SuspendLayout();
this->groupBox6->SuspendLayout();
this->tabPage6->SuspendLayout();

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

10

```

        this->groupBox7->SuspendLayout();
        this->tabPage7->SuspendLayout();
        this->groupBox8->SuspendLayout();
        this->toolStrip1->SuspendLayout();
        this->SuspendLayout();
        //
        // menuStrip1
        //
        this->menuStrip1->BackColor = System::Drawing::SystemColors::Control;
        this->menuStrip1->BackgroundImageLayout = System::Windows::Forms::ImageLayout::
Zoom;
        this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::
ToolStripItem^ >(2) {this->fileToolStripMenuItem,
        this->helpToolStripMenuItem});
        this->menuStrip1->Location = System::Drawing::Point(0, 0);
        this->menuStrip1->Name = L"menuStrip1";
        this->menuStrip1->Size = System::Drawing::Size(944, 24);
        this->menuStrip1->TabIndex = 0;
        this->menuStrip1->Text = L"menuStrip1";
        //
        // fileToolStripMenuItem
        //
        this->fileToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::
Windows::Forms::ToolStripItem^ >(6) {this->newToolStripMenuItem,
        this->openToolStripMenuItem, this->toolStripSeparator, this->
saveToolStripMenuItem, this->toolStripSeparator1, this->
exportAutoCadDataToolStripMenuItem});
        this->fileToolStripMenuItem->Name = L"fileToolStripMenuItem";
        this->fileToolStripMenuItem->Size = System::Drawing::Size(35, 20);
        this->fileToolStripMenuItem->Text = L"&File";
        //
        // newToolStripMenuItem
        //
        this->newToolStripMenuItem->Image = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"newToolStripMenuItem.Image")));
        this->newToolStripMenuItem->ImageTransparentColor = System::Drawing::Color::
Magenta;
        this->newToolStripMenuItem->Name = L"newToolStripMenuItem";
        this->newToolStripMenuItem->ShortcutKeys = static_cast<System::Windows::Forms::
Keys>((System::Windows::Forms::Keys::Control | System::Windows::Forms::Keys::N));
        this->newToolStripMenuItem->Size = System::Drawing::Size(225, 22);
        this->newToolStripMenuItem->Text = L"&New";
        this->newToolStripMenuItem->Click += gcnew System::EventHandler(this, &Form1::
newToolStripMenuItem_Click);
        //
        // openToolStripMenuItem
        //
        this->openToolStripMenuItem->Image = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"openToolStripMenuItem.Image")));
        this->openToolStripMenuItem->ImageTransparentColor = System::Drawing::Color::
Magenta;
        this->openToolStripMenuItem->Name = L"openToolStripMenuItem";
        this->openToolStripMenuItem->ShortcutKeys = static_cast<System::Windows::Forms::
Keys>((System::Windows::Forms::Keys::Control | System::Windows::Forms::Keys::O));
        this->openToolStripMenuItem->Size = System::Drawing::Size(225, 22);
        this->openToolStripMenuItem->Text = L"&Open";
        this->openToolStripMenuItem->Click += gcnew System::EventHandler(this, &Form1::
openToolStripMenuItem_Click);
        //
        // toolStripSeparator
        //
        this->toolStripSeparator->Name = L"toolStripSeparator";
        this->toolStripSeparator->Size = System::Drawing::Size(222, 6);
        //
        // saveToolStripMenuItem
        //
        this->saveToolStripMenuItem->Image = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"saveToolStripMenuItem.Image")));
        this->saveToolStripMenuItem->ImageTransparentColor = System::Drawing::Color::

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

11

```

Magenta;
    this->saveToolStripMenuItem->Name = L"saveToolStripMenuItem";
    this->saveToolStripMenuItem->ShortcutKeys = static_cast<System::Windows::Forms::
Keys>((System::Windows::Forms::Keys::Control | System::Windows::Forms::Keys::S));
    this->saveToolStripMenuItem->Size = System::Drawing::Size(225, 22);
    this->saveToolStripMenuItem->Text = L"&Export Printable Text";
    this->saveToolStripMenuItem->Click += gcnew System::EventHandler(this, &Form1::
saveToolStripMenuItem_Click);
    //
    // toolStripSeparator1
    //
    this->toolStripSeparator1->Name = L"toolStripSeparator1";
    this->toolStripSeparator1->Size = System::Drawing::Size(222, 6);
    //
    // exportAutoCadDataToolStripMenuItem
    //
    this->exportAutoCadDataToolStripMenuItem->Image = (cli::safe_cast<System::Drawing::
:Image^ >(resources->GetObject(L"exportAutoCadDataToolStripMenuItem.Image")));
    this->exportAutoCadDataToolStripMenuItem->Name = L
"exportAutoCadDataToolStripMenuItem";
    this->exportAutoCadDataToolStripMenuItem->Size = System::Drawing::Size(225, 22);
    this->exportAutoCadDataToolStripMenuItem->Text = L"Export AutoCad Data";
    this->exportAutoCadDataToolStripMenuItem->Click += gcnew System::EventHandler(this
, &Form1::exportAutoCadDataToolStripMenuItem_Click);
    //
    // helpToolStripMenuItem
    //
    this->helpToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::
Windows::Forms::ToolStripItem^ >(5) {this->contentsToolStripMenuItem,
    this->indexToolStripMenuItem, this->searchToolStripMenuItem, this->
toolStripSeparator5, this->aboutToolStripMenuItem});
    this->helpToolStripMenuItem->Name = L"helpToolStripMenuItem";
    this->helpToolStripMenuItem->Size = System::Drawing::Size(40, 20);
    this->helpToolStripMenuItem->Text = L"&Help";
    //
    // contentsToolStripMenuItem
    //
    this->contentsToolStripMenuItem->Name = L"contentsToolStripMenuItem";
    this->contentsToolStripMenuItem->Size = System::Drawing::Size(129, 22);
    this->contentsToolStripMenuItem->Text = L"&Contents";
    //
    // indexToolStripMenuItem
    //
    this->indexToolStripMenuItem->Name = L"indexToolStripMenuItem";
    this->indexToolStripMenuItem->Size = System::Drawing::Size(129, 22);
    this->indexToolStripMenuItem->Text = L"&Index";
    //
    // searchToolStripMenuItem
    //
    this->searchToolStripMenuItem->Name = L"searchToolStripMenuItem";
    this->searchToolStripMenuItem->Size = System::Drawing::Size(129, 22);
    this->searchToolStripMenuItem->Text = L"&Search";
    //
    // toolStripSeparator5
    //
    this->toolStripSeparator5->Name = L"toolStripSeparator5";
    this->toolStripSeparator5->Size = System::Drawing::Size(126, 6);
    //
    // aboutToolStripMenuItem
    //
    this->aboutToolStripMenuItem->Name = L"aboutToolStripMenuItem";
    this->aboutToolStripMenuItem->Size = System::Drawing::Size(129, 22);
    this->aboutToolStripMenuItem->Text = L"&About...";
    //
    // tabPage2
    //
    this->tabPage2->BackColor = System::Drawing::SystemColors::ControlLightLight;
    this->tabPage2->BackgroundImageLayout = System::Windows::Forms::ImageLayout::
Stretch;

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

12

```

this->tabPage2->Controls->Add(this->textBox159);
this->tabPage2->Controls->Add(this->button11);
this->tabPage2->Controls->Add(this->groupBox5);
this->tabPage2->Controls->Add(this->groupBox3);
this->tabPage2->Location = System::Drawing::Point(4, 22);
this->tabPage2->Name = L"tabPage2";
this->tabPage2->Padding = System::Windows::Forms::Padding(3);
this->tabPage2->Size = System::Drawing::Size(910, 547);
this->tabPage2->TabIndex = 3;
this->tabPage2->Text = L"Results";
//
// textBox159
//
this->textBox159->BackColor = System::Drawing::Color::Red;
this->textBox159->ForeColor = System::Drawing::SystemColors::Window;
this->textBox159->Location = System::Drawing::Point(453, 448);
this->textBox159->Name = L"textBox159";
this->textBox159->Size = System::Drawing::Size(100, 20);
this->textBox159->TabIndex = 4;
this->textBox159->Text = L"NOT SET YET";
this->textBox159->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// button11
//
this->button11->Location = System::Drawing::Point(348, 446);
this->button11->Name = L"button11";
this->button11->Size = System::Drawing::Size(99, 23);
this->button11->TabIndex = 3;
this->button11->Text = L"FINAL HEIGHT = ";
this->button11->UseVisualStyleBackColor = true;
this->button11->Click += gcnew System::EventHandler(this, &Form1::button11_Click);
//
// groupBox5
//
this->groupBox5->BackColor = System::Drawing::SystemColors::ControlLightLight;
this->groupBox5->BackgroundImageLayout = System::Windows::Forms::ImageLayout::
Stretch;
this->groupBox5->Controls->Add(this->label45);
this->groupBox5->Controls->Add(this->label46);
this->groupBox5->Controls->Add(this->textBox40);
this->groupBox5->Controls->Add(this->textBox41);
this->groupBox5->Controls->Add(this->label47);
this->groupBox5->Controls->Add(this->textBox42);
this->groupBox5->Controls->Add(this->textBox43);
this->groupBox5->Controls->Add(this->label48);
this->groupBox5->Controls->Add(this->textBox68);
this->groupBox5->Controls->Add(this->textBox69);
this->groupBox5->Controls->Add(this->textBox70);
this->groupBox5->Controls->Add(this->textBox71);
this->groupBox5->Controls->Add(this->textBox72);
this->groupBox5->Controls->Add(this->textBox73);
this->groupBox5->Controls->Add(this->textBox74);
this->groupBox5->Controls->Add(this->label49);
this->groupBox5->Controls->Add(this->label50);
this->groupBox5->Controls->Add(this->label51);
this->groupBox5->Controls->Add(this->label52);
this->groupBox5->Controls->Add(this->label53);
this->groupBox5->Controls->Add(this->label54);
this->groupBox5->Controls->Add(this->label55);
this->groupBox5->Controls->Add(this->label56);
this->groupBox5->Location = System::Drawing::Point(460, 56);
this->groupBox5->Name = L"groupBox5";
this->groupBox5->Size = System::Drawing::Size(447, 330);
this->groupBox5->TabIndex = 2;
this->groupBox5->TabStop = false;
this->groupBox5->Text = L"RUNWAY 2";
//
// label45
//

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

13

```

this->label45->AutoSize = true;
this->label45->BackColor = System::Drawing::Color::White;
this->label45->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->label45->ForeColor = System::Drawing::SystemColors::ControlText;
this->label45->Location = System::Drawing::Point(169, 301);
this->label45->Name = L"label45";
this->label45->Size = System::Drawing::Size(29, 13);
this->label45->TabIndex = 19;
this->label45->Text = L"THE";
//
// label46
//
this->label46->AutoSize = true;
this->label46->Location = System::Drawing::Point(377, 288);
this->label46->Name = L"label46";
this->label46->Size = System::Drawing::Size(57, 13);
this->label46->TabIndex = 5;
this->label46->Text = L"SURFACE";
//
// textBox40
//
this->textBox40->Location = System::Drawing::Point(234, 304);
this->textBox40->Name = L"textBox40";
this->textBox40->Size = System::Drawing::Size(137, 20);
this->textBox40->TabIndex = 20;
this->textBox40->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox41
//
this->textBox41->Location = System::Drawing::Point(234, 278);
this->textBox41->Name = L"textBox41";
this->textBox41->Size = System::Drawing::Size(137, 20);
this->textBox41->TabIndex = 19;
this->textBox41->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// label47
//
this->label47->AutoSize = true;
this->label47->BackColor = System::Drawing::Color::White;
this->label47->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->label47->ForeColor = System::Drawing::SystemColors::ControlText;
this->label47->Location = System::Drawing::Point(157, 288);
this->label47->Name = L"label47";
this->label47->Size = System::Drawing::Size(71, 13);
this->label47->TabIndex = 18;
this->label47->Text = L"DEFINED BY";
//
// textBox42
//
this->textBox42->BackColor = System::Drawing::Color::Blue;
this->textBox42->ForeColor = System::Drawing::Color::White;
this->textBox42->Location = System::Drawing::Point(84, 304);
this->textBox42->Name = L"textBox42";
this->textBox42->Size = System::Drawing::Size(67, 20);
this->textBox42->TabIndex = 17;
this->textBox42->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox43
//
this->textBox43->BackColor = System::Drawing::Color::Blue;
this->textBox43->ForeColor = System::Drawing::Color::White;
this->textBox43->Location = System::Drawing::Point(84, 278);
this->textBox43->Name = L"textBox43";
this->textBox43->Size = System::Drawing::Size(67, 20);
this->textBox43->TabIndex = 16;

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

14

```

this->textBox43->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// label48
//
this->label48->AutoSize = true;
this->label48->Location = System::Drawing::Point(6, 288);
this->label48->Name = L"label48";
this->label48->Size = System::Drawing::Size(72, 26);
this->label48->TabIndex = 15;
this->label48->Text = L"ALLOWABLE\r\n HEIGHT IS \r\n";
//
// textBox68
//
this->textBox68->Location = System::Drawing::Point(327, 111);
this->textBox68->Name = L"textBox68";
this->textBox68->Size = System::Drawing::Size(100, 20);
this->textBox68->TabIndex = 14;
this->textBox68->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox69
//
this->textBox69->Location = System::Drawing::Point(187, 111);
this->textBox69->Name = L"textBox69";
this->textBox69->Size = System::Drawing::Size(100, 20);
this->textBox69->TabIndex = 13;
this->textBox69->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox70
//
this->textBox70->Location = System::Drawing::Point(30, 111);
this->textBox70->Name = L"textBox70";
this->textBox70->Size = System::Drawing::Size(100, 20);
this->textBox70->TabIndex = 12;
this->textBox70->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox71
//
this->textBox71->Location = System::Drawing::Point(327, 60);
this->textBox71->Name = L"textBox71";
this->textBox71->Size = System::Drawing::Size(100, 20);
this->textBox71->TabIndex = 11;
this->textBox71->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox72
//
this->textBox72->Location = System::Drawing::Point(221, 60);
this->textBox72->Name = L"textBox72";
this->textBox72->Size = System::Drawing::Size(100, 20);
this->textBox72->TabIndex = 10;
this->textBox72->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox73
//
this->textBox73->Location = System::Drawing::Point(115, 60);
this->textBox73->Name = L"textBox73";
this->textBox73->Size = System::Drawing::Size(100, 20);
this->textBox73->TabIndex = 9;
this->textBox73->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox74
//
this->textBox74->Location = System::Drawing::Point(9, 60);
this->textBox74->Name = L"textBox74";
this->textBox74->Size = System::Drawing::Size(100, 20);
this->textBox74->TabIndex = 8;
this->textBox74->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// label49
//

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

15

```

this->label49->AutoSize = true;
this->label49->Location = System::Drawing::Point(357, 95);
this->label49->Name = L"label49";
this->label49->Size = System::Drawing::Size(53, 13);
this->label49->TabIndex = 7;
this->label49->Text = L"CONICAL";
//
// label50
//
this->label50->AutoSize = true;
this->label50->BackColor = System::Drawing::Color::White;
this->label50->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->label50->Location = System::Drawing::Point(169, 95);
this->label50->Name = L"label50";
this->label50->Size = System::Drawing::Size(118, 13);
this->label50->TabIndex = 6;
this->label50->Text = L"OUTER HORIZONTAL";
//
// label51
//
this->label51->AutoSize = true;
this->label51->Location = System::Drawing::Point(341, 44);
this->label51->Name = L"label51";
this->label51->Size = System::Drawing::Size(86, 13);
this->label51->TabIndex = 5;
this->label51->Text = L"TRANSITIONAL";
//
// label52
//
this->label52->AutoSize = true;
this->label52->Location = System::Drawing::Point(27, 95);
this->label52->Name = L"label52";
this->label52->Size = System::Drawing::Size(114, 13);
this->label52->TabIndex = 4;
this->label52->Text = L"INNER HORIZONTAL";
//
// label53
//
this->label53->AutoSize = true;
this->label53->Location = System::Drawing::Point(231, 44);
this->label53->Name = L"label53";
this->label53->Size = System::Drawing::Size(81, 13);
this->label53->TabIndex = 3;
this->label53->Text = L"SAFETY ZONE";
//
// label54
//
this->label54->AutoSize = true;
this->label54->Location = System::Drawing::Point(133, 44);
this->label54->Name = L"label54";
this->label54->Size = System::Drawing::Size(58, 13);
this->label54->TabIndex = 2;
this->label54->Text = L"TAKE OFF";
//
// label55
//
this->label55->AutoSize = true;
this->label55->Location = System::Drawing::Point(27, 44);
this->label55->Name = L"label55";
this->label55->Size = System::Drawing::Size(66, 13);
this->label55->TabIndex = 1;
this->label55->Text = L"APPROACH";
//
// label56
//
this->label56->AutoSize = true;
this->label56->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

16

```

System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0));
this->label56->Location = System::Drawing::Point(60, 20);
this->label56->Name = L"label56";
this->label56->Size = System::Drawing::Size(205, 13);
this->label56->TabIndex = 0;
this->label56->Text = L"THE POINT IS IN THE SURFACES";
//
// groupBox3
//
this->groupBox3->BackColor = System::Drawing::SystemColors::ControlLightLight;
this->groupBox3->BackgroundImageLayout = System::Windows::Forms::ImageLayout::
Stretch;
this->groupBox3->Controls->Add(this->label44);
this->groupBox3->Controls->Add(this->label43);
this->groupBox3->Controls->Add(this->textBox47);
this->groupBox3->Controls->Add(this->textBox46);
this->groupBox3->Controls->Add(this->label39);
this->groupBox3->Controls->Add(this->textBox45);
this->groupBox3->Controls->Add(this->textBox44);
this->groupBox3->Controls->Add(this->label38);
this->groupBox3->Controls->Add(this->textBox39);
this->groupBox3->Controls->Add(this->textBox38);
this->groupBox3->Controls->Add(this->textBox37);
this->groupBox3->Controls->Add(this->textBox36);
this->groupBox3->Controls->Add(this->textBox35);
this->groupBox3->Controls->Add(this->textBox34);
this->groupBox3->Controls->Add(this->textBox33);
this->groupBox3->Controls->Add(this->label35);
this->groupBox3->Controls->Add(this->label34);
this->groupBox3->Controls->Add(this->label33);
this->groupBox3->Controls->Add(this->label32);
this->groupBox3->Controls->Add(this->label31);
this->groupBox3->Controls->Add(this->label30);
this->groupBox3->Controls->Add(this->label29);
this->groupBox3->Controls->Add(this->label27);
this->groupBox3->Location = System::Drawing::Point(6, 56);
this->groupBox3->Name = L"groupBox3";
this->groupBox3->Size = System::Drawing::Size(447, 330);
this->groupBox3->TabIndex = 0;
this->groupBox3->TabStop = false;
this->groupBox3->Text = L"RUNWAY 1";
//
// label44
//
this->label44->AutoSize = true;
this->label44->BackColor = System::Drawing::Color::White;
this->label44->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0));
this->label44->ForeColor = System::Drawing::SystemColors::ControlText;
this->label44->Location = System::Drawing::Point(169, 301);
this->label44->Name = L"label44";
this->label44->Size = System::Drawing::Size(29, 13);
this->label44->TabIndex = 19;
this->label44->Text = L"THE";
//
// label43
//
this->label43->AutoSize = true;
this->label43->Location = System::Drawing::Point(377, 288);
this->label43->Name = L"label43";
this->label43->Size = System::Drawing::Size(57, 13);
this->label43->TabIndex = 5;
this->label43->Text = L"SURFACE";
//
// textBox47
//
this->textBox47->Location = System::Drawing::Point(234, 304);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

17

```

this->textBox47->Name = L"textBox47";
this->textBox47->Size = System::Drawing::Size(137, 20);
this->textBox47->TabIndex = 20;
this->textBox47->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox46
//
this->textBox46->Location = System::Drawing::Point(234, 278);
this->textBox46->Name = L"textBox46";
this->textBox46->Size = System::Drawing::Size(137, 20);
this->textBox46->TabIndex = 19;
this->textBox46->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// label39
//
this->label39->AutoSize = true;
this->label39->BackColor = System::Drawing::Color::White;
this->label39->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->label39->ForeColor = System::Drawing::SystemColors::ControlText;
this->label39->Location = System::Drawing::Point(157, 288);
this->label39->Name = L"label39";
this->label39->Size = System::Drawing::Size(71, 13);
this->label39->TabIndex = 18;
this->label39->Text = L"DEFINED BY";
//
// textBox45
//
this->textBox45->BackColor = System::Drawing::Color::Blue;
this->textBox45->ForeColor = System::Drawing::Color::White;
this->textBox45->Location = System::Drawing::Point(84, 304);
this->textBox45->Name = L"textBox45";
this->textBox45->Size = System::Drawing::Size(67, 20);
this->textBox45->TabIndex = 17;
this->textBox45->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox44
//
this->textBox44->BackColor = System::Drawing::Color::Blue;
this->textBox44->ForeColor = System::Drawing::Color::White;
this->textBox44->Location = System::Drawing::Point(84, 278);
this->textBox44->Name = L"textBox44";
this->textBox44->Size = System::Drawing::Size(67, 20);
this->textBox44->TabIndex = 16;
this->textBox44->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// label38
//
this->label38->AutoSize = true;
this->label38->Location = System::Drawing::Point(6, 288);
this->label38->Name = L"label38";
this->label38->Size = System::Drawing::Size(72, 26);
this->label38->TabIndex = 15;
this->label38->Text = L"ALLOWABLE\r\n HEIGHT IS \r\n";
//
// textBox39
//
this->textBox39->Location = System::Drawing::Point(327, 111);
this->textBox39->Name = L"textBox39";
this->textBox39->Size = System::Drawing::Size(100, 20);
this->textBox39->TabIndex = 14;
this->textBox39->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox38
//
this->textBox38->Location = System::Drawing::Point(187, 111);
this->textBox38->Name = L"textBox38";
this->textBox38->Size = System::Drawing::Size(100, 20);

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

18

```

this->textBox38->TabIndex = 13;
this->textBox38->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox37
//
this->textBox37->Location = System::Drawing::Point(30, 111);
this->textBox37->Name = L"textBox37";
this->textBox37->Size = System::Drawing::Size(100, 20);
this->textBox37->TabIndex = 12;
this->textBox37->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox36
//
this->textBox36->Location = System::Drawing::Point(327, 60);
this->textBox36->Name = L"textBox36";
this->textBox36->Size = System::Drawing::Size(100, 20);
this->textBox36->TabIndex = 11;
this->textBox36->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox35
//
this->textBox35->Location = System::Drawing::Point(221, 60);
this->textBox35->Name = L"textBox35";
this->textBox35->Size = System::Drawing::Size(100, 20);
this->textBox35->TabIndex = 10;
this->textBox35->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox34
//
this->textBox34->Location = System::Drawing::Point(115, 60);
this->textBox34->Name = L"textBox34";
this->textBox34->Size = System::Drawing::Size(100, 20);
this->textBox34->TabIndex = 9;
this->textBox34->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox33
//
this->textBox33->Location = System::Drawing::Point(9, 60);
this->textBox33->Name = L"textBox33";
this->textBox33->Size = System::Drawing::Size(100, 20);
this->textBox33->TabIndex = 8;
this->textBox33->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// label35
//
this->label35->AutoSize = true;
this->label35->Location = System::Drawing::Point(357, 95);
this->label35->Name = L"label35";
this->label35->Size = System::Drawing::Size(53, 13);
this->label35->TabIndex = 7;
this->label35->Text = L"CONICAL";
//
// label34
//
this->label34->AutoSize = true;
this->label34->BackColor = System::Drawing::Color::White;
this->label34->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->label34->Location = System::Drawing::Point(169, 95);
this->label34->Name = L"label34";
this->label34->Size = System::Drawing::Size(118, 13);
this->label34->TabIndex = 6;
this->label34->Text = L"OUTER HORIZONTAL";
//
// label33
//
this->label33->AutoSize = true;
this->label33->Location = System::Drawing::Point(341, 44);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

19

```

this->label33->Name = L"label33";
this->label33->Size = System::Drawing::Size(86, 13);
this->label33->TabIndex = 5;
this->label33->Text = L"TRANSITIONAL";
//
// label32
//
this->label32->AutoSize = true;
this->label32->Location = System::Drawing::Point(27, 95);
this->label32->Name = L"label32";
this->label32->Size = System::Drawing::Size(114, 13);
this->label32->TabIndex = 4;
this->label32->Text = L"INNER HORIZONTAL";
//
// label31
//
this->label31->AutoSize = true;
this->label31->Location = System::Drawing::Point(231, 44);
this->label31->Name = L"label31";
this->label31->Size = System::Drawing::Size(81, 13);
this->label31->TabIndex = 3;
this->label31->Text = L"SAFETY ZONE";
//
// label30
//
this->label30->AutoSize = true;
this->label30->Location = System::Drawing::Point(133, 44);
this->label30->Name = L"label30";
this->label30->Size = System::Drawing::Size(58, 13);
this->label30->TabIndex = 2;
this->label30->Text = L"TAKE OFF";
//
// label29
//
this->label29->AutoSize = true;
this->label29->Location = System::Drawing::Point(27, 44);
this->label29->Name = L"label29";
this->label29->Size = System::Drawing::Size(66, 13);
this->label29->TabIndex = 1;
this->label29->Text = L"APPROACH";
//
// label27
//
this->label27->AutoSize = true;
this->label27->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->label27->Location = System::Drawing::Point(60, 20);
this->label27->Name = L"label27";
this->label27->Size = System::Drawing::Size(205, 13);
this->label27->TabIndex = 0;
this->label27->Text = L"THE POINT IS IN THE SURFACES";
//
// tabPage1
//
this->tabPage1->BackColor = System::Drawing::Color::Black;
this->tabPage1->Controls->Add(this->textBox2);
this->tabPage1->Controls->Add(this->textBox1);
this->tabPage1->Controls->Add(this->label3);
this->tabPage1->Controls->Add(this->label2);
this->tabPage1->Controls->Add(this->label1);
this->tabPage1->Controls->Add(this->groupBox1);
this->tabPage1->Controls->Add(this->groupBox2);
this->tabPage1->Location = System::Drawing::Point(4, 22);
this->tabPage1->Name = L"tabPage1";
this->tabPage1->Padding = System::Windows::Forms::Padding(3);
this->tabPage1->Size = System::Drawing::Size(910, 547);
this->tabPage1->TabIndex = 0;
this->tabPage1->Text = L"Runway Data";

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

20

```

//
// textBox2
//
this->textBox2->Location = System::Drawing::Point(465, 46);
this->textBox2->Name = L"textBox2";
this->textBox2->Size = System::Drawing::Size(100, 20);
this->textBox2->TabIndex = 4;
this->textBox2->TextAlign = System::Windows::Forms::HorizontalAlignment::Right;
this->textBox2->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox2_TextChanged);
this->textBox2->MouseHover += gcnew System::EventHandler(this, &Form1::
textBox2_MouseHover);
//
// textBox1
//
this->textBox1->Location = System::Drawing::Point(465, 19);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(100, 20);
this->textBox1->TabIndex = 3;
this->textBox1->TextAlign = System::Windows::Forms::HorizontalAlignment::Right;
this->textBox1->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox1_TextChanged);
this->textBox1->MouseHover += gcnew System::EventHandler(this, &Form1::
textBox1_MouseHover_1);
//
// label3
//
this->label3->AutoSize = true;
this->label3->BackColor = System::Drawing::Color::Transparent;
this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->label3->ForeColor = System::Drawing::SystemColors::Control;
this->label3->Location = System::Drawing::Point(418, 49);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(15, 13);
this->label3->TabIndex = 2;
this->label3->Text = L"Y";
//
// label2
//
this->label2->AutoSize = true;
this->label2->BackColor = System::Drawing::Color::Transparent;
this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->label2->ForeColor = System::Drawing::SystemColors::Control;
this->label2->Location = System::Drawing::Point(418, 22);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(15, 13);
this->label2->TabIndex = 1;
this->label2->Text = L"X";
//
// label1
//
this->label1->AutoSize = true;
this->label1->BackColor = System::Drawing::Color::Transparent;
this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->label1->ForeColor = System::Drawing::SystemColors::Control;
this->label1->Location = System::Drawing::Point(296, 35);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(97, 13);
this->label1->TabIndex = 0;
this->label1->Text = L"Determine Point";
//
// groupBox1
//

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

21

```

        this->groupBox1->BackColor = System::Drawing::Color::Black;
        this->groupBox1->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"groupBox1.BackgroundImage"))));
        this->groupBox1->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Zoom
;

        this->groupBox1->Controls->Add(this->button8);
        this->groupBox1->Controls->Add(this->button7);
        this->groupBox1->Controls->Add(this->button3);
        this->groupBox1->Controls->Add(this->button2);
        this->groupBox1->Controls->Add(this->button1);
        this->groupBox1->Controls->Add(this->textBox7);
        this->groupBox1->Controls->Add(this->label11);
        this->groupBox1->Controls->Add(this->textBox6);
        this->groupBox1->Controls->Add(this->label10);
        this->groupBox1->Controls->Add(this->textBox5);
        this->groupBox1->Controls->Add(this->textBox4);
        this->groupBox1->Controls->Add(this->textBox3);
        this->groupBox1->Controls->Add(this->label9);
        this->groupBox1->Controls->Add(this->label8);
        this->groupBox1->Controls->Add(this->label7);
        this->groupBox1->Controls->Add(this->label6);
        this->groupBox1->Controls->Add(this->label5);
        this->groupBox1->Controls->Add(this->label4);
        this->groupBox1->Controls->Add(this->comboBox2);
        this->groupBox1->Controls->Add(this->comboBox1);
        this->groupBox1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
        this->groupBox1->ForeColor = System::Drawing::SystemColors::Control;
        this->groupBox1->Location = System::Drawing::Point(6, 76);
        this->groupBox1->Name = L"groupBox1";
        this->groupBox1->Size = System::Drawing::Size(453, 462);
        this->groupBox1->TabIndex = 7;
        this->groupBox1->TabStop = false;
        this->groupBox1->Text = L"RUNWAY 1";
        //
        // button8
        //
        this->button8->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"button8.BackgroundImage")));
        this->button8->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;

        this->button8->ForeColor = System::Drawing::SystemColors::ControlText;
        this->button8->Location = System::Drawing::Point(293, 369);
        this->button8->Name = L"button8";
        this->button8->Size = System::Drawing::Size(102, 87);
        this->button8->TabIndex = 23;
        this->button8->Text = L"ComputeHeight";
        this->button8->TextAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button8->UseVisualStyleBackColor = true;
        this->button8->Visible = false;
        this->button8->Click += gcnew System::EventHandler(this, &Form1::button8_Click);
        this->button8->MouseHover += gcnew System::EventHandler(this, &Form1::
button8_MouseHover);
        //
        // button7
        //
        this->button7->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"button7.BackgroundImage")));
        this->button7->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;

        this->button7->ForeColor = System::Drawing::SystemColors::ControlText;
        this->button7->Location = System::Drawing::Point(151, 369);
        this->button7->Name = L"button7";
        this->button7->Size = System::Drawing::Size(102, 87);
        this->button7->TabIndex = 22;
        this->button7->Text = L"ComputeInclusion";
        this->button7->TextAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button7->UseVisualStyleBackColor = true;

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

22

```

        this->button7->Visible = false;
        this->button7->Click += gcnew System::EventHandler(this, &Form1::button7_Click);
        this->button7->MouseHover += gcnew System::EventHandler(this, &Form1::
button7_MouseHover);
        //
        // button3
        //
        this->button3->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"button3.BackgroundImage")));
        this->button3->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;
        this->button3->ForeColor = System::Drawing::SystemColors::ControlText;
        this->button3->Location = System::Drawing::Point(293, 274);
        this->button3->Name = L"button3";
        this->button3->Size = System::Drawing::Size(102, 87);
        this->button3->TabIndex = 18;
        this->button3->Text = L"ComputeHeight";
        this->button3->TextAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button3->UseVisualStyleBackColor = true;
        this->button3->Visible = false;
        this->button3->Click += gcnew System::EventHandler(this, &Form1::button3_Click);
        this->button3->MouseHover += gcnew System::EventHandler(this, &Form1::
button3_MouseHover);
        //
        // button2
        //
        this->button2->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"button2.BackgroundImage")));
        this->button2->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;
        this->button2->ForeColor = System::Drawing::SystemColors::ControlText;
        this->button2->Location = System::Drawing::Point(151, 274);
        this->button2->Name = L"button2";
        this->button2->Size = System::Drawing::Size(102, 87);
        this->button2->TabIndex = 17;
        this->button2->Text = L"ComputeInclusion";
        this->button2->TextAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button2->UseVisualStyleBackColor = true;
        this->button2->Visible = false;
        this->button2->Click += gcnew System::EventHandler(this, &Form1::button2_Click);
        this->button2->MouseHover += gcnew System::EventHandler(this, &Form1::
button2_MouseHover);
        //
        // button1
        //
        this->button1->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;
        this->button1->ForeColor = System::Drawing::SystemColors::ControlText;
        this->button1->Location = System::Drawing::Point(27, 354);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(102, 23);
        this->button1->TabIndex = 16;
        this->button1->Text = L"GenerateSurfaces";
        this->button1->TextAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button1->UseVisualStyleBackColor = true;
        this->button1->Click += gcnew System::EventHandler(this, &Form1::button1_Click);
        this->button1->MouseHover += gcnew System::EventHandler(this, &Form1::
button1_MouseHover);
        //
        // textBox7
        //
        this->textBox7->Location = System::Drawing::Point(110, 244);
        this->textBox7->Name = L"textBox7";
        this->textBox7->Size = System::Drawing::Size(100, 20);
        this->textBox7->TabIndex = 14;
        this->textBox7->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox7_TextChanged);
        this->textBox7->MouseHover += gcnew System::EventHandler(this, &Form1::
textBox7_MouseHover);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

23

```

//
// label11
//
this->label11->AutoSize = true;
this->label11->Location = System::Drawing::Point(27, 247);
this->label11->Name = L"label11";
this->label11->Size = System::Drawing::Size(44, 13);
this->label11->TabIndex = 13;
this->label11->Text = L"Azimuth";
//
// textBox6
//
this->textBox6->Location = System::Drawing::Point(110, 206);
this->textBox6->Name = L"textBox6";
this->textBox6->Size = System::Drawing::Size(100, 20);
this->textBox6->TabIndex = 12;
this->textBox6->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox6_TextChanged);
//
// label10
//
this->label10->AutoSize = true;
this->label10->Location = System::Drawing::Point(27, 209);
this->label10->Name = L"label10";
this->label10->Size = System::Drawing::Size(40, 13);
this->label10->TabIndex = 11;
this->label10->Text = L"Length";
//
// textBox5
//
this->textBox5->Location = System::Drawing::Point(191, 159);
this->textBox5->Name = L"textBox5";
this->textBox5->Size = System::Drawing::Size(100, 20);
this->textBox5->TabIndex = 10;
this->textBox5->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox5_TextChanged);
//
// textBox4
//
this->textBox4->Location = System::Drawing::Point(191, 136);
this->textBox4->Name = L"textBox4";
this->textBox4->Size = System::Drawing::Size(100, 20);
this->textBox4->TabIndex = 9;
this->textBox4->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox4_TextChanged);
//
// textBox3
//
this->textBox3->Location = System::Drawing::Point(191, 112);
this->textBox3->Name = L"textBox3";
this->textBox3->Size = System::Drawing::Size(100, 20);
this->textBox3->TabIndex = 8;
this->textBox3->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox3_TextChanged);
//
// label9
//
this->label9->AutoSize = true;
this->label9->Location = System::Drawing::Point(171, 162);
this->label9->Name = L"label9";
this->label9->Size = System::Drawing::Size(14, 13);
this->label9->TabIndex = 7;
this->label9->Text = L"Z";
//
// label8
//
this->label8->AutoSize = true;
this->label8->Location = System::Drawing::Point(171, 139);
this->label8->Name = L"label8";

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

24

```

this->label8->Size = System::Drawing::Size(14, 13);
this->label8->TabIndex = 6;
this->label8->Text = L"Y";
//
// label7
//
this->label7->AutoSize = true;
this->label7->Location = System::Drawing::Point(171, 115);
this->label7->Name = L"label7";
this->label7->Size = System::Drawing::Size(14, 13);
this->label7->TabIndex = 5;
this->label7->Text = L"X";
//
// label6
//
this->label6->AutoSize = true;
this->label6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Underline, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->label6->Location = System::Drawing::Point(87, 139);
this->label6->Name = L"label6";
this->label6->Size = System::Drawing::Size(38, 13);
this->label6->TabIndex = 4;
this->label6->Text = L"Centre";
//
// label5
//
this->label5->AutoSize = true;
this->label5->Location = System::Drawing::Point(24, 68);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(72, 13);
this->label5->TabIndex = 3;
this->label5->Text = L"Code Number";
//
// label4
//
this->label4->AutoSize = true;
this->label4->Location = System::Drawing::Point(24, 27);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(80, 13);
this->label4->TabIndex = 2;
this->label4->Text = L"Approach Type";
//
// comboBox2
//
this->comboBox2->DropDownStyle = System::Windows::Forms::ComboBoxStyle::
DropDownList;
this->comboBox2->FormattingEnabled = true;
this->comboBox2->Items->AddRange(gcnew cli::array< System::Object^ >(4) {L"1", L
"2", L"3", L"4"});
this->comboBox2->Location = System::Drawing::Point(110, 68);
this->comboBox2->Name = L"comboBox2";
this->comboBox2->Size = System::Drawing::Size(121, 21);
this->comboBox2->TabIndex = 1;
this->comboBox2->SelectedIndexChanged += gcnew System::EventHandler(this, &Form1::
comboBox2_SelectedIndexChanged);
//
// comboBox1
//
this->comboBox1->DropDownStyle = System::Windows::Forms::ComboBoxStyle::
DropDownList;
this->comboBox1->FormattingEnabled = true;
this->comboBox1->Items->AddRange(gcnew cli::array< System::Object^ >(3) {L"Non-
instrument", L"Non-precision", L"Precision"});
this->comboBox1->Location = System::Drawing::Point(110, 24);
this->comboBox1->Name = L"comboBox1";
this->comboBox1->Size = System::Drawing::Size(121, 21);
this->comboBox1->TabIndex = 0;
this->comboBox1->SelectedIndexChanged += gcnew System::EventHandler(this, &Form1::

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

25

```

comboBox1_SelectedIndexChanged);
    //
    // groupBox2
    //
    this->groupBox2->BackColor = System::Drawing::Color::Black;
    this->groupBox2->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"groupBox2.BackgroundImage")));
    this->groupBox2->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Zoom
;

    this->groupBox2->Controls->Add(this->button10);
    this->groupBox2->Controls->Add(this->button9);
    this->groupBox2->Controls->Add(this->button6);
    this->groupBox2->Controls->Add(this->button5);
    this->groupBox2->Controls->Add(this->button4);
    this->groupBox2->Controls->Add(this->label122);
    this->groupBox2->Controls->Add(this->label121);
    this->groupBox2->Controls->Add(this->textBox12);
    this->groupBox2->Controls->Add(this->textBox11);
    this->groupBox2->Controls->Add(this->textBox10);
    this->groupBox2->Controls->Add(this->textBox9);
    this->groupBox2->Controls->Add(this->textBox8);
    this->groupBox2->Controls->Add(this->label120);
    this->groupBox2->Controls->Add(this->label119);
    this->groupBox2->Controls->Add(this->label118);
    this->groupBox2->Controls->Add(this->label117);
    this->groupBox2->Controls->Add(this->comboBox4);
    this->groupBox2->Controls->Add(this->label116);
    this->groupBox2->Controls->Add(this->comboBox3);
    this->groupBox2->Controls->Add(this->label115);
    this->groupBox2->ForeColor = System::Drawing::SystemColors::Control;
    this->groupBox2->Location = System::Drawing::Point(465, 76);
    this->groupBox2->Name = L"groupBox2";
    this->groupBox2->Size = System::Drawing::Size(437, 462);
    this->groupBox2->TabIndex = 8;
    this->groupBox2->TabStop = false;
    this->groupBox2->Text = L"RUNWAY 2";
    //
    // button10
    //
    this->button10->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"button10.BackgroundImage")));
    this->button10->BackgroundImageLayout = System::Windows::Forms::ImageLayout::
Center;
    this->button10->ForeColor = System::Drawing::SystemColors::ControlText;
    this->button10->Location = System::Drawing::Point(291, 369);
    this->button10->Name = L"button10";
    this->button10->Size = System::Drawing::Size(102, 87);
    this->button10->TabIndex = 25;
    this->button10->Text = L"ComputeHeight";
    this->button10->TextAlign = System::Drawing::ContentAlignment::TopCenter;
    this->button10->UseVisualStyleBackColor = true;
    this->button10->Visible = false;
    this->button10->Click += gcnew System::EventHandler(this, &Form1::button10_Click);
    //
    // button9
    //
    this->button9->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"button9.BackgroundImage")));
    this->button9->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;

    this->button9->ForeColor = System::Drawing::SystemColors::ControlText;
    this->button9->Location = System::Drawing::Point(159, 369);
    this->button9->Name = L"button9";
    this->button9->Size = System::Drawing::Size(102, 87);
    this->button9->TabIndex = 24;
    this->button9->Text = L"ComputeInclusion";
    this->button9->TextAlign = System::Drawing::ContentAlignment::TopCenter;
    this->button9->UseVisualStyleBackColor = true;
    this->button9->Visible = false;

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

26

```

        this->button9->Click += gcnew System::EventHandler(this, &Form1::button9_Click);
        //
        // button6
        //
        this->button6->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"button6.BackgroundImage")));
        this->button6->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;

        this->button6->ForeColor = System::Drawing::SystemColors::ControlText;
        this->button6->Location = System::Drawing::Point(291, 274);
        this->button6->Name = L"button6";
        this->button6->Size = System::Drawing::Size(102, 87);
        this->button6->TabIndex = 20;
        this->button6->Text = L"ComputeHeight";
        this->button6->TextAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button6->UseVisualStyleBackColor = true;
        this->button6->Visible = false;
        this->button6->Click += gcnew System::EventHandler(this, &Form1::button6_Click);
        //
        // button5
        //
        this->button5->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"button5.BackgroundImage")));
        this->button5->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;

        this->button5->ForeColor = System::Drawing::SystemColors::ControlText;
        this->button5->Location = System::Drawing::Point(159, 274);
        this->button5->Name = L"button5";
        this->button5->Size = System::Drawing::Size(102, 87);
        this->button5->TabIndex = 19;
        this->button5->Text = L"ComputeInclusion";
        this->button5->TextAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button5->UseVisualStyleBackColor = true;
        this->button5->Visible = false;
        this->button5->Click += gcnew System::EventHandler(this, &Form1::button5_Click);
        //
        // button4
        //
        this->button4->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center
;

        this->button4->ForeColor = System::Drawing::SystemColors::ControlText;
        this->button4->Location = System::Drawing::Point(25, 354);
        this->button4->Name = L"button4";
        this->button4->Size = System::Drawing::Size(102, 23);
        this->button4->TabIndex = 18;
        this->button4->Text = L"GenerateSurfaces";
        this->button4->TextAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button4->UseVisualStyleBackColor = true;
        this->button4->Click += gcnew System::EventHandler(this, &Form1::button4_Click);
        //
        // label22
        //
        this->label22->AutoSize = true;
        this->label22->Location = System::Drawing::Point(28, 247);
        this->label22->Name = L"label22";
        this->label22->Size = System::Drawing::Size(44, 13);
        this->label22->TabIndex = 17;
        this->label22->Text = L"Azimuth";
        //
        // label21
        //
        this->label21->AutoSize = true;
        this->label21->Location = System::Drawing::Point(28, 213);
        this->label21->Name = L"label21";
        this->label21->Size = System::Drawing::Size(40, 13);
        this->label21->TabIndex = 16;
        this->label21->Text = L"Length";
        //
        // textBox12

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

27

```

//
this->textBox12->Location = System::Drawing::Point(114, 244);
this->textBox12->Name = L"textBox12";
this->textBox12->Size = System::Drawing::Size(100, 20);
this->textBox12->TabIndex = 15;
this->textBox12->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox12_TextChanged);
this->textBox12->MouseHover += gcnew System::EventHandler(this, &Form1::
textBox12_MouseHover);
//
// textBox11
//
this->textBox11->Location = System::Drawing::Point(114, 206);
this->textBox11->Name = L"textBox11";
this->textBox11->Size = System::Drawing::Size(100, 20);
this->textBox11->TabIndex = 14;
this->textBox11->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox11_TextChanged);
//
// textBox10
//
this->textBox10->Location = System::Drawing::Point(188, 159);
this->textBox10->Name = L"textBox10";
this->textBox10->Size = System::Drawing::Size(100, 20);
this->textBox10->TabIndex = 13;
this->textBox10->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox10_TextChanged);
//
// textBox9
//
this->textBox9->Location = System::Drawing::Point(188, 136);
this->textBox9->Name = L"textBox9";
this->textBox9->Size = System::Drawing::Size(100, 20);
this->textBox9->TabIndex = 12;
this->textBox9->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox9_TextChanged);
//
// textBox8
//
this->textBox8->Location = System::Drawing::Point(188, 112);
this->textBox8->Name = L"textBox8";
this->textBox8->Size = System::Drawing::Size(100, 20);
this->textBox8->TabIndex = 11;
this->textBox8->TextChanged += gcnew System::EventHandler(this, &Form1::
textBox8_TextChanged);
//
// label20
//
this->label20->AutoSize = true;
this->label20->Location = System::Drawing::Point(168, 162);
this->label20->Name = L"label20";
this->label20->Size = System::Drawing::Size(14, 13);
this->label20->TabIndex = 10;
this->label20->Text = L"Z";
//
// label19
//
this->label19->AutoSize = true;
this->label19->Location = System::Drawing::Point(168, 139);
this->label19->Name = L"label19";
this->label19->Size = System::Drawing::Size(14, 13);
this->label19->TabIndex = 9;
this->label19->Text = L"Y";
//
// label18
//
this->label18->AutoSize = true;
this->label18->Location = System::Drawing::Point(168, 115);
this->label18->Name = L"label18";

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

28

```

this->label18->Size = System::Drawing::Size(14, 13);
this->label18->TabIndex = 8;
this->label18->Text = L"X";
//
// label17
//
this->label17->AutoSize = true;
this->label17->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Underline, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->label17->Location = System::Drawing::Point(81, 139);
this->label17->Name = L"label17";
this->label17->Size = System::Drawing::Size(38, 13);
this->label17->TabIndex = 7;
this->label17->Text = L"Centre";
//
// comboBox4
//
this->comboBox4->DropDownStyle = System::Windows::Forms::ComboBoxStyle::
DropDnList;
this->comboBox4->FormattingEnabled = true;
this->comboBox4->Items->AddRange(gcnew cli::array< System::Object^  >(4) {L"1", L
"2", L"3", L"4"});
this->comboBox4->Location = System::Drawing::Point(114, 65);
this->comboBox4->Name = L"comboBox4";
this->comboBox4->Size = System::Drawing::Size(121, 21);
this->comboBox4->TabIndex = 6;
this->comboBox4->SelectedIndexChanged += gcnew System::EventHandler(this, &Form1::
comboBox4_SelectedIndexChanged);
//
// label16
//
this->label16->AutoSize = true;
this->label16->Location = System::Drawing::Point(28, 68);
this->label16->Name = L"label16";
this->label16->Size = System::Drawing::Size(72, 13);
this->label16->TabIndex = 5;
this->label16->Text = L"Code Number";
//
// comboBox3
//
this->comboBox3->DropDownStyle = System::Windows::Forms::ComboBoxStyle::
DropDnList;
this->comboBox3->FormattingEnabled = true;
this->comboBox3->Items->AddRange(gcnew cli::array< System::Object^  >(3) {L"Non-
instrument", L"Non-precision", L"Precision"});
this->comboBox3->Location = System::Drawing::Point(114, 24);
this->comboBox3->Name = L"comboBox3";
this->comboBox3->Size = System::Drawing::Size(121, 21);
this->comboBox3->TabIndex = 4;
this->comboBox3->SelectedIndexChanged += gcnew System::EventHandler(this, &Form1::
comboBox3_SelectedIndexChanged);
//
// label15
//
this->label15->AutoSize = true;
this->label15->Location = System::Drawing::Point(28, 27);
this->label15->Name = L"label15";
this->label15->Size = System::Drawing::Size(80, 13);
this->label15->TabIndex = 3;
this->label15->Text = L"Approach Type";
//
// tabControl1
//
this->tabControl1->Controls->Add(this->tabPage1);
this->tabControl1->Controls->Add(this->tabPage2);
this->tabControl1->Controls->Add(this->tabPage4);
this->tabControl1->Controls->Add(this->tabPage5);
this->tabControl1->Controls->Add(this->tabPage6);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

29

```

this->tabControl1->Controls->Add(this->tabPage7);
this->tabControl1->Location = System::Drawing::Point(0, 27);
this->tabControl1->Name = L"tabControl1";
this->tabControl1->SelectedIndex = 0;
this->tabControl1->Size = System::Drawing::Size(918, 573);
this->tabControl1->TabIndex = 1;
//
// tabPage4
//
this->tabPage4->BackColor = System::Drawing::Color::Silver;
this->tabPage4->Controls->Add(this->groupBox4);
this->tabPage4->Location = System::Drawing::Point(4, 22);
this->tabPage4->Name = L"tabPage4";
this->tabPage4->Padding = System::Windows::Forms::Padding(3);
this->tabPage4->Size = System::Drawing::Size(910, 547);
this->tabPage4->TabIndex = 4;
this->tabPage4->Text = L"Runway Strip and transitional surface";
//
// groupBox4
//
this->groupBox4->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"groupBox4.BackgroundImage")));
this->groupBox4->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Zoom
;

this->groupBox4->Controls->Add(this->textBox67);
this->groupBox4->Controls->Add(this->textBox32);
this->groupBox4->Controls->Add(this->textBox65);
this->groupBox4->Controls->Add(this->textBox30);
this->groupBox4->Controls->Add(this->textBox66);
this->groupBox4->Controls->Add(this->textBox31);
this->groupBox4->Location = System::Drawing::Point(3, 6);
this->groupBox4->Name = L"groupBox4";
this->groupBox4->Size = System::Drawing::Size(911, 538);
this->groupBox4->TabIndex = 0;
this->groupBox4->TabStop = false;
this->groupBox4->Text = L"Upper values are refering to RUNWAY 1
Lower val"
L"ues to RUNWAY 2";
//
// textBox67
//
this->textBox67->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->textBox67->Location = System::Drawing::Point(514, 371);
this->textBox67->Name = L"textBox67";
this->textBox67->ReadOnly = true;
this->textBox67->Size = System::Drawing::Size(69, 20);
this->textBox67->TabIndex = 44;
this->textBox67->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox32
//
this->textBox32->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->textBox32->Location = System::Drawing::Point(514, 151);
this->textBox32->Name = L"textBox32";
this->textBox32->ReadOnly = true;
this->textBox32->Size = System::Drawing::Size(69, 20);
this->textBox32->TabIndex = 43;
this->textBox32->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox65
//
this->textBox65->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(0)));
this->textBox65->Location = System::Drawing::Point(801, 281);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

30

```

    this->textBox65->Name = L"textBox65";
    this->textBox65->ReadOnly = true;
    this->textBox65->Size = System::Drawing::Size(40, 20);
    this->textBox65->TabIndex = 42;
    this->textBox65->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox30
    //
    this->textBox30->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox30->Location = System::Drawing::Point(801, 234);
    this->textBox30->Name = L"textBox30";
    this->textBox30->ReadOnly = true;
    this->textBox30->Size = System::Drawing::Size(40, 20);
    this->textBox30->TabIndex = 41;
    this->textBox30->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox66
    //
    this->textBox66->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox66->Location = System::Drawing::Point(486, 281);
    this->textBox66->Name = L"textBox66";
    this->textBox66->ReadOnly = true;
    this->textBox66->Size = System::Drawing::Size(55, 20);
    this->textBox66->TabIndex = 40;
    this->textBox66->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox31
    //
    this->textBox31->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox31->Location = System::Drawing::Point(486, 234);
    this->textBox31->Name = L"textBox31";
    this->textBox31->ReadOnly = true;
    this->textBox31->Size = System::Drawing::Size(55, 20);
    this->textBox31->TabIndex = 19;
    this->textBox31->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // tabPage5
    //
    this->tabPage5->BackColor = System::Drawing::Color::Silver;
    this->tabPage5->Controls->Add(this->groupBox6);
    this->tabPage5->Location = System::Drawing::Point(4, 22);
    this->tabPage5->Name = L"tabPage5";
    this->tabPage5->Padding = System::Windows::Forms::Padding(3);
    this->tabPage5->Size = System::Drawing::Size(910, 547);
    this->tabPage5->TabIndex = 5;
    this->tabPage5->Text = L"Cyclic surfaces";
    //
    // groupBox6
    //
    this->groupBox6->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"groupBox6.BackgroundImage")));
    this->groupBox6->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Zoom
;

    this->groupBox6->Controls->Add(this->textBox58);
    this->groupBox6->Controls->Add(this->textBox57);
    this->groupBox6->Controls->Add(this->textBox23);
    this->groupBox6->Controls->Add(this->textBox22);
    this->groupBox6->Location = System::Drawing::Point(8, 6);
    this->groupBox6->Name = L"groupBox6";
    this->groupBox6->Size = System::Drawing::Size(896, 535);
    this->groupBox6->TabIndex = 0;
    this->groupBox6->TabStop = false;
    this->groupBox6->Text = L"Upper values are refering to RUNWAY 1

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

31

```

        Lower val"
        L"ues to RUNWAY 2";
    //
    // textBox58
    //
    this->textBox58->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox58->Location = System::Drawing::Point(329, 346);
    this->textBox58->Name = L"textBox58";
    this->textBox58->ReadOnly = true;
    this->textBox58->Size = System::Drawing::Size(100, 20);
    this->textBox58->TabIndex = 32;
    this->textBox58->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox57
    //
    this->textBox57->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox57->Location = System::Drawing::Point(470, 310);
    this->textBox57->Name = L"textBox57";
    this->textBox57->ReadOnly = true;
    this->textBox57->Size = System::Drawing::Size(100, 20);
    this->textBox57->TabIndex = 31;
    this->textBox57->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox23
    //
    this->textBox23->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox23->Location = System::Drawing::Point(329, 204);
    this->textBox23->Name = L"textBox23";
    this->textBox23->ReadOnly = true;
    this->textBox23->Size = System::Drawing::Size(100, 20);
    this->textBox23->TabIndex = 11;
    this->textBox23->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox22
    //
    this->textBox22->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox22->Location = System::Drawing::Point(470, 187);
    this->textBox22->Name = L"textBox22";
    this->textBox22->ReadOnly = true;
    this->textBox22->Size = System::Drawing::Size(100, 20);
    this->textBox22->TabIndex = 10;
    this->textBox22->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // tabPage6
    //
    this->tabPage6->BackColor = System::Drawing::Color::Silver;
    this->tabPage6->Controls->Add(this->groupBox7);
    this->tabPage6->Location = System::Drawing::Point(4, 22);
    this->tabPage6->Name = L"tabPage6";
    this->tabPage6->Padding = System::Windows::Forms::Padding(3);
    this->tabPage6->Size = System::Drawing::Size(910, 547);
    this->tabPage6->TabIndex = 6;
    this->tabPage6->Text = L"Approach Surface";
    //
    // groupBox7
    //
    this->groupBox7->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"groupBox7.BackgroundImage")));
    this->groupBox7->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Zoom
;
    this->groupBox7->Controls->Add(this->textBox21);

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

32

```

this->groupBox7->Controls->Add(this->textBox56);
this->groupBox7->Controls->Add(this->textBox55);
this->groupBox7->Controls->Add(this->textBox54);
this->groupBox7->Controls->Add(this->textBox53);
this->groupBox7->Controls->Add(this->textBox52);
this->groupBox7->Controls->Add(this->textBox51);
this->groupBox7->Controls->Add(this->textBox50);
this->groupBox7->Controls->Add(this->textBox20);
this->groupBox7->Controls->Add(this->textBox19);
this->groupBox7->Controls->Add(this->textBox18);
this->groupBox7->Controls->Add(this->textBox17);
this->groupBox7->Controls->Add(this->textBox16);
this->groupBox7->Controls->Add(this->textBox15);
this->groupBox7->Controls->Add(this->textBox49);
this->groupBox7->Controls->Add(this->textBox14);
this->groupBox7->Controls->Add(this->textBox48);
this->groupBox7->Controls->Add(this->textBox13);
this->groupBox7->Location = System::Drawing::Point(8, 6);
this->groupBox7->Name = L"groupBox7";
this->groupBox7->Size = System::Drawing::Size(896, 533);
this->groupBox7->TabIndex = 0;
this->groupBox7->TabStop = false;
this->groupBox7->Text = L"Upper values are referring to RUNWAY 1
    Lower val"
    L"ues to RUNWAY 2";
//
// textBox21
//
this->textBox21->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox21->Location = System::Drawing::Point(345, 107);
this->textBox21->Name = L"textBox21";
this->textBox21->ReadOnly = true;
this->textBox21->Size = System::Drawing::Size(54, 20);
this->textBox21->TabIndex = 38;
this->textBox21->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox56
//
this->textBox56->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox56->Location = System::Drawing::Point(345, 223);
this->textBox56->Name = L"textBox56";
this->textBox56->ReadOnly = true;
this->textBox56->Size = System::Drawing::Size(54, 20);
this->textBox56->TabIndex = 37;
this->textBox56->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox55
//
this->textBox55->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox55->Location = System::Drawing::Point(123, 390);
this->textBox55->Name = L"textBox55";
this->textBox55->ReadOnly = true;
this->textBox55->Size = System::Drawing::Size(62, 20);
this->textBox55->TabIndex = 36;
this->textBox55->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox54
//
this->textBox54->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox54->Location = System::Drawing::Point(123, 186);
this->textBox54->Name = L"textBox54";

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

33

```

    this->textBox54->ReadOnly = true;
    this->textBox54->Size = System::Drawing::Size(62, 20);
    this->textBox54->TabIndex = 35;
    this->textBox54->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox53
    //
    this->textBox53->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox53->Location = System::Drawing::Point(323, 482);
    this->textBox53->Name = L"textBox53";
    this->textBox53->ReadOnly = true;
    this->textBox53->Size = System::Drawing::Size(56, 20);
    this->textBox53->TabIndex = 34;
    this->textBox53->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox52
    //
    this->textBox52->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox52->Location = System::Drawing::Point(279, 186);
    this->textBox52->Name = L"textBox52";
    this->textBox52->ReadOnly = true;
    this->textBox52->Size = System::Drawing::Size(56, 20);
    this->textBox52->TabIndex = 33;
    this->textBox52->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox51
    //
    this->textBox51->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox51->Location = System::Drawing::Point(460, 507);
    this->textBox51->Name = L"textBox51";
    this->textBox51->ReadOnly = true;
    this->textBox51->Size = System::Drawing::Size(55, 20);
    this->textBox51->TabIndex = 32;
    this->textBox51->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox50
    //
    this->textBox50->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox50->Location = System::Drawing::Point(418, 186);
    this->textBox50->Name = L"textBox50";
    this->textBox50->ReadOnly = true;
    this->textBox50->Size = System::Drawing::Size(54, 20);
    this->textBox50->TabIndex = 31;
    this->textBox50->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox20
    //
    this->textBox20->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox20->Location = System::Drawing::Point(123, 350);
    this->textBox20->Name = L"textBox20";
    this->textBox20->ReadOnly = true;
    this->textBox20->Size = System::Drawing::Size(62, 20);
    this->textBox20->TabIndex = 30;
    this->textBox20->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox19
    //
    this->textBox19->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

34

```

        static_cast<System::Byte>(0));
this->textBox19->Location = System::Drawing::Point(123, 143);
this->textBox19->Name = L"textBox19";
this->textBox19->ReadOnly = true;
this->textBox19->Size = System::Drawing::Size(62, 20);
this->textBox19->TabIndex = 29;
this->textBox19->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
//  textBox18
//
this->textBox18->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0));
this->textBox18->Location = System::Drawing::Point(323, 421);
this->textBox18->Name = L"textBox18";
this->textBox18->ReadOnly = true;
this->textBox18->Size = System::Drawing::Size(56, 20);
this->textBox18->TabIndex = 28;
this->textBox18->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
//  textBox17
//
this->textBox17->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0));
this->textBox17->Location = System::Drawing::Point(279, 143);
this->textBox17->Name = L"textBox17";
this->textBox17->ReadOnly = true;
this->textBox17->Size = System::Drawing::Size(56, 20);
this->textBox17->TabIndex = 27;
this->textBox17->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
//  textBox16
//
this->textBox16->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0));
this->textBox16->Location = System::Drawing::Point(460, 464);
this->textBox16->Name = L"textBox16";
this->textBox16->ReadOnly = true;
this->textBox16->Size = System::Drawing::Size(55, 20);
this->textBox16->TabIndex = 26;
this->textBox16->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
//  textBox15
//
this->textBox15->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0));
this->textBox15->Location = System::Drawing::Point(418, 143);
this->textBox15->Name = L"textBox15";
this->textBox15->ReadOnly = true;
this->textBox15->Size = System::Drawing::Size(54, 20);
this->textBox15->TabIndex = 25;
this->textBox15->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
//  textBox49
//
this->textBox49->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0));
this->textBox49->Location = System::Drawing::Point(536, 223);
this->textBox49->Name = L"textBox49";
this->textBox49->ReadOnly = true;
this->textBox49->Size = System::Drawing::Size(56, 20);
this->textBox49->TabIndex = 24;
this->textBox49->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
//  textBox14

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

35

```

//
this->textBox14->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox14->Location = System::Drawing::Point(536, 107);
this->textBox14->Name = L"textBox14";
this->textBox14->ReadOnly = true;
this->textBox14->Size = System::Drawing::Size(56, 20);
this->textBox14->TabIndex = 23;
this->textBox14->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox48
//
this->textBox48->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox48->Location = System::Drawing::Point(501, 186);
this->textBox48->Name = L"textBox48";
this->textBox48->ReadOnly = true;
this->textBox48->Size = System::Drawing::Size(51, 20);
this->textBox48->TabIndex = 22;
this->textBox48->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox13
//
this->textBox13->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox13->Location = System::Drawing::Point(501, 143);
this->textBox13->Name = L"textBox13";
this->textBox13->ReadOnly = true;
this->textBox13->Size = System::Drawing::Size(51, 20);
this->textBox13->TabIndex = 1;
this->textBox13->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// tabPage7
//
this->tabPage7->BackColor = System::Drawing::Color::Silver;
this->tabPage7->Controls->Add(this->groupBox8);
this->tabPage7->Location = System::Drawing::Point(4, 22);
this->tabPage7->Name = L"tabPage7";
this->tabPage7->Padding = System::Windows::Forms::Padding(3);
this->tabPage7->Size = System::Drawing::Size(910, 547);
this->tabPage7->TabIndex = 7;
this->tabPage7->Text = L"Take off surface";
//
// groupBox8
//
this->groupBox8->BackgroundImage = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"groupBox8.BackgroundImage")));
this->groupBox8->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Zoom
;

this->groupBox8->Controls->Add(this->textBox60);
this->groupBox8->Controls->Add(this->textBox59);
this->groupBox8->Controls->Add(this->textBox64);
this->groupBox8->Controls->Add(this->textBox63);
this->groupBox8->Controls->Add(this->textBox62);
this->groupBox8->Controls->Add(this->textBox61);
this->groupBox8->Controls->Add(this->textBox29);
this->groupBox8->Controls->Add(this->textBox28);
this->groupBox8->Controls->Add(this->textBox27);
this->groupBox8->Controls->Add(this->textBox26);
this->groupBox8->Controls->Add(this->textBox25);
this->groupBox8->Controls->Add(this->textBox24);
this->groupBox8->Location = System::Drawing::Point(8, 6);
this->groupBox8->Name = L"groupBox8";
this->groupBox8->Size = System::Drawing::Size(896, 533);
this->groupBox8->TabIndex = 0;
this->groupBox8->TabStop = false;

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

36

```

    this->groupBox8->Text = L"Upper values are refering to RUNWAY 1
        Lower val"
        L"ues to RUNWAY 2";
    //
    // textBox60
    //
    this->textBox60->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox60->Location = System::Drawing::Point(179, 238);
    this->textBox60->Name = L"textBox60";
    this->textBox60->ReadOnly = true;
    this->textBox60->Size = System::Drawing::Size(61, 20);
    this->textBox60->TabIndex = 40;
    this->textBox60->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox59
    //
    this->textBox59->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox59->Location = System::Drawing::Point(224, 196);
    this->textBox59->Name = L"textBox59";
    this->textBox59->ReadOnly = true;
    this->textBox59->Size = System::Drawing::Size(60, 20);
    this->textBox59->TabIndex = 39;
    this->textBox59->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox64
    //
    this->textBox64->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox64->Location = System::Drawing::Point(489, 222);
    this->textBox64->Name = L"textBox64";
    this->textBox64->ReadOnly = true;
    this->textBox64->Size = System::Drawing::Size(57, 20);
    this->textBox64->TabIndex = 38;
    this->textBox64->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox63
    //
    this->textBox63->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox63->Location = System::Drawing::Point(522, 434);
    this->textBox63->Name = L"textBox63";
    this->textBox63->ReadOnly = true;
    this->textBox63->Size = System::Drawing::Size(63, 20);
    this->textBox63->TabIndex = 37;
    this->textBox63->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox62
    //
    this->textBox62->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));
    this->textBox62->Location = System::Drawing::Point(635, 196);
    this->textBox62->Name = L"textBox62";
    this->textBox62->ReadOnly = true;
    this->textBox62->Size = System::Drawing::Size(70, 20);
    this->textBox62->TabIndex = 36;
    this->textBox62->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // textBox61
    //
    this->textBox61->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(0)));

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

37

```

this->textBox61->Location = System::Drawing::Point(365, 196);
this->textBox61->Name = L"textBox61";
this->textBox61->ReadOnly = true;
this->textBox61->Size = System::Drawing::Size(63, 20);
this->textBox61->TabIndex = 35;
this->textBox61->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox29
//
this->textBox29->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8. 25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox29->Location = System::Drawing::Point(489, 129);
this->textBox29->Name = L"textBox29";
this->textBox29->ReadOnly = true;
this->textBox29->Size = System::Drawing::Size(57, 20);
this->textBox29->TabIndex = 17;
this->textBox29->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox28
//
this->textBox28->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8. 25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox28->Location = System::Drawing::Point(522, 399);
this->textBox28->Name = L"textBox28";
this->textBox28->ReadOnly = true;
this->textBox28->Size = System::Drawing::Size(63, 20);
this->textBox28->TabIndex = 16;
this->textBox28->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox27
//
this->textBox27->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8. 25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox27->Location = System::Drawing::Point(635, 155);
this->textBox27->Name = L"textBox27";
this->textBox27->ReadOnly = true;
this->textBox27->Size = System::Drawing::Size(70, 20);
this->textBox27->TabIndex = 15;
this->textBox27->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox26
//
this->textBox26->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8. 25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox26->Location = System::Drawing::Point(365, 155);
this->textBox26->Name = L"textBox26";
this->textBox26->ReadOnly = true;
this->textBox26->Size = System::Drawing::Size(63, 20);
this->textBox26->TabIndex = 14;
this->textBox26->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox25
//
this->textBox25->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8. 25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
this->textBox25->Location = System::Drawing::Point(179, 114);
this->textBox25->Name = L"textBox25";
this->textBox25->ReadOnly = true;
this->textBox25->Size = System::Drawing::Size(61, 20);
this->textBox25->TabIndex = 13;
this->textBox25->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
//
// textBox24
//

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

38

```

    this->textBox24->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.
25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(0)));
    this->textBox24->Location = System::Drawing::Point(224, 155);
    this->textBox24->Name = L"textBox24";
    this->textBox24->ReadOnly = true;
    this->textBox24->Size = System::Drawing::Size(60, 20);
    this->textBox24->TabIndex = 12;
    this->textBox24->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;
    //
    // saveFileDialog1
    //
    this->saveFileDialog1->Filter = L"Text Files (.txt)|*.txt";
    this->saveFileDialog1->FileOk += gcnew System::ComponentModel::CancelEventHandler
(this, &Form1::saveFileDialog1_FileOk);
    //
    // toolStrip1
    //
    this->toolStrip1->BackColor = System::Drawing::SystemColors::ControlText;
    this->toolStrip1->Dock = System::Windows::Forms::DockStyle::Right;
    this->toolStrip1->GripStyle = System::Windows::Forms::ToolStripGripStyle::Hidden;
    this->toolStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::
ToolStripItem^ >(5) {this->newToolStripButton,
    this->openToolStripButton, this->saveToolStripButton, this->
toolStripSeparator4, this->toolStripButton1});
    this->toolStrip1->Location = System::Drawing::Point(920, 24);
    this->toolStrip1->Name = L"toolStrip1";
    this->toolStrip1->Size = System::Drawing::Size(24, 576);
    this->toolStrip1->TabIndex = 2;
    this->toolStrip1->Text = L"toolStrip1";
    //
    // newToolStripButton
    //
    this->newToolStripButton->DisplayStyle = System::Windows::Forms::
ToolStripItemDisplayStyle::Image;
    this->newToolStripButton->Image = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"newToolStripButton.Image")));
    this->newToolStripButton->ImageTransparentColor = System::Drawing::Color::Magenta;
    this->newToolStripButton->Name = L"newToolStripButton";
    this->newToolStripButton->Size = System::Drawing::Size(21, 20);
    this->newToolStripButton->Text = L"&New";
    this->newToolStripButton->Click += gcnew System::EventHandler(this, &Form1::
newToolStripMenuItem_Click);
    //
    // openToolStripButton
    //
    this->openToolStripButton->DisplayStyle = System::Windows::Forms::
ToolStripItemDisplayStyle::Image;
    this->openToolStripButton->Image = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"openToolStripButton.Image")));
    this->openToolStripButton->ImageTransparentColor = System::Drawing::Color::Magenta;
;
    this->openToolStripButton->Name = L"openToolStripButton";
    this->openToolStripButton->Size = System::Drawing::Size(21, 20);
    this->openToolStripButton->Text = L"&Open";
    //
    // saveToolStripButton
    //
    this->saveToolStripButton->DisplayStyle = System::Windows::Forms::
ToolStripItemDisplayStyle::Image;
    this->saveToolStripButton->Image = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"saveToolStripButton.Image")));
    this->saveToolStripButton->ImageTransparentColor = System::Drawing::Color::Magenta;
;
    this->saveToolStripButton->Name = L"saveToolStripButton";
    this->saveToolStripButton->Size = System::Drawing::Size(21, 20);
    this->saveToolStripButton->Text = L"&Export Printable Text";
    this->saveToolStripButton->Click += gcnew System::EventHandler(this, &Form1::
saveToolStripMenuItem_Click);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

39

```

//
// toolStripSeparator4
//
this->toolStripSeparator4->Name = L"toolStripSeparator4";
this->toolStripSeparator4->Size = System::Drawing::Size(21, 6);
//
// toolStripButton1
//
this->toolStripButton1->DisplayStyle = System::Windows::Forms::
ToolStripItemDisplayStyle::Image;
this->toolStripButton1->Image = (cli::safe_cast<System::Drawing::Image^ >
(resources->GetObject(L"toolStripButton1.Image")));
this->toolStripButton1->ImageTransparentColor = System::Drawing::Color::Magenta;
this->toolStripButton1->Name = L"toolStripButton1";
this->toolStripButton1->Size = System::Drawing::Size(21, 20);
this->toolStripButton1->Text = L"Export AutoCad Data";
this->toolStripButton1->Click += gcnew System::EventHandler(this, &Form1::
exportAutoCadDataToolStripMenuItem_Click);
//
// toolTip1
//
this->toolTip1->Popup += gcnew System::Windows::Forms::PopupEventHandler(this, &
Form1::toolTip1_Popup);
//
// saveFileDialog2
//
this->saveFileDialog2->Filter = L"DXF Files (.DXF)|*.DXF";
//
// Form1
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->BackColor = System::Drawing::SystemColors::ControlText;
this->ClientSize = System::Drawing::Size(944, 600);
this->Controls->Add(this->toolStrip1);
this->Controls->Add(this->tabControl1);
this->Controls->Add(this->menuStrip1);
this->Icon = (cli::safe_cast<System::Drawing::Icon^ >(resources->GetObject(L"$
this.Icon")));
this->MainMenuStrip = this->menuStrip1;
this->MaximizeBox = false;
this->Name = L"Form1";
this->Text = L"rPORT";
this->Load += gcnew System::EventHandler(this, &Form1::Form1_Load);
this->menuStrip1->ResumeLayout(false);
this->menuStrip1->PerformLayout();
this->tabPage2->ResumeLayout(false);
this->tabPage2->PerformLayout();
this->groupBox5->ResumeLayout(false);
this->groupBox5->PerformLayout();
this->groupBox3->ResumeLayout(false);
this->groupBox3->PerformLayout();
this->tabPage1->ResumeLayout(false);
this->tabPage1->PerformLayout();
this->groupBox1->ResumeLayout(false);
this->groupBox1->PerformLayout();
this->groupBox2->ResumeLayout(false);
this->groupBox2->PerformLayout();
this->tabControl1->ResumeLayout(false);
this->tabPage4->ResumeLayout(false);
this->groupBox4->ResumeLayout(false);
this->groupBox4->PerformLayout();
this->tabPage5->ResumeLayout(false);
this->groupBox6->ResumeLayout(false);
this->groupBox6->PerformLayout();
this->tabPage6->ResumeLayout(false);
this->groupBox7->ResumeLayout(false);
this->groupBox7->PerformLayout();
this->tabPage7->ResumeLayout(false);

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

40

```

        this->groupBox8->ResumeLayout(false);
        this->groupBox8->PerformLayout();
        this->toolStrip1->ResumeLayout(false);
        this->toolStrip1->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();
    }

#pragma endregion

private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
}

private: System::Void comboBox1_SelectedIndexChanged(
    System::Object^ sender, System::EventArgs^ e)
{
    asetType(comboBox1->SelectedIndex);
}

private: System::Void comboBox2_SelectedIndexChanged(
    System::Object^ sender, System::EventArgs^ e)
{
    asetCode(comboBox2->SelectedIndex);
}

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    button2->Visible = true ;
    button7->Visible = true ;
    asetDimensions(aPortCode);
    textBox13->Text = System::Convert::ToString (aAP_Xthreshold)+ " m" ;
    textBox14->Text = System::Convert::ToString (aAP_Ythreshold)+ " m" ;
    textBox15->Text = System::Convert::ToString (aAP_l1) + " m" ;
    textBox16->Text = System::Convert::ToString (aAP_s1) ;
    textBox17->Text = System::Convert::ToString (aAP_l2)+ " m" ;
    textBox18->Text = System::Convert::ToString (aAP_s2) ;
    textBox19->Text = System::Convert::ToString (aAP_l3)+ " m" ;
    textBox20->Text = System::Convert::ToString (aAP_s3) ;
    textBox21->Text = System::Convert::ToString (aAP_divergence) ;
    textBox22->Text = System::Convert::ToString (aRin)+ " m" ;
    textBox23->Text = System::Convert::ToString (aRout)+ " m" ;
    textBox24->Text = System::Convert::ToString (aTO_X)+ " m" ;
    textBox25->Text = System::Convert::ToString (aTO_Y)+ " m" ;
    textBox26->Text = System::Convert::ToString (aTO_l1)+ " m" ;
    textBox27->Text = System::Convert::ToString (aTO_l2)+ " m" ;
    textBox28->Text = System::Convert::ToString (aTO_s) ;
    textBox29->Text = System::Convert::ToString (aTO_divergence);
    textBox30->Text = System::Convert::ToString (aSezorX)+ " m" ;
    textBox31->Text = System::Convert::ToString (aSezorY)+ " m" ;
    textBox32->Text = "45/" +System::Convert::ToString (aTR_s)+ " m" ;

    acreateSurfaces() ;

    asetScout();

    //CHECKING THE RESULTS
    /*textBox75->Text = System::Convert::ToString(aapproach[1].x);
    textBox76->Text = System::Convert::ToString(aapproach[1].y);
    textBox77->Text = System::Convert::ToString(aapproach[2].x);
    textBox78->Text = System::Convert::ToString(aapproach[2].y);
    textBox79->Text = System::Convert::ToString(aapproach[3].x);
    textBox80->Text = System::Convert::ToString(aapproach[3].y);
    textBox81->Text = System::Convert::ToString(aapproach[4].x);
    textBox82->Text = System::Convert::ToString(aapproach[4].y);

    textBox83->Text = System::Convert::ToString(atakeoff[1].x);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

41

```

        textBox84->Text = System::Convert::ToString(atakeoff[1].y);
        textBox85->Text = System::Convert::ToString(atakeoff[2].x);
        textBox86->Text = System::Convert::ToString(atakeoff[2].y);
        textBox87->Text = System::Convert::ToString(atakeoff[3].x);
        textBox88->Text = System::Convert::ToString(atakeoff[3].y);
        textBox89->Text = System::Convert::ToString(atakeoff[4].x);
        textBox90->Text = System::Convert::ToString(atakeoff[4].y);
        textBox91->Text = System::Convert::ToString(atakeoff[5].x);
        textBox92->Text = System::Convert::ToString(atakeoff[5].y);
        textBox93->Text = System::Convert::ToString(atakeoff[6].x);
        textBox94->Text = System::Convert::ToString(atakeoff[6].y);

        textBox95->Text = System::Convert::ToString(a-transitional[1].x);
        textBox96->Text = System::Convert::ToString(a-transitional[1].y);
        textBox97->Text = System::Convert::ToString(a-transitional[2].x);
        textBox98->Text = System::Convert::ToString(a-transitional[2].y);
        textBox99->Text = System::Convert::ToString(a-transitional[3].x);
        textBox100->Text = System::Convert::ToString(a-transitional[3].y);
        textBox101->Text = System::Convert::ToString(a-transitional[4].x);
        textBox102->Text = System::Convert::ToString(a-transitional[4].y);

        textBox103->Text = System::Convert::ToString(asafetyzone[1].x);
        textBox104->Text = System::Convert::ToString(asafetyzone[1].y);
        textBox105->Text = System::Convert::ToString(asafetyzone[2].x);
        textBox106->Text = System::Convert::ToString(asafetyzone[2].y);
        textBox107->Text = System::Convert::ToString(asafetyzone[3].x);
        textBox108->Text = System::Convert::ToString(asafetyzone[3].y);
        textBox109->Text = System::Convert::ToString(asafetyzone[4].x);
        textBox110->Text = System::Convert::ToString(asafetyzone[4].y);

        textBox147->Text = System::Convert::ToString(aRW_Mbar.x);
        textBox148->Text = System::Convert::ToString(aRW_Mbar.y);
        textBox149->Text = System::Convert::ToString(aScoutbar.x);
        textBox150->Text = System::Convert::ToString(aScoutbar.y);
        textBox151->Text = System::Convert::ToString(aScout2.x);
        textBox152->Text = System::Convert::ToString(aScout2.y);
        textBox158->Text = System::Convert::ToString(aScout.x);
        textBox157->Text = System::Convert::ToString(aScout.y);
        textBox156->Text = System::Convert::ToString(aScoutl80.x);
        textBox155->Text = System::Convert::ToString(aScoutl80.y);

        textBox154->Text = System::Convert::ToString(aAzim) ;
        textBox153->Text = System::Convert::ToString(afaxis) ; */
    }

private: System::Void textBox6_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    aRW_lenght = System::Convert::ToDouble(textBox6->Text);
    aRW_AP.x = -(aRW_lenght/2);
    aRW_TO.x = (aRW_lenght/2);
    aRW_AP.y = aRW_TO.y = 0 ;
}

private: System::Void textBox3_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    aRW_M.x = aRW_M2.x = 0 ;
    aRW_Mbar.x = System::Convert::ToDouble(textBox3->Text) ;
}

private: System::Void textBox4_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    aRW_M.y = aRW_M2.y = 0 ;
    aRW_Mbar.y = System::Convert::ToDouble(textBox4->Text) ;
}

private: System::Void textBox5_TextChanged(System::Object^ sender, System::EventArgs^ e)
{

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

42

```

        aRW_Mbar.z = System::Convert::ToDouble(textBox5->Text) ;
        aRW_M.z = aRW_M2.z = aRW_Mbar.z ;
    }

private: System::Void textBox1_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    aScoutbar.x = System::Convert::ToDouble(textBox1->Text) ;
    bScoutbar.x = System::Convert::ToDouble(textBox1->Text) ;
}

private: System::Void textBox2_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    aScoutbar.y = System::Convert::ToDouble(textBox2->Text) ;
    bScoutbar.y = System::Convert::ToDouble(textBox2->Text) ;
}

private: System::Void textBox7_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    aAzim = - (System::Convert::ToDouble(textBox7->Text)) ;
    afixis = -(aAzim+90);
}

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
{
    button3->Visible = true;
    //ACTUAL INCLUSION COMPUTATION
    aisINapproach = inPolygon(aScout, aapproach, 4) ;
    aisINTakeoff = inPolygon(aScout, atakeoff, 6) ;
    aisINsafetyzone = inPolygon(aScout, asafetyzone, 4) ;
    transitional(aScout, atransitional) ;
    cyclic(aScout, aRin, aRout) ;
    //DISPLAY OF THE RESULTS
    textBox33->Text=System::Convert::ToString(aisINapproach);
    textBox34->Text=System::Convert::ToString(aisINTakeoff);
    textBox35->Text=System::Convert::ToString(aisINsafetyzone);
    textBox36->Text=System::Convert::ToString(aisINtransitional);
    textBox37->Text=System::Convert::ToString(aisINinner);
    textBox38->Text=System::Convert::ToString(aisINouter);
    textBox39->Text=System::Convert::ToString(aisINconical);
}

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e)
{
    aScout.z = 100000 ;
    //Now computing the actual altitude
    acomputeHeight(aScout);
    aScout.z = aRW_Mbar.z + aScout.z ;
    textBox44->Text = System::Convert::ToString(aScout.z) ;

    textBox46->Text = " ";
    char * c;
    for(c = aScout.description;*c;c++)
        textBox46->Text += System::Char::ToString(*c) ;
}

private: System::Void button7_Click(System::Object^ sender, System::EventArgs^ e)
{
    button8->Visible = true;
    //ACTUAL INCLUSION COMPUTATION
    aisINapproach = inPolygon(aScout180, aapproach, 4) ;
    aisINTakeoff = inPolygon(aScout180, atakeoff, 6) ;
    aisINsafetyzone = inPolygon(aScout180, asafetyzone, 4) ;
    transitional(aScout180, atransitional) ;
    cyclic(aScout180, aRin, aRout) ;
    //DISPLAY OF THE RESULTS
    textBox33->Text=System::Convert::ToString(aisINapproach);
    textBox34->Text=System::Convert::ToString(aisINTakeoff);
    textBox35->Text=System::Convert::ToString(aisINsafetyzone);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

43

```

        textBox36->Text=System::Convert::ToString(aisINtransitional);
        textBox37->Text=System::Convert::ToString(aisINinner);
        textBox38->Text=System::Convert::ToString(aisINouter);
        textBox39->Text=System::Convert::ToString(aisINconical);
    }

private: System::Void button8_Click(System::Object^ sender, System::EventArgs^ e)
{
    aScout180.z = 100000 ;
    //Now computing the actual altitude
    acomputeHeight(aScout180);
    aScout180.z = aRW_Mbar.z + aScout180.z ;
    textBox45->Text = System::Convert::ToString(aScout180.z) ;

    textBox47->Text = "";
    char * c;
    for(c = aScout180.description;*c;c++)
        textBox47->Text += System::Char::ToString(*c) ;
}

private: System::Void saveToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e)
{
    saveFileDialog1->Title = "Save rPORT output as text..." ;

    saveFileDialog1->ShowDialog();
    System::String ^ myname = saveFileDialog1->FileName ;

    char* username = (char*)(void*)Marshal::StringToHGlobalAnsi(myname);

    strcat(username, ".txt");

    saveVariables(username);
    //saveVariables("data.txt");
}

private: System::Void openToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e)
{
}

private: System::Void comboBox3_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e)
{
    bsetType(comboBox3->SelectedIndex);
}

private: System::Void comboBox4_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e)
{
    bsetCode(comboBox4->SelectedIndex);
    //to check the assignement
    //textBox8->Text = "" + bPortCode ;
}

private: System::Void textBox8_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    bRW_M.x = bRW_M2.x = 0 ;
    bRW_Mbar.x = System::Convert::ToDouble(textBox8->Text) ;
}

private: System::Void textBox9_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    bRW_M.y = bRW_M2.y = 0 ;
    bRW_Mbar.y = System::Convert::ToDouble(textBox9->Text) ;
}

private: System::Void textBox10_TextChanged(System::Object^ sender, System::EventArgs^ e)
{

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

44

```

        bRW_Mbar.z = System::Convert::ToDouble(textBox10->Text) ;
        bRW_M.z = bRW_M2.z = bRW_Mbar.z ;
    }

private: System::Void textBox11_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    bRW_lenght = System::Convert::ToDouble(textBox11->Text);
    bRW_AP.x = -(bRW_lenght/2);
    bRW_TO.x = (bRW_lenght/2);
    bRW_AP.y = bRW_TO.y = 0 ;
}

private: System::Void textBox12_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    bAzim = - (System::Convert::ToDouble(textBox12->Text)) ;
    bfxaxis = -(bAzim+90);
}

private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e)
{
    button5->Visible = true;
    button9->Visible = true;
    bsetDimensions(bPortCode);
    textBox48->Text = System::Convert::ToString (bAP_Xthreshold) + " m";
    textBox49->Text = System::Convert::ToString (bAP_Ythreshold)+ " m" ;
    textBox50->Text = System::Convert::ToString (bAP_l1) + " m";
    textBox51->Text = System::Convert::ToString (bAP_s1) ;
    textBox52->Text = System::Convert::ToString (bAP_l2) + " m";
    textBox53->Text = System::Convert::ToString (bAP_s2) ;
    textBox54->Text = System::Convert::ToString (bAP_l3) + " m";
    textBox55->Text = System::Convert::ToString (bAP_s3) ;
    textBox56->Text = System::Convert::ToString (bAP_divergence) ;
    textBox57->Text = System::Convert::ToString (bRin) + " m";
    textBox58->Text = System::Convert::ToString (bRout) + " m";
    textBox59->Text = System::Convert::ToString (bTO_X) + " m";
    textBox60->Text = System::Convert::ToString (bTO_Y) + " m";
    textBox61->Text = System::Convert::ToString (bTO_l1)+ " m" ;
    textBox62->Text = System::Convert::ToString (bTO_l2)+ " m" ;
    textBox63->Text = System::Convert::ToString (bTO_s) ;
    textBox64->Text = System::Convert::ToString (bTO_divergence) ;
    textBox65->Text = System::Convert::ToString (bSezorX)+ " m" ;
    textBox66->Text = System::Convert::ToString (bSezorY)+ " m" ;
    textBox67->Text = "45/" +System::Convert::ToString (bTR_s)+ " m" ;

    bcreateSurfaces() ;

    bsetScout();

    //CHECKING THE RESULTS
    /* textBox111->Text = System::Convert::ToString(bapproach[1].x);
    textBox112->Text = System::Convert::ToString(bapproach[1].y);
    textBox113->Text = System::Convert::ToString(bapproach[2].x);
    textBox114->Text = System::Convert::ToString(bapproach[2].y);
    textBox115->Text = System::Convert::ToString(bapproach[3].x);
    textBox116->Text = System::Convert::ToString(bapproach[3].y);
    textBox117->Text = System::Convert::ToString(bapproach[4].x);
    textBox118->Text = System::Convert::ToString(bapproach[4].y);

    textBox119->Text = System::Convert::ToString(btakeoff[1].x);
    textBox120->Text = System::Convert::ToString(btakeoff[1].y);
    textBox121->Text = System::Convert::ToString(btakeoff[2].x);
    textBox122->Text = System::Convert::ToString(btakeoff[2].y);
    textBox123->Text = System::Convert::ToString(btakeoff[3].x);
    textBox124->Text = System::Convert::ToString(btakeoff[3].y);
    textBox125->Text = System::Convert::ToString(btakeoff[4].x);
    textBox126->Text = System::Convert::ToString(btakeoff[4].y);
    textBox127->Text = System::Convert::ToString(btakeoff[5].x);
    textBox128->Text = System::Convert::ToString(btakeoff[5].y);

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

45

```

        textBox129->Text = System::Convert::ToString(btakeoff[6].x);
        textBox130->Text = System::Convert::ToString(btakeoff[6].y);

        textBox131->Text = System::Convert::ToString(btransitional[1].x);
        textBox132->Text = System::Convert::ToString(btransitional[1].y);
        textBox133->Text = System::Convert::ToString(btransitional[2].x);
        textBox134->Text = System::Convert::ToString(btransitional[2].y);
        textBox135->Text = System::Convert::ToString(btransitional[3].x);
        textBox136->Text = System::Convert::ToString(btransitional[3].y);
        textBox137->Text = System::Convert::ToString(btransitional[4].x);
        textBox138->Text = System::Convert::ToString(btransitional[4].y);

        textBox139->Text = System::Convert::ToString(bsafetyzone[1].x);
        textBox140->Text = System::Convert::ToString(bsafetyzone[1].y);
        textBox141->Text = System::Convert::ToString(bsafetyzone[2].x);
        textBox142->Text = System::Convert::ToString(bsafetyzone[2].y);
        textBox143->Text = System::Convert::ToString(bsafetyzone[3].x);
        textBox144->Text = System::Convert::ToString(bsafetyzone[3].y);
        textBox145->Text = System::Convert::ToString(bsafetyzone[4].x);
        textBox146->Text = System::Convert::ToString(bsafetyzone[4].y);*/
    }

private: System::Void button5_Click(System::Object^ sender, System::EventArgs^ e)
{
    button6->Visible = true;
    //ACTUAL INCLUSION COMPUTATION
    bisINapproach = inPolygon(bScout, bapproach, 4) ;
    bisINTakeoff = inPolygon(bScout, btakeoff, 6) ;
    bisINsafetyzone = inPolygon(bScout, bsafetyzone, 4) ;
    transitionalB(bScout, btransitional) ;
    cyclicB(bScout, bRin, bRout) ;
    //DISPLAY OF THE RESULTS
    textBox74->Text=System::Convert::ToString(bisINapproach);
    textBox73->Text=System::Convert::ToString(bisINTakeoff);
    textBox72->Text=System::Convert::ToString(bisINsafetyzone);
    textBox71->Text=System::Convert::ToString(bisINtransitional);
    textBox70->Text=System::Convert::ToString(bisINinner);
    textBox69->Text=System::Convert::ToString(bisINouter);
    textBox68->Text=System::Convert::ToString(bisINconical);
}

private: System::Void button6_Click(System::Object^ sender, System::EventArgs^ e)
{
    bScout.z = 100000 ;
    //Now computing the actual altitude
    bcomputeHeight(bScout);
    bScout.z = bRW_Mbar.z + bScout.z ;
    textBox43->Text = System::Convert::ToString(bScout.z) ;

    textBox41->Text = "";
    char * c;
    for(c = bScout.description;*c;c++)
        textBox41->Text += System::Char::ToString(*c) ;
}

private: System::Void button9_Click(System::Object^ sender, System::EventArgs^ e)
{
    button10->Visible = true ;
    //ACTUAL INCLUSION COMPUTATION
    bisINapproach = inPolygon(bScout180, bapproach, 4) ;
    bisINTakeoff = inPolygon(bScout180, btakeoff, 6) ;
    bisINsafetyzone = inPolygon(bScout180, bsafetyzone, 4) ;
    transitionalB(bScout180, btransitional) ;
    cyclicB(bScout180, bRin, bRout) ;
    //DISPLAY OF THE RESULTS
    textBox74->Text=System::Convert::ToString(bisINapproach);
    textBox73->Text=System::Convert::ToString(bisINTakeoff);
    textBox72->Text=System::Convert::ToString(bisINsafetyzone);
    textBox71->Text=System::Convert::ToString(bisINtransitional);
}

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

46

```

        textBox70->Text=System::Convert::ToString(bisINinner);
        textBox69->Text=System::Convert::ToString(bisINouter);
        textBox68->Text=System::Convert::ToString(bisINconical);
    }

private: System::Void button10_Click(System::Object^ sender, System::EventArgs^ e)
{
    bScout180.z = 100000 ;
    //Now computing the actual altitude
    bcomputeHeight(bScout180);
    bScout180.z = bRW_Mbar.z + bScout180.z ;
    textBox42->Text = System::Convert::ToString(bScout180.z) ;

    textBox40->Text = " ";
    char * c;
    for(c = bScout180.description;*c;c++)
        textBox40->Text += System::Char::ToString(*c) ;
}

private: System::Void textBox1_MouseHover(System::Object^ sender, System::EventArgs^ e)
{
}

private: System::Void saveFileDialog1_FileOk(System::Object^ sender, System::ComponentModel::CancelEventArgs^ e)
{
}

private: System::Void button11_Click(System::Object^ sender, System::EventArgs^ e)
{
    NikosAthanasίου = finalAltitude(aScout,aScout180,bScout,bScout180) ;
    textBox159->Text = System::Convert::ToString(NikosAthanasίου) ;
}

private: System::Void newToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e)
{
    textBox1->Text="0";
    textBox2->Text="0";
    textBox3->Text="0";
    textBox4->Text="0";
    textBox5->Text="0";
    textBox6->Text="0";
    textBox7->Text="0";
    textBox8->Text="0";
    textBox9->Text="0";
    textBox10->Text="0";
    textBox11->Text="0";
    textBox12->Text="0";

}

private: System::Void toolTip1_Popup(System::Object^ sender, System::Windows::Forms::PopupEventArgs^ e)
{
    // toolTip1->ToolTipTitle = "Proper Data Input";
}

private: System::Void textBox1_MouseHover_1(System::Object^ sender, System::EventArgs^ e)
{
    toolTip1->ToolTipTitle = "Proper Data Input";
    toolTip1->Show(
        "Use the comma character for decimal digits and insert (if needed) the negative symbol",
        last",textBox1);
}

private: System::Void textBox2_MouseHover(System::Object^ sender, System::EventArgs^ e)

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

47

```

    {
        toolTip1->ToolTipTitle = "Proper Data Input";
        toolTip1->Show(
            "Use the comma character for decimal digits and insert (if needed) the negative symbol
last",textBox2);
    }
private: System::Void textBox7_MouseHover(System::Object^ sender, System::EventArgs^ e)
    {
        toolTip1->ToolTipTitle = "Proper Data Input";
        toolTip1->Show(
            "Azimuth is in counter-clockwise degrees",textBox7);
    }
private: System::Void textBox12_MouseHover(System::Object^ sender, System::EventArgs^ e)
    {
        toolTip1->ToolTipTitle = "Proper Data Input";
        toolTip1->Show(
            "Azimuth is in counter-clockwise degrees",textBox12);
    }

private: System::Void button1_MouseHover(System::Object^ sender, System::EventArgs^ e)
    {
        toolTip1->ToolTipTitle = "COMPUTATION STEP 1";
        toolTip1->Show(
            "Click to set the dimensions of the obstacle limitation surfaces",button1);
    }
private: System::Void button2_MouseHover(System::Object^ sender, System::EventArgs^ e)
    {
        toolTip1->ToolTipTitle = "COMPUTATION STEP 2A-DETERMINE IN WHICH SURFACE THE
POINT IS INCLUDED";
        toolTip1->Show(
            "Click if approach & take-off surfaces are set from right to left in the local
coordinate system",button2);
    }
private: System::Void button3_MouseHover(System::Object^ sender, System::EventArgs^ e)
    {
        toolTip1->ToolTipTitle = "COMPUTATION STEP 3A-DETERMINE ALLOABLE HEIGHT";
        toolTip1->Show(
            "Calculation is done for the same runway direction as in step 2A",button3);
    }
private: System::Void button7_MouseHover(System::Object^ sender, System::EventArgs^ e)
    {
        toolTip1->ToolTipTitle = "COMPUTATION STEP 2B-DETERMINE IN WHICH SURFACE THE
POINT IS INCLUDED";
        toolTip1->Show(
            "Click if approach & take-off surfaces are set from left to right in the local
coordinate system",button7);
    }
private: System::Void button8_MouseHover(System::Object^ sender, System::EventArgs^ e)
    {
        toolTip1->ToolTipTitle = "COMPUTATION STEP 3B-DETERMINE ALLOABLE HEIGHT";
        toolTip1->Show(
            "Calculation is done for the same runway direction as in step 2B",button8);
    }
private: System::Void exportAutoCadDataToolStripMenuItem_Click(System::Object^ sender, System
::EventArgs^ e)
    {
        saveFileDialog2->Title = "Save rPORT output as AutoCad file..." ;

        saveFileDialog2->ShowDialog();
        System::String ^ myname = saveFileDialog2->FileName ;

        char* CADname = (char*)(void*)Marshal::StringToHGlobalAnsi(myname);

        strcat(CADname, ".DXF");

        global(atransitionalbar,atransitional,afaxis,aRW_Mbar,6);
        global(asafetyzonebar,asafetyzone,afaxis,aRW_Mbar,6);
        global(atakeoffbar,atakeoff,afaxis,aRW_Mbar,8);
        global(aapproachbar, aapproach,afaxis,aRW_Mbar,6);
    }

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\Form1.h

48

```
    global(btransitionalbar,btransitional,bfaxis,bRW_Mbar,6);
    global(bsafetyzonebar,bsafetyzone,bfaxis,bRW_Mbar,6);
    global(btakeoffbar,btakeoff,bfaxis,bRW_Mbar,8);
    global(bapproachbar, bapproach,bfaxis,bRW_Mbar,6);

    exportDXF(CADname);

}

};

}
```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\myData.h

1

```

//CUSTOM DATA TYPES
struct mypoint {
    double x,y,z ;
    char *description; //Pointers are fun to use 147
};

struct myline {
    mypoint p1, p2;
};

extern double NikosAthanasious;

//
//-----//
//RUNWAY 1

//***** BASIC FUNCTIONS *****/
//GEOMETRIC ALGORITHMS
int intersect(myline la, myline lb);
int ccw(mypoint po, mypoint pa, mypoint pb);
// *****THE WAY IT WORKS*****
//checking for point t, in the polygon p that has N vertices
int inside(mypoint t, mypoint *p, int N);
int outline(mypoint t, mypoint *p, int N);

void transitional(mypoint a, mypoint *p);
void cyclic(mypoint t, int rin, int rout) ;

void acomputeHeight(mypoint &H) ;

void global (mypoint *pbar, mypoint *p, double angle, mypoint RWCbar, int N);
void global_reverse (mypoint *pbar, mypoint *p, double angle, mypoint RWCbar, int N);

//OTHERS
void asetDimensions(int a);
void acreateSurfaces();
void asetScout();
double finalAltitude(mypoint A, mypoint B, mypoint C, mypoint D);
void saveVariables(char *filename);
void exportDXF(char *filename);

//RUNDOWN FUNCTIONS
bool inPolygon(mypoint t, mypoint *p, int N);

//PROGRAM CONTROL FUNCTIONS
void asetType(int a);
void asetCode(int a);

//***** VARIABLES - RUNWAY 1 *****/
//PROGRAM CONTROL VARIABLES
extern bool aisINapproach, aisINtakeoff, aisINSafetyzone, aisINtransitional,
aisINconical, aisINinner, aisINouter ;
extern int atype, acode, aPortCode;

//USER DEFINED VARIABLES
extern double arW_lenght, aAzim, aFaxis;
extern mypoint arW_M, arW_M2, arW_Mbar ;//only arW_Mbar actually defined by the GUI
extern mypoint arW_AP, arW_AP2, arW_APbar ;
extern mypoint arW_TO, arW_TO2, arW_TObar ;

```


c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\myData.h

2

```

extern mypoint aScout, aScout2, aScoutbar, aScout180 ;//only aRW_Mbar actually defined by the
GUI

//DEPENDENT VARIABLES
//*****APPROACH SURFACE - runway 1
extern int aAP_Xthreshold, aAP_Ythreshold, aAP_l1, aAP_l2, aAP_l3 ;
extern double aAP_s1, aAP_s2, aAP_s3, aAP_divergence ;
extern mypoint approach[6]; //actual surface representation
extern mypoint approachbar[6]; //global system

//*****CONICAL SURFACE - runway 1
extern int aRin, aRout ;

//*****TAKE OFF SURFACE - runway 1
extern int aTO_X, aTO_Y, aTO_l1, aTO_l2 ;
extern double aTO_s, aTO_divergence;
extern mypoint atakeoff[8]; //actual surface representation
extern mypoint atakeoffbar[8];

//*****SAFETY ZONE
extern int aSezorX, aSezorY ;
extern mypoint asafetyzone[6]; //actual surface representation
extern mypoint asafetyzonebar[6];

//*****TRANSITIONAL SURFACE
extern double aTR_s;
extern mypoint atransitional[6]; //actual surface representation
extern mypoint atransitionalbar[6];

//
//-----//
//RUNWAY 2

//***** BASIC FUNCTIONS *****/

//GEOMETRIC ALGORITHMS

void transitionalB(mypoint a, mypoint *p);
void cyclicB(mypoint t, int rin, int rout) ;
void bcomputeHeight(mypoint &H) ;

//OTHERS
void bsetDimensions(int a);
void bcreateSurfaces();
void bsetScout();

//RUNDOWN FUNCTIONS

//PROGRAM CONTROL FUNCTIONS
void bsetType(int a);
void bsetCode(int a);

////////// VARIABLES - RUNWAY 2 //////////

//PROGRAM CONTROL VARIABLES
extern bool bisINapproach, bisINtakeoff, bisINSafetyzone, bisINtransitional,
bisINconical, bisINinner, bisINouter ;
extern int btype, bcode, bPortCode;

//USER DEFINED VARIABLES
extern double bRW_lenght, bAzim, bfxaxis;
extern mypoint bRW_M, bRW_M2, bRW_Mbar ;//only aRW_Mbar actually defined by the GUI
extern mypoint bRW_AP, bRW_AP2, bRW_APbar ;
extern mypoint bRW_TO, bRW_TO2, bRW_TObar ;

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\myData.h

3

```
extern mypoint bScout, bScout2, bScoutbar, bScout180 ;//only bScoutbar actually defined by the GUI

//DEPENDENT VARIABLES

//*****APPROACH SURFACE - runway 2
extern int bAP_Xthreshold, bAP_Ythreshold, bAP_l1, bAP_l2, bAP_l3 ;
extern double bAP_s1, bAP_s2, bAP_s3, bAP_divergence ;
extern mypoint bapproach[6]; //actual surface representation
extern mypoint bapproachbar[6];

//*****CONICAL SURFACE - runway 2
extern int bRin, bRout ;

//*****TAKE OFF SURFACE - runway 2
extern int bTO_X, bTO_Y, bTO_l1, bTO_l2 ;
extern double bTO_s, bTO_divergence;
extern mypoint btakeoff[8]; //actual surface representation
extern mypoint btakeoffbar[8];

//*****SAFETY ZONE - runway 2
extern int bSezorX, bSezorY ;
extern mypoint bsafetyzone[6]; //actual surface representation
extern mypoint bsafetyzonebar[6];

//*****TRANSITIONAL SURFACE - runway 2
extern double bTR_s;
extern mypoint btransitional[6]; //actual surface representation
extern mypoint btransitionalbar[6];
```

c:\Documents and Settings\Nick\My Documents\... Studio 2008\Projects\rPORT\rPORT\resource.h 1

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by app.rc
//
#define IDI_ICON1 101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 102
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1000
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif
```

```
c:\Documents and Settings\Nick\My Documents\Visual Studio 2008\Projects\rPORT\rPORT\stdafx.h_1
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
#pragma once

// TODO: reference additional headers your program requires here
```

c:\Documents and Settings\Nick\My Documents\... 2008\Projects\rPORT\rPORT\AssemblyInfo.cpp 1

```
#include "stdafx.h"

using namespace System;
using namespace System::Reflection;
using namespace System::Runtime::CompilerServices;
using namespace System::Runtime::InteropServices;
using namespace System::Security::Permissions;

//
// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
//
[assembly:AssemblyTitleAttribute("rPORT")];
[assembly:AssemblyDescriptionAttribute("")];
[assembly:AssemblyConfigurationAttribute("")];
[assembly:AssemblyCompanyAttribute("F")];
[assembly:AssemblyProductAttribute("rPORT")];
[assembly:AssemblyCopyrightAttribute("Copyright (c) F 2009")];
[assembly:AssemblyTrademarkAttribute("")];
[assembly:AssemblyCultureAttribute("")];

//
// Version information for an assembly consists of the following four values:
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
//
// You can specify all the value or you can default the Revision and Build Numbers
// by using the '*' as shown below:

[assembly:AssemblyVersionAttribute("1.0.*")];

[assembly:ComVisible(false)];

[assembly:CLSCompliantAttribute(true)];

[assembly:SecurityPermission(SecurityAction::RequestMinimum, UnmanagedCode = true)];
```

c:\Documents and Settings\Nick\My Documents\... 2008\Projects\rPORT\rPORT\computeHeight.cpp 1

```
#include "stdafx.h"
#include "myData.h"
#include <math.h>

double NikosAthanasίου;

extern int aAP_Xthreshold, aAP_Ythreshold, aAP_l1, aAP_l2, aAP_l3 ;
extern double aAP_s1, aAP_s2, aAP_s3, aAP_divergence ;
extern int aRin, aRout ;
extern int aTO_X, aTO_Y, aTO_l1, aTO_l2 ;
extern double aTO_s, aTO_divergence;
extern int aSezorX, aSezorY ;
extern double aTR_s;

extern bool aisINapproach, aisINtakeoff, aisINSafetyzone, aisINtransitional,
aisINconical, aisINinner, aisINouter ;

void acomputeHeight(mypoint &H)
{
    double distance = (H.x*H.x)+(H.y*H.y) ;
    distance = sqrt(distance) ;
    double tempHeight, s=0.05 ;

    if(aisINSafetyzone){
        H.z = 0; H.description = "SAFETY ZONE";
        return ;}

    if(aisINinner){
        H.z = 45 ; H.description="INNER HORIZONTAL" ;}

    if(aisINouter){
        tempHeight = 45 + s*(-aRin+aRout);
        if(tempHeight<H.z){H.z=tempHeight; H.description="OUTER HORIZONTAL";}
    }

    if(aisINconical){
        tempHeight = 45 + s*(distance-aRin);
        if(tempHeight<H.z){H.z=tempHeight; H.description="CONICAL";}
    }

    if(aisINapproach){
        double tempX = -H.x ;
        if(tempX>=(aAP_Xthreshold+aAP_l1+aAP_l2)){tempHeight=(aAP_l1*aAP_s1)+(aAP_s2*aAP_l2);}
        else if(tempX>=(aAP_Xthreshold+aAP_l1)){tempHeight=(aAP_l1*aAP_s1)+(aAP_s2*(tempX-
aAP_l1-aAP_Xthreshold));}
        else tempHeight = aAP_s1*(tempX-aAP_Xthreshold);
        if(tempHeight<H.z){H.z=tempHeight; H.description="APPROACH";}
    }

    if(aisINtakeoff){
        tempHeight = (H.x-aTO_X)*aTO_s;
        if(tempHeight<H.z){H.z=tempHeight; H.description="TAKE OFF";}
    }

    if(aisINtransitional){
        double tempY = H.y ;
        if(H.y<0)tempY = -H.y ;
        tempHeight = (tempY - aSezorY)*aTR_s;
        if(tempHeight<H.z){H.z=tempHeight; H.description="TRANSITIONAL";}
    }
}

//          ////////////////////////////////////          RUNWAY 2

extern int bAP_Xthreshold, bAP_Ythreshold, bAP_l1, bAP_l2, bAP_l3 ;
extern double bAP_s1, bAP_s2, bAP_s3, bAP_divergence ;
```

c:\Documents and Settings\Nick\My Documents\... 2008\Projects\rPORT\rPORT\computeHeight.cpp 2

```

extern int bRin, bRout ;
extern int bTO_X, bTO_Y, bTO_l1, bTO_l2 ;
extern double bTO_s, bTO_divergence;
extern int bSezorX, bSezorY ;
extern double bTR_s;

extern bool bisINapproach, bisINTakeoff, bisINSafetyzone, bisINtransitional,
bisINconical, bisINinner, bisINouter ;

void bcomputeHeight(mypoint &H)
{
    double distance = (H.x*H.x)+(H.y*H.y) ;
    distance = sqrt(distance) ;
    double tempHeight, s=0.05 ;

    if(bisINSafetyzone){
        H.z = 0; H.description = "SAFETY ZONE";
        return ;}

    if(bisINinner){
        H.z = 45 ; H.description="INNER HORIZONTAL" ;}

    if(bisINouter){
        tempHeight = 45 + s*(-bRin+bRout);
        if(tempHeight<H.z){H.z=tempHeight; H.description="OUTER HORIZONTAL";}
    }

    if(bisINconical){
        tempHeight = 45 + s*(distance-bRin);
        if(tempHeight<H.z){H.z=tempHeight; H.description="CONICAL";}
    }

    if(bisINapproach){
        double tempX = -H.x ;
        if(tempX>=(bAP_Xthreshold+bAP_l1+bAP_l2)){tempHeight=(bAP_l1*bAP_s1)+(bAP_s2*bAP_l2);}
        else if(tempX>=(bAP_Xthreshold+bAP_l1)){tempHeight=(bAP_l1*bAP_s1)+(bAP_s2*(tempX-
bAP_l1-bAP_Xthreshold));}
        else tempHeight = bAP_s1*(tempX-bAP_Xthreshold);
        if(tempHeight<H.z){H.z=tempHeight; H.description="APPROACH";}
    }

    if(bisINTakeoff){
        tempHeight = (H.x-bTO_X)*bTO_s;
        if(tempHeight<H.z){H.z=tempHeight; H.description="TAKE OFF";}
    }

    if(bisINtransitional){
        double tempY = H.y ;
        if(H.y<0)tempY = -H.y ;
        tempHeight = (tempY - bSezorY)*bTR_s;
        if(tempHeight<H.z){H.z=tempHeight; H.description="TRANSITIONAL";}
    }
}

double finalAltitude(mypoint A, mypoint B, mypoint C, mypoint D)
{
    double tempHeight = 100000;

    if(A.z < tempHeight) tempHeight = A.z ;
    if(B.z < tempHeight) tempHeight = B.z ;
    if(C.z < tempHeight) tempHeight = C.z ;
    if(D.z < tempHeight) tempHeight = D.z ;

    return tempHeight ;
}

```

c:\Documents and Settings\Nick\My Documents\...\Projects\rPORT\rPORT\pointManipulation.cpp 1

```
#include "stdafx.h"
#include "myData.h"
#include <math.h>

mypoint aScout, aScout2, aScoutbar, aScout180 ;

extern mypoint aRW_M, aRW_M2, aRW_Mbar ;

double aAzim, afaixis ;

void asetScout()
{
    //Actual point manipulation algorithm
    double PI = 3.14159265 ;
    double a = -(afaixis*PI)/(180.0); //faixis angle local->2 so angle 2->local=-afaixis(radians)
    aScout2.x = aScoutbar.x - aRW_Mbar.x ;
    aScout2.y = aScoutbar.y - aRW_Mbar.y ;

    aScout.x = (aScout2.x*cos(a))+(aScout2.y*sin(a)) ;
    aScout.y = (aScout2.y*cos(a))-(aScout2.x*sin(a)) ;

    aScout180.x = -aScout.x ;
    aScout180.y = -aScout.y ;
}

mypoint bScout, bScout2, bScoutbar, bScout180 ;
extern mypoint bRW_M, bRW_M2, bRW_Mbar ;

double bAzim, bfaixis ;

void bsetScout()
{
    //Actual point manipulation algorithm
    double PI = 3.14159265 ;
    double a = -(bfaixis*PI/180); //faixis angle local->2 so angle 2->local=-afaixis(radians)
    bScout2.x = bScoutbar.x - bRW_Mbar.x ;
    bScout2.y = bScoutbar.y - bRW_Mbar.y ;

    bScout.x = (bScout2.x*cos(a))+(bScout2.y*sin(a)) ;
    bScout.y = (bScout2.y*cos(a))-(bScout2.x*sin(a)) ;

    bScout180.x = -bScout.x ;
    bScout180.y = -bScout.y ;
}
```


c:\Documents and Settings\Nick\My Documents\... 2008\Projects\rPORT\rPORT\programControl.cpp 1

```
#include "stdafx.h"
#include "myData.h"
```

```
int atype, acode, aPortCode ;
bool aisINapproach, aisINtakeoff, aisINsafetyzone, aisINtransitional,
aisINconical, aisINinner, aisINouter ;
```

```
void asetType(int a)
{
    atype = (a + 1)*100 ;
    aPortCode = atype+acode ;
}
```

```
void asetCode(int a)
{
    acode = (a + 1) ;
    aPortCode = atype+acode ;
}
```

```
int btype, bcode, bPortCode ;
bool bisINapproach, bisINtakeoff, bisINsafetyzone, bisINtransitional,
bisINconical, bisINinner, bisINouter ;
```

```
void bsetType(int a)
{
    btype = (a + 1)*100 ;
    bPortCode = btype+bcode ;
}
```

```
void bsetCode(int a)
{
    bcode = (a + 1) ;
    bPortCode = btype+bcode ;
}
```

c:\Documents and Settings\Nick\My Documents\Visual Studio 2008\Projects\rPORT\rPORT\rPORT.cpp 1

```
// rPORT.cpp : main project file.

#include "stdafx.h"
#include "Form1.h"

using namespace rPORT;

[STAThreadAttribute]
int main(array<System::String ^> ^args)
{
    // Enabling Windows XP visual effects before any controls are created
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);

    // Create the main window and run it
    Application::Run(gcnew Form1());

    return 0;
}
```

```
c:\Documents and Settings\Nick\My Documents\...Studio 2008\Projects\rPORT\rPORT\saveload.cpp 1
```

```
#include "stdafx.h"
#include "myData.h"
#include <stdio.h>
#include <fstream>

extern int atype, acode, btype, bcode ;
extern double aRW_lenght, aAzim,bRW_lenght, bAzim;
extern mypoint aScout,aScout180,bScout,bScout180;
extern double NikosAthanasiou;
extern mypoint aSout, bScout;
extern mypoint asafetyzone[6],aapproach[6],atakeoff[8],atransitional[6];
extern int aRin, aRout ;
extern mypoint bsafetyzone[6], bapproach[6], btakeoff[8], btransitional[6];
extern int bRin, bRout ;

void savePoint3(FILE * fout, char *desc, mypoint &point)
{
    fprintf(fout,"%s:  X = %lf    Y = %lf    Altitude = %lf\n",desc,point.x,point.y,point.z);
}

void savePoint2(FILE * fout, char *desc, mypoint &point) {

    //fprintf(fout,"Point(%s):  %lf %lf %lf\n",desc,point.x,point.y,point.z);

    fprintf(fout,"%s X=%lf        Y=%lf \n",desc,point.x,point.y);

}

void saveDouble(FILE * fout, char *desc, double x) {
    fprintf(fout,"%s : %lf\n",desc, x);
}

void saveVariables(char * filename) {

    FILE *fout = fopen(filename,"w");

    fprintf(fout,"
                                NATIONAL TECHNICAL
UNIVERSITY OF ATHENS\n");
    fprintf(fout,"
                                School of Civil Engineering - Department of
Transportation Planning and Engineering \n\n");
    fprintf(fout,"
                                ATHANASIOU NIKOLAOS cod
01102069 - DIPLOMATIKH ERGASIA\n\n");

    fprintf(fout,"
                                SURVEY OF ALLOABLE ALTITUDE DUE TO RESTRICTION BY THE
OBSTACLE LIMITATION SURFACES\n");
    fprintf(fout,"
                                \n\n\n");
    fprintf(fout,"
                                ***** DETERMINE
ALTITUDE FOR THE POINT *****\n");
    savePoint2(fout, "
                                ",
aSoutbar) ;

    fprintf(fout, "\n\n
                                The local coordinate system for each runway is set to its
centre with the X axis at the runways's axis\n\n\n");
    fprintf(fout, "\n
                                RUNWAY 1 - DATA\n");
    savePoint2(fout, "
                                POINT in local coordinate system is:",aSout);

    switch(atype){
        case 100:
            fprintf(fout,"
                                Approach Type -> Non instrument\n");
            break;
        case 200:
            fprintf(fout,"
                                Approach Type -> No presicion\n");
            break;
    }
}
```

c:\Documents and Settings\Nick\My Documents\...Studio 2008\Projects\rPORT\rPORT\saveload.cpp 2

```

        case 300:
            fprintf(fout, "          Approach Type -> Precision\n");
            break;
    }

    switch(acode){
        case 1:
            fprintf(fout, "          Code number -> 1\n");
            break;
        case 2:
            fprintf(fout, "          Code number -> 2\n");
            break;
        case 3:
            fprintf(fout, "          Code number -> 3\n");
            break;
        case 4:
            fprintf(fout, "          Code number -> 4\n");
            break;
    }
    savePoint3(fout, "          Runway Centre", aRW_Mbar);
    saveDouble(fout, "          Length(meters)", aRW_lenght);
    saveDouble(fout, "          Azimuth(counter-clockwise degrees)", -aAzim);

    fprintf(fout, "\n          RUNWAY 1 - RESULTS\n");

    if(aScout.z != 999999)
        fprintf(fout, "          1st alloable height = %lf\n", aScout.z);
    else fprintf(fout, "          1st alloable height = NOT SET\n");

    if(aScout180.z != 999999)
        fprintf(fout, "          2nd alloable height = %lf\n", aScout180.z);
    else fprintf(fout, "          2nd alloable height = NOT SET\n");

    fprintf(fout, "\n\n          RUNWAY 2 - DATA\n");
    savePoint2(fout, "          POINT in local coordinate system is:", bScout);

    switch(btype){
        case 100:
            fprintf(fout, "          Approach Type -> Non instrument\n");
            break;
        case 200:
            fprintf(fout, "          Approach Type -> No presicion\n");
            break;
        case 300:
            fprintf(fout, "          Approach Type -> Precision\n");
            break;
    }

    switch(bcode){
        case 1:
            fprintf(fout, "          Code number -> 1\n");
            break;
        case 2:
            fprintf(fout, "          Code number -> 2\n");
            break;
        case 3:
            fprintf(fout, "          Code number -> 3\n");
            break;
        case 4:
            fprintf(fout, "          Code number -> 4\n");
            break;
    }
    savePoint3(fout, "          Runway Centre", bRW_Mbar);
    saveDouble(fout, "          Length(meters)", bRW_lenght);
    saveDouble(fout, "          Azimuth(counter-clockwise degrees)", -bAzim);

    fprintf(fout, "\n          RUNWAY 2 - RESULTS\n");

    if(bScout.z != 999999)

```

```
c:\Documents and Settings\Nick\My Documents\...Studio 2008\Projects\rPORT\rPORT\saveload.cpp 3
```

```

fprintf(fout,"          1st alloable height = %lf\n",bScout.z);
else fprintf(fout,"          1st alloable height = NOT SET\n");

if(bScout180.z != 999999)
fprintf(fout,"          2nd alloable height = %lf\n",bScout180.z);
else fprintf(fout,"          2nd alloable height = NOT SET\n");

if(NikosAthanasiou != 999999)
fprintf(fout,"\\n\\n\\n          F I N A L A
L T I T U D E = %lf meters", NikosAthanasiou);
else fprintf(fout,"\\n\\n\\n\\n\\n\\n\\n
F I N A L A L T I T U D E = N O T S E T");

fprintf(fout,"\\n\\n\\n\\n\\n\\n\\n\\n\\n\\n          COORDINATES OF THE OBSTACLE LIMITATION
SURFACES IN EACH RUNWAY'S LOCAL SYSTEM");

fprintf(fout,"\\n\\n          RUNWAY 1");

fprintf(fout,"\\n          SAFETY ZONE\\n");
for(int i=1;i<=4;i++){
    savePoint2(fout,"          ", asafetyzone[i]);
}

fprintf(fout,"          APPROACH SURFACE\\n");
for(int i=1;i<=4;i++){
    savePoint2(fout,"          ", aapproach[i]);
}

fprintf(fout,"          TAKE OFF SURFACE\\n");
for(int i=1;i<=6;i++){
    savePoint2(fout,"          ", atakeoff[i]);
}

fprintf(fout,"          CONICAL SURFACE\\n");
fprintf(fout,"          Rin=%d          Rout=%d\\n",aRin,aRout);

fprintf(fout,"          TRANSITIONAL\\n");
for(int i=1;i<=4;i++){
    savePoint2(fout,"          ", atransitional[i]);
}

fprintf(fout,"\\n          RUNWAY 2");

fprintf(fout,"\\n          SAFETY ZONE\\n");
for(int i=1;i<=4;i++){
    savePoint2(fout,"          ", bsafetyzone[i]);
}

fprintf(fout,"          APPROACH SURFACE\\n");
for(int i=1;i<=4;i++){
    savePoint2(fout,"          ", bapproach[i]);
}

fprintf(fout,"          TAKE OFF SURFACE\\n");
for(int i=1;i<=6;i++){
    savePoint2(fout,"          ", btakeoff[i]);
}

fprintf(fout,"          CONICAL SURFACE\\n");
fprintf(fout,"          Rin=%d          Rout=%d\\n",bRin,bRout);

fprintf(fout,"          TRANSITIONAL\\n");
for(int i=1;i<=4;i++){
    savePoint2(fout,"          ", btransitional[i]);
}

```

c:\Documents and Settings\Nick\My Documents\...\Studio 2008\Projects\rPORT\rPORT\saveload.cpp 4

```
//savePoint(fout,"aScout",aScout);  
fclose(fout);  
}  
  
void loadPoint(FILE * fin, mypoint &point) {  
    fscanf(fin,"Point %lf %lf %lf\n",&point.x,&point.y,&point.z);  
}  
  
void loadVariables(char * filename) {  
    FILE *fin = fopen(filename,"r");  
    loadPoint(fin,aScout);  
    fclose(fin);  
}
```

c:\Documents and Settings\Nick\My Documents\...\Projects\rPORT\rPORT\spatialSearching.cpp 1

```
#include "stdafx.h"
#include <limits.h>
#include "myData.h"
#include <math.h>

extern bool aisINapproach, aisINtakeoff, aisINsafetyzone, aisINtransitional,
aisINconical, aisINinner, aisINouter ;
extern bool bisINapproach, bisINtakeoff, bisINsafetyzone, bisINtransitional,
bisINconical, bisINinner, bisINouter ;

int intersect(myline la, myline lb)
{
    return(
        ( ccw(la.p1, la.p2, lb.p1)*ccw(la.p1, la.p2, lb.p2))<=0)
        &&
        ( ccw(lb.p1, lb.p2, la.p1)*ccw(lb.p1, lb.p2, la.p2) )<=0)
    );
}

int ccw(mypoint po, mypoint pa, mypoint pb)
{
    double dx1, dx2, dy1, dy2 ;

    dx1 = pa.x - po.x ;
    dy1 = pa.y - po.y ;
    dx2 = pb.x - po.x ;
    dy2 = pb.y - po.y ;

    if (dx1*dy2 > dy1*dx2) return +1;
    if (dx1*dy2 < dy1*dx2) return -1;
    if ( (dx1*dx2 < 0) || (dy1*dy2<0) ) return -1;
    if ( (dx1*dx1 + dy1*dy1) < (dx2*dx2 + dy2*dy2) ) return +1 ;
    return 0;
}

int inside(mypoint t, mypoint *p, int N)
{
    int i, count = 0, j=0;
    myline lt, lp ;
    p[0] = p[N] ; p[N+1] = p[1] ; // sentinel values

    lt.p1 = t; lt.p2 = t; lt.p2.x = 5000000; // test line is orizontal,
    //travelling towards the infinite positive x
    //the alternative lt.p2.x = 300*INT_MAX wouldn't work

    for(i=1; i<=N; i++)
    {
        lp.p1=p[i] ; lp.p2=p[i]; // degenerate a polygon segment into a polygon vertex
        if(!intersect(lp,lt)){
            lp.p2 = p[j] ; j = i;
            if( intersect(lp,lt) ) count ++ ;
        }
    }

    //cout << "\nNumber of inersections = " << count << "\n" ;

    return count & 1 ;
}

int outline(mypoint t, mypoint *p,int N)
{
    mypoint pa,pb; // variables that call the ccw
```

c:\Documents and Settings\Nick\My Documents\...\Projects\rPORT\rPORT\spatialSearching.cpp 2

```

p[0]=p[N];p[N+1]=p[1]; //sentinel values

for(int i=0; i<=N; i++){
    pa=p[i];pb=p[i+1];
    if( ((pa.x==t.x)&&(pa.y==t.y)) || ((pb.x==t.x)&&(pb.y==t.y)) )
    {
        // "Point is a vertex"
        return 1;
    }

    if(!ccw(pa,pb,t)) {
        // "Point on outline"
        return 1;
    }
}

return 0;
}

bool inPolygon(mypoint t, mypoint *p,int N)
{
    if( inside(t, p, N) || outline(t, p, N) ){
        return true ;
    }

    return false ;
}

void cyclic(mypoint t, int rin, int rout)
{
    aisINconical=false ;
    aisINinner=false;
    aisINouter=false;
    double distance = (t.x*t.x)+(t.y*t.y) ;
    distance = sqrt(distance) ;
    if(distance <= rin && !(aisINSafetyzone||aisINtransitional)) { aisINinner=true ; return;}

    if(distance>= rout){ aisINouter=true ; return;}

    if(distance<rout && distance>rin) aisINconical=true ;
}

void transitional(mypoint a, mypoint *p)
{
    if(inPolygon(a, p, 4)) aisINtransitional=true ;
    if(aisINapproach) aisINtransitional=false ;
    if(aisINTakeoff) aisINtransitional=false ;
    if(aisINSafetyzone) aisINtransitional=false ;
}

void cyclicB(mypoint t, int rin, int rout)
{
    bisINconical=false ;
    bisINinner=false;
    bisINouter=false;

```


c:\Documents and Settings\Nick\My Documents\...\Projects\rPORT\rPORT\spatialSearching.cpp 3

```
double distance = (t.x*t.x)+(t.y*t.y) ;
distance = sqrt(distance) ;
if(distance <= rin && !(bisINsafetyzone||bisINtransitional)) { bisINinner=true ; return;}

if(distance>= rout){ bisINouter=true ; return;}

if(distance<rout && distance>rin) bisINconical=true ;

}

void transitionalB(mypoint a, mypoint *p)
{
    if(inPolygon(a, p, 4)) bisINtransitional=true ;
    if(bisINapproach) bisINtransitional=false ;
    if(bisINTakeoff) bisINtransitional=false ;
    if(bisINsafetyzone) bisINtransitional=false ;
}
```

```
c:\Documents and Settings\Nick\My Documents\... Studio 2008\Projects\rPORT\rPORT\stdafx.cpp_1
// stdafx.cpp : source file that includes just the standard includes
// rPORT.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"
```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 1

```
#include "stdafx.h"
#include "myData.h"

double aRW_lenght;

mypoint aRW_M, aRW_M2, aRW_Mbar ;
mypoint aRW_AP, aRW_AP2, aRW_APbar ;
mypoint aRW_TO, aRW_TO2, aRW_TObar ;

int aAP_Xthreshold, aAP_Ythreshold, aAP_l1, aAP_l2, aAP_l3 ;
double aAP_s1, aAP_s2, aAP_s3, aAP_divergence ;
int aRin, aRout ;
int aTO_X, aTO_Y, aTO_l1, aTO_l2 ;
double aTO_s, aTO_divergence;
int aSezorX, aSezorY ;
double aTR_s;
mypoint aapproach[6], atakeoff[8], asafetyzone[6], atransitional[6];

void asetDimensions(int a)
{
    switch(a) {
        case 101:
            aAP_Xthreshold = 30;
            aAP_Ythreshold = 60;
            aAP_l1 = 1600;
            aAP_l2 = 0;
            aAP_l3 = 0;
            aAP_s1 = 0.05;
            aAP_s2 = 0;
            aAP_s3 = 0;
            aAP_divergence = 0.1;
            aRin = 2000;
            aRout = 2700;
            aTO_X = 30;
            aTO_Y = 60;
            aTO_l1 = 1600;
            aTO_l2 = 0;
            aTO_s = 0.05;
            aTO_divergence = 0.100;
            aSezorX = 30;
            aSezorY = 30;
            aTR_s = 0.2;
            break;

        case 102:
            aAP_Xthreshold = 60;
            aAP_Ythreshold = 80;
            aAP_l1 = 2500;
            aAP_l2 = 0;
            aAP_l3 = 0;
            aAP_s1 = 0.04;
            aAP_s2 = 0;
            aAP_s3 = 0;
            aAP_divergence = 0.1;
            aRin = 2500;
            aRout = 3600;
            aTO_X = 60;
            aTO_Y = 80;
            aTO_l1 = 2500;
            aTO_l2 = 0;
            aTO_s = 0.04;
            aTO_divergence = 0.1;
            aSezorX = 60;
            aSezorY = 40;
```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 2

```
        aTR_s = 0.2;
        break;

    case 103:
        aAP_Xthreshold = 60;
        aAP_Ythreshold = 150;
        aAP_l1 = 3000;
        aAP_l2 = 0;
        aAP_l3 = 0;
        aAP_s1 = 0.0333;
        aAP_s2 = 0;
        aAP_s3 = 0;
        aAP_divergence = 0.1;
        aRin = 4000;
        aRout = 5500;
        aTO_X = 60;
        aTO_Y = 180;
        aTO_l1 = 4080;
        aTO_l2 = 10920;
        aTO_s = 0.02;
        aTO_divergence = 0.125;
        aSezorX = 60;
        aSezorY = 75;
        aTR_s = 0.143;
        break;

    case 104:
        aAP_Xthreshold = 60;
        aAP_Ythreshold = 150;
        aAP_l1 = 3000;
        aAP_l2 = 0;
        aAP_l3 = 0;
        aAP_s1 = 0.025;
        aAP_s2 = 0;
        aAP_s3 = 0;
        aAP_divergence = 0.1;
        aRin = 4000;
        aRout = 6000;
        aTO_X = 60;
        aTO_Y = 180;
        aTO_l1 = 4080;
        aTO_l2 = 10920;
        aTO_s = 0.02;
        aTO_divergence = 0.125;
        aSezorX = 60;
        aSezorY = 75;
        aTR_s = 0.143;
        break;

    case 201:
        aAP_Xthreshold = 60;
        aAP_Ythreshold = 150;
        aAP_l1 = 2500;
        aAP_l2 = 0;
        aAP_l3 = 0;
        aAP_s1 = 0.033;
        aAP_s2 = 0;
        aAP_s3 = 0;
        aAP_divergence = 0.15;
        aRin = 3500;
        aRout = 4700;
        aTO_X = 30;
        aTO_Y = 60;
        aTO_l1 = 1600;
        aTO_l2 = 0;
        aTO_s = 0.05;
        aTO_divergence = 0.1;
        aSezorX = 60;
        aSezorY = 75;
```

c:\Documents and Settings\Nick\My Documents\...\2008\Projects\rPORT\rPORT\surfaceCreation.cpp 3

```
    aTR_s = 0.2;
    break;

case 202:
    aAP_Xthreshold = 60;
    aAP_Ythreshold = 150;
    aAP_l1 = 2500;
    aAP_l2 = 0;
    aAP_l3 = 0;
    aAP_s1 = 0.033;
    aAP_s2 = 0;
    aAP_s3 = 0;
    aAP_divergence = 0.15;
    aRin = 3500;
    aRout = 4700;
    aTO_X = 60;
    aTO_Y = 80;
    aTO_l1 = 2500;
    aTO_l2 = 0;
    aTO_s = 0.04;
    aTO_divergence = 0.1;
    aSezorX = 60;
    aSezorY = 75;
    aTR_s = 0.2;
    break;

case 203:
    aAP_Xthreshold = 60;
    aAP_Ythreshold = 300;
    aAP_l1 = 3000;
    aAP_l2 = 3600;
    aAP_l3 = 8400;
    aAP_s1 = 0.02;
    aAP_s2 = 0.025;
    aAP_s3 = 0;
    aAP_divergence = 0.15;
    aRin = 4000;
    aRout = 5500;
    aTO_X = 60;
    aTO_Y = 180;
    aTO_l1 = 4080;
    aTO_l2 = 10920;
    aTO_s = 0.02;
    aTO_divergence = 0.125;
    aSezorX = 60;
    aSezorY = 150;
    aTR_s = 0.143;
    break;

case 204:
    aAP_Xthreshold = 60;
    aAP_Ythreshold = 300;
    aAP_l1 = 3000;
    aAP_l2 = 3600;
    aAP_l3 = 8400;
    aAP_s1 = 0.02;
    aAP_s2 = 0.025;
    aAP_s3 = 0;
    aAP_divergence = 0.15;
    aRin = 4000;
    aRout = 6000;
    aTO_X = 60;
    aTO_Y = 180;
    aTO_l1 = 4080;
    aTO_l2 = 10920;
    aTO_s = 0.02;
    aTO_divergence = 0.125;
    aSezorX = 60;
    aSezorY = 150;
```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 4

```
        aTR_s = 0.143;
        break;

    case 301:
        aAP_Xthreshold = 60;
        aAP_Ythreshold = 150;
        aAP_l1 = 3000;
        aAP_l2 = 12000;
        aAP_l3 = 0;
        aAP_s1 = 0.025;
        aAP_s2 = 0.03;
        aAP_s3 = 0;
        aAP_divergence = 0.15;
        aRin = 3500;
        aRout = 4700;
        aTO_X = 30;
        aTO_Y = 60;
        aTO_l1 = 1600;
        aTO_l2 = 0;
        aTO_s = 0.05;
        aTO_divergence = 0.1;
        aSezorX = 60;
        aSezorY = 75;
        aTR_s = 0.2;
        break;

    case 302:
        aAP_Xthreshold = 60;
        aAP_Ythreshold = 150;
        aAP_l1 = 3000;
        aAP_l2 = 12000;
        aAP_l3 = 0;
        aAP_s1 = 0.025;
        aAP_s2 = 0.03;
        aAP_s3 = 0;
        aAP_divergence = 0.15;
        aRin = 3500;
        aRout = 4700;
        aTO_X = 60;
        aTO_Y = 80;
        aTO_l1 = 2500;
        aTO_l2 = 0;
        aTO_s = 0.04;
        aTO_divergence = 0.1;
        aSezorX = 60;
        aSezorY = 75;
        aTR_s = 0.2;
        break;

    case 303:
        aAP_Xthreshold = 60;
        aAP_Ythreshold = 300;
        aAP_l1 = 3000;
        aAP_l2 = 3600;
        aAP_l3 = 8400;
        aAP_s1 = 0.02;
        aAP_s2 = 0.025;
        aAP_s3 = 0;
        aAP_divergence = 0.15;
        aRin = 4000;
        aRout = 6000;
        aTO_X = 60;
        aTO_Y = 180;
        aTO_l1 = 4080;
        aTO_l2 = 10920;
        aTO_s = 0.02;
        aTO_divergence = 0.125;
        aSezorX = 60;
        aSezorY = 150;
```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 5

```

        aTR_s = 0.143;
        break;

    case 304:
        aAP_Xthreshold = 60;
        aAP_Ythreshold = 300;
        aAP_l1 = 3000;
        aAP_l2 = 3600;
        aAP_l3 = 8400;
        aAP_s1 = 0.02;
        aAP_s2 = 0.025;
        aAP_s3 = 0;
        aAP_divergence = 0.15;
        aRin = 4000;
        aRout = 6000;
        aTO_X = 60;
        aTO_Y = 180;
        aTO_l1 = 4080;
        aTO_l2 = 10920;
        aTO_s = 0.02;
        aTO_divergence = 0.125;
        aSezorX = 60;
        aSezorY = 150;
        aTR_s = 0.143;
        break;

    }

}

void acreateSurfaces()
{
    asafetyzone[1].x = (aRW_lenght/2) + aSezorX ;
    asafetyzone[2].x = asafetyzone[1].x ;
    asafetyzone[3].x = -(asafetyzone[2].x);
    asafetyzone[4].x = asafetyzone[3].x ;
    asafetyzone[1].y = -aSezorY ;
    asafetyzone[2].y = aSezorY ;
    asafetyzone[3].y = aSezorY ;
    asafetyzone[4].y = -aSezorY ;

    aapproach[1].x = aRW_AP.x - aAP_Xthreshold ;
    //aAP_Ythreshold is always even so no digits lost during int>int>double eg 100->50->50.00
    aapproach[1].y = aAP_Ythreshold/2 ;
    aapproach[2].x = aapproach[1].x - (aAP_l1 + aAP_l2 + aAP_l3) ;
    aapproach[2].y = aapproach[1].y + ( (aAP_l1 + aAP_l2 + aAP_l3) * aAP_divergence) ;
    aapproach[3].x = aapproach[2].x ;
    aapproach[3].y = -aapproach[2].y ;
    aapproach[4].x = aapproach[1].x ;
    aapproach[4].y = -aapproach[1].y ;

    atakeoff[1].x = aRW_TO.x + aTO_X ;
    atakeoff[1].y = -(aTO_Y/2);
    atakeoff[2].x = atakeoff[1].x + aTO_l1 ;
    atakeoff[2].y = atakeoff[1].y - ((aTO_l1)*(aTO_divergence));
    atakeoff[3].x = atakeoff[2].x + aTO_l2 ;
    atakeoff[3].y = atakeoff[2].y ;
    atakeoff[4].x = atakeoff[3].x ;
    atakeoff[4].y = -atakeoff[3].y ;
    atakeoff[5].x = atakeoff[2].x ;
    atakeoff[5].y = -atakeoff[2].y ;
    atakeoff[6].x = atakeoff[1].x ;
    atakeoff[6].y = -atakeoff[1].y ;

    atransitional[1].x = asafetyzone[1].x ;//
    atransitional[1].y = -(aSezorY + (45/aTR_s)) ;
    atransitional[2].x = atransitional[1].x ;
    atransitional[2].y = -atransitional[1].y ;

```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 6

```

    atransitional[3].x = asafetyzone[3].x - (45/(aTR_s*aAP_divergence)) ; //
    atransitional[3].y = atransitional[2].y ;
    atransitional[4].x = atransitional[3].x ;
    atransitional[4].y = atransitional[1].y ;
}

```

//////////////////////////////////////RUNWAY 2

```
double bRW_lenght;
```

```

mypoint bRW_M, bRW_M2, bRW_Mbar ;
mypoint bRW_AP, bRW_AP2, bRW_APbar ;
mypoint bRW_TO, bRW_TO2, bRW_TObar ;

```

```

int bAP_Xthreshold, bAP_Ythreshold, bAP_l1, bAP_l2, bAP_l3 ;
double bAP_s1, bAP_s2, bAP_s3, bAP_divergence ;
int bRin, bRout ;
int bTO_X, bTO_Y, bTO_l1, bTO_l2 ;
double bTO_s, bTO_divergence;
int bSezorX, bSezorY ;
double bTR_s;
mypoint bapproach[6], btakeoff[8], bsafetyzone[6], btransitional[6];

```

```

void bsetDimensions(int a)
{
    switch(a) {
        case 101:
            bAP_Xthreshold = 30;
            bAP_Ythreshold = 60;
            bAP_l1 = 1600;
            bAP_l2 = 0;
            bAP_l3 = 0;
            bAP_s1 = 0.05;
            bAP_s2 = 0;
            bAP_s3 = 0;
            bAP_divergence = 0.1;
            bRin = 2000;
            bRout = 2700;
            bTO_X = 30;
            bTO_Y = 60;
            bTO_l1 = 1600;
            bTO_l2 = 0;
            bTO_s = 0.05;
            bTO_divergence = 0.100;
            bSezorX = 30;
            bSezorY = 30;
            bTR_s = 0.2;
            break;

        case 102:
            bAP_Xthreshold = 60;
            bAP_Ythreshold = 80;
            bAP_l1 = 2500;
            bAP_l2 = 0;
            bAP_l3 = 0;

```


c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 7

```
bAP_s1 = 0.04;  
bAP_s2 = 0;  
bAP_s3 = 0;  
bAP_divergence = 0.1;  
bRin = 2500;  
bRout = 3600;  
bTO_X = 60;  
bTO_Y = 80;  
bTO_l1 = 2500;  
bTO_l2 = 0;  
bTO_s = 0.04;  
bTO_divergence = 0.1;  
bSezorX = 60;  
bSezorY = 40;  
bTR_s = 0.2;  
break;
```

```
case 103:  
    bAP_Xthreshold = 60;  
    bAP_Ythreshold = 150;  
    bAP_l1 = 3000;  
    bAP_l2 = 0;  
    bAP_l3 = 0;  
    bAP_s1 = 0.0333;  
    bAP_s2 = 0;  
    bAP_s3 = 0;  
    bAP_divergence = 0.1;  
    bRin = 4000;  
    bRout = 5500;  
    bTO_X = 60;  
    bTO_Y = 180;  
    bTO_l1 = 4080;  
    bTO_l2 = 10920;  
    bTO_s = 0.02;  
    bTO_divergence = 0.125;  
    bSezorX = 60;  
    bSezorY = 75;  
    bTR_s = 0.143;  
    break;
```

```
case 104:  
    bAP_Xthreshold = 60;  
    bAP_Ythreshold = 150;  
    bAP_l1 = 3000;  
    bAP_l2 = 0;  
    bAP_l3 = 0;  
    bAP_s1 = 0.025;  
    bAP_s2 = 0;  
    bAP_s3 = 0;  
    bAP_divergence = 0.1;  
    bRin = 4000;  
    bRout = 6000;  
    bTO_X = 60;  
    bTO_Y = 180;  
    bTO_l1 = 4080;  
    bTO_l2 = 10920;  
    bTO_s = 0.02;  
    bTO_divergence = 0.125;  
    bSezorX = 60;  
    bSezorY = 75;  
    bTR_s = 0.143;  
    break;
```

```
case 201:  
    bAP_Xthreshold = 60;  
    bAP_Ythreshold = 150;  
    bAP_l1 = 2500;  
    bAP_l2 = 0;  
    bAP_l3 = 0;
```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 8

```
bAP_s1 = 0.033;  
bAP_s2 = 0;  
bAP_s3 = 0;  
bAP_divergence = 0.15;  
bRin = 3500;  
bRout = 4700;  
bTO_X = 30;  
bTO_Y = 60;  
bTO_l1 = 1600;  
bTO_l2 = 0;  
bTO_s = 0.05;  
bTO_divergence = 0.1;  
bSezorX = 60;  
bSezorY = 75;  
bTR_s = 0.2;  
break;
```

```
case 202:  
bAP_Xthreshold = 60;  
bAP_Ythreshold = 150;  
bAP_l1 = 2500;  
bAP_l2 = 0;  
bAP_l3 = 0;  
bAP_s1 = 0.033;  
bAP_s2 = 0;  
bAP_s3 = 0;  
bAP_divergence = 0.15;  
bRin = 3500;  
bRout = 4700;  
bTO_X = 60;  
bTO_Y = 80;  
bTO_l1 = 2500;  
bTO_l2 = 0;  
bTO_s = 0.04;  
bTO_divergence = 0.1;  
bSezorX = 60;  
bSezorY = 75;  
bTR_s = 0.2;  
break;
```

```
case 203:  
bAP_Xthreshold = 60;  
bAP_Ythreshold = 300;  
bAP_l1 = 3000;  
bAP_l2 = 3600;  
bAP_l3 = 8400;  
bAP_s1 = 0.02;  
bAP_s2 = 0.025;  
bAP_s3 = 0;  
bAP_divergence = 0.15;  
bRin = 4000;  
bRout = 5500;  
bTO_X = 60;  
bTO_Y = 180;  
bTO_l1 = 4080;  
bTO_l2 = 10920;  
bTO_s = 0.02;  
bTO_divergence = 0.125;  
bSezorX = 60;  
bSezorY = 150;  
bTR_s = 0.143;  
break;
```

```
case 204:  
bAP_Xthreshold = 60;  
bAP_Ythreshold = 300;  
bAP_l1 = 3000;  
bAP_l2 = 3600;  
bAP_l3 = 8400;
```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 9

```
bAP_s1 = 0.02;  
bAP_s2 = 0.025;  
bAP_s3 = 0;  
bAP_divergence = 0.15;  
bRin = 4000;  
bRout = 6000;  
bTO_X = 60;  
bTO_Y = 180;  
bTO_l1 = 4080;  
bTO_l2 = 10920;  
bTO_s = 0.02;  
bTO_divergence = 0.125;  
bSezorX = 60;  
bSezorY = 150;  
bTR_s = 0.143;  
break;
```

```
case 301:  
    bAP_Xthreshold = 60;  
    bAP_Ythreshold = 150;  
    bAP_l1 = 3000;  
    bAP_l2 = 12000;  
    bAP_l3 = 0;  
    bAP_s1 = 0.025;  
    bAP_s2 = 0.03;  
    bAP_s3 = 0;  
    bAP_divergence = 0.15;  
    bRin = 3500;  
    bRout = 4700;  
    bTO_X = 30;  
    bTO_Y = 60;  
    bTO_l1 = 1600;  
    bTO_l2 = 0;  
    bTO_s = 0.05;  
    bTO_divergence = 0.1;  
    bSezorX = 60;  
    bSezorY = 75;  
    bTR_s = 0.2;  
    break;
```

```
case 302:  
    bAP_Xthreshold = 60;  
    bAP_Ythreshold = 150;  
    bAP_l1 = 3000;  
    bAP_l2 = 12000;  
    bAP_l3 = 0;  
    bAP_s1 = 0.025;  
    bAP_s2 = 0.03;  
    bAP_s3 = 0;  
    bAP_divergence = 0.15;  
    bRin = 3500;  
    bRout = 4700;  
    bTO_X = 60;  
    bTO_Y = 80;  
    bTO_l1 = 2500;  
    bTO_l2 = 0;  
    bTO_s = 0.04;  
    bTO_divergence = 0.1;  
    bSezorX = 60;  
    bSezorY = 75;  
    bTR_s = 0.2;  
    break;
```

```
case 303:  
    bAP_Xthreshold = 60;  
    bAP_Ythreshold = 300;  
    bAP_l1 = 3000;  
    bAP_l2 = 3600;  
    bAP_l3 = 8400;
```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 10

```

        bAP_s1 = 0.02;
        bAP_s2 = 0.025;
        bAP_s3 = 0;
        bAP_divergence = 0.15;
        bRin = 4000;
        bRout = 6000;
        bTO_X = 60;
        bTO_Y = 180;
        bTO_l1 = 4080;
        bTO_l2 = 10920;
        bTO_s = 0.02;
        bTO_divergence = 0.125;
        bSezorX = 60;
        bSezorY = 150;
        bTR_s = 0.143;
        break;

    case 304:
        bAP_Xthreshold = 60;
        bAP_Ythreshold = 300;
        bAP_l1 = 3000;
        bAP_l2 = 3600;
        bAP_l3 = 8400;
        bAP_s1 = 0.02;
        bAP_s2 = 0.025;
        bAP_s3 = 0;
        bAP_divergence = 0.15;
        bRin = 4000;
        bRout = 6000;
        bTO_X = 60;
        bTO_Y = 180;
        bTO_l1 = 4080;
        bTO_l2 = 10920;
        bTO_s = 0.02;
        bTO_divergence = 0.125;
        bSezorX = 60;
        bSezorY = 150;
        bTR_s = 0.143;
        break;

    }

}

void bcreateSurfaces()
{
    bsafetyzone[1].x = (bRW_lenght/2) + bSezorX ;
    bsafetyzone[2].x = bsafetyzone[1].x ;
    bsafetyzone[3].x = -(bsafetyzone[2].x);
    bsafetyzone[4].x = bsafetyzone[3].x ;
    bsafetyzone[1].y = -bSezorY ;
    bsafetyzone[2].y = bSezorY ;
    bsafetyzone[3].y = bSezorY ;
    bsafetyzone[4].y = -bSezorY ;

    bapproach[1].x = bRW_AP.x - bAP_Xthreshold ;
    //aAP_Ythreshold is always even so no digits lost during int>int>double eg 100->50->50.00
    bapproach[1].y = bAP_Ythreshold/2 ;
    bapproach[2].x = bapproach[1].x - (bAP_l1 + bAP_l2 + bAP_l3) ;
    bapproach[2].y = bapproach[1].y + ( (bAP_l1 + bAP_l2 + bAP_l3) * bAP_divergence) ;
    bapproach[3].x = bapproach[2].x ;
    bapproach[3].y = -bapproach[2].y ;
    bapproach[4].x = bapproach[1].x ;
    bapproach[4].y = -bapproach[1].y ;

    btakeoff[1].x = bRW_TO.x + bTO_X ;
    btakeoff[1].y = -(bTO_Y/2);
    btakeoff[2].x = btakeoff[1].x + bTO_l1 ;

```

c:\Documents and Settings\Nick\My Documents\...2008\Projects\rPORT\rPORT\surfaceCreation.cpp 11

```
btakeoff[2].y = btakeoff[1].y - ((bTO_l1)*(bTO_divergence));
btakeoff[3].x = btakeoff[2].x + bTO_l2 ;
btakeoff[3].y = btakeoff[2].y ;
btakeoff[4].x = btakeoff[3].x ;
btakeoff[4].y = -btakeoff[3].y ;
btakeoff[5].x = btakeoff[2].x ;
btakeoff[5].y = -btakeoff[2].y ;
btakeoff[6].x = btakeoff[1].x ;
btakeoff[6].y = -btakeoff[1].y ;

btransitional[1].x = bsafetyzone[1].x ;//
btransitional[1].y = -(bSezorY + (45/bTR_s)) ;
btransitional[2].x = btransitional[1].x ;
btransitional[2].y = -btransitional[1].y ;
btransitional[3].x = bsafetyzone[3].x - (45/(bTR_s*bAP_divergence)) ; //
btransitional[3].y = btransitional[2].y ;
btransitional[4].x = btransitional[3].x ;
btransitional[4].y = btransitional[1].y ;
}
```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\export2Autocad.cpp

1

```

#include "stdafx.h"
#include "myData.h"
#include <stdio.h>
#include <fstream>
#include <math.h>

extern mypoint aScout, aScout180;
extern mypoint bScout, bScout180;
extern mypoint aRW_Mbar, bRW_Mbar;
extern int aRin, aRout;
extern int bRin, bRout;

extern mypoint atransitional[6]; //actual surface representation
mypoint atransitionalbar[6];
extern mypoint asafetyzone[6]; //actual surface representation
mypoint asafetyzonebar[6];
extern mypoint atakeoff[8]; //actual surface representation
mypoint atakeoffbar[8];
extern mypoint aapproach[6]; //actual surface representation
mypoint aapproachbar[6]; //global system

extern mypoint btransitional[6]; //actual surface representation
mypoint btransitionalbar[6];
extern mypoint bsafetyzone[6]; //actual surface representation
mypoint bsafetyzonebar[6];
extern mypoint btakeoff[8]; //actual surface representation
mypoint btakeoffbar[8];
extern mypoint bapproach[6]; //actual surface representation
mypoint bapproachbar[6];

void printDXF(FILE *fout, char *desc, mypoint *shape, int N)
{
    for(int i=1; i<=N; i++){
        fprintf(fout, "0\nLINE\n8\n%s\n", desc);
        fprintf(fout, "10\n %lf\n 20\n %lf\n 11\n %lf\n 21\n %lf\n", shape[i].x, shape[i].y,
            shape[i+1].x,
            shape[i+1].y);
    }
}

void global (mypoint *pbar, mypoint *p, double angle, mypoint RW_Cbar, int N)
{
    double PI = 3.14159265;
    double a = (angle*PI)/(180.0);

    for(int i = 0; i<= N; i++){
        pbar[i].x = (p[i].x * cos(a)) + (p[i].y * sin(a));
        pbar[i].y = (p[i].y * cos(a)) - (p[i].x * sin(a));

        pbar[i].x = pbar[i].x + RW_Cbar.x;
        pbar[i].y = pbar[i].y + RW_Cbar.y;
    }
}

void global_reverse (mypoint *pbar, mypoint *p, double angle, mypoint RW_Cbar, int N)
{
    double PI = 3.14159265;
    double a = ((angle*PI)/(180.0)) + PI;

    for(int i = 0; i<= N; i++){
        pbar[i].x = (p[i].x * cos(a)) + (p[i].y * sin(a));
        pbar[i].y = (p[i].y * cos(a)) - (p[i].x * sin(a));

        pbar[i].x = pbar[i].x + RW_Cbar.x;
        pbar[i].y = pbar[i].y + RW_Cbar.y;
    }
}

```

c:\Documents and Settings\Nick\Desktop\rPORT\rPORT\export2Autocad.cpp

2

```

void exportDXF(char *filename)
{
    double aRinDXF = aRin;
    double bRinDXF = bRin;
    double aRoutDXF = aRout;
    double bRoutDXF = bRout;

    FILE *fout = fopen(filename, "w");

    fprintf(fout, "0\nSECTION\n2\nENTITIES\n");

    if (aScout.z != 999999){
        printDXF(fout, "RUNWAY 1 Strip", asafetyzonebar, 4);
        printDXF(fout, "RUNWAY 1 Transitional surface", atransitionalbar, 4);
        printDXF(fout, "RUNWAY 1 Approach surface", aapproachbar, 4);
        printDXF(fout, "RUNWAY 1 Take off surface", atakeoffbar, 6);
        fprintf(fout, "0\nCIRCLE\n8\nRunway 1 Conical surface\n10\n%lf\n20\n%lf\n40\n%lf\n",
        aRW_Mbar.x, aRW_Mbar.y, aRinDXF);
        fprintf(fout, "0\nCIRCLE\n8\nRunway 1 Conical surface\n10\n%lf\n20\n%lf\n40\n%lf\n",
        aRW_Mbar.x, aRW_Mbar.y, aRoutDXF);
    }

    if (bScout.z != 999999){
        printDXF(fout, "RUNWAY 2 Strip", bsafetyzonebar, 4);
        printDXF(fout, "RUNWAY 2 Transitional surface", btransitionalbar, 4);
        printDXF(fout, "RUNWAY 2 Approach surface", bapproachbar, 4);
        printDXF(fout, "RUNWAY 2 Take off surface", btakeoffbar, 6);
        fprintf(fout, "0\nCIRCLE\n8\nRunway 2 Conical surface\n10\n%lf\n20\n%lf\n40\n%lf\n",
        bRW_Mbar.x, bRW_Mbar.y, bRinDXF);
        fprintf(fout, "0\nCIRCLE\n8\nRunway 2 Conical surface\n10\n%lf\n20\n%lf\n40\n%lf\n",
        bRW_Mbar.x, bRW_Mbar.y, bRoutDXF);
    }

    if (aScout180.z != 999999){
        global_reverse(atriansitionalbar, atransitional, afaixis, aRW_Mbar, 6);
        global_reverse(asafetyzonebar, asafetyzone, afaixis, aRW_Mbar, 6);
        global_reverse(atakeoffbar, atakeoff, afaixis, aRW_Mbar, 8);
        global_reverse(aapproachbar, aapproach, afaixis, aRW_Mbar, 6);
        printDXF(fout, "RUNWAY 1 Strip", asafetyzonebar, 4);
        printDXF(fout, "RUNWAY 1 Transitional surface", atransitionalbar, 4);
        printDXF(fout, "RUNWAY 1 Approach surface", aapproachbar, 4);
        printDXF(fout, "RUNWAY 1 Take off surface", atakeoffbar, 6);
        fprintf(fout, "0\nCIRCLE\n8\nRunway 1 Conical surface\n10\n%lf\n20\n%lf\n40\n%lf\n",
        aRW_Mbar.x, aRW_Mbar.y, aRinDXF);
        fprintf(fout, "0\nCIRCLE\n8\nRunway 1 Conical surface\n10\n%lf\n20\n%lf\n40\n%lf\n",
        aRW_Mbar.x, aRW_Mbar.y, aRoutDXF);
    }

    if (bScout180.z != 999999){
        global_reverse(btransitionalbar, btransitional, bfaixis, bRW_Mbar, 6);
        global_reverse(bsafetyzonebar, bsafetyzone, bfaixis, bRW_Mbar, 6);
        global_reverse(btakeoffbar, btakeoff, bfaixis, bRW_Mbar, 8);
        global_reverse(bapproachbar, bapproach, bfaixis, bRW_Mbar, 6);
        printDXF(fout, "RUNWAY 2 Strip", bsafetyzonebar, 4);
        printDXF(fout, "RUNWAY 2 Transitional surface", btransitionalbar, 4);
        printDXF(fout, "RUNWAY 2 Approach surface", bapproachbar, 4);
        printDXF(fout, "RUNWAY 2 Take off surface", btakeoffbar, 6);
        fprintf(fout, "0\nCIRCLE\n8\nRunway 2 Conical surface\n10\n%lf\n20\n%lf\n40\n%lf\n",
        bRW_Mbar.x, bRW_Mbar.y, bRinDXF);
        fprintf(fout, "0\nCIRCLE\n8\nRunway 2 Conical surface\n10\n%lf\n20\n%lf\n40\n%lf\n",
        bRW_Mbar.x, bRW_Mbar.y, bRoutDXF);
    }

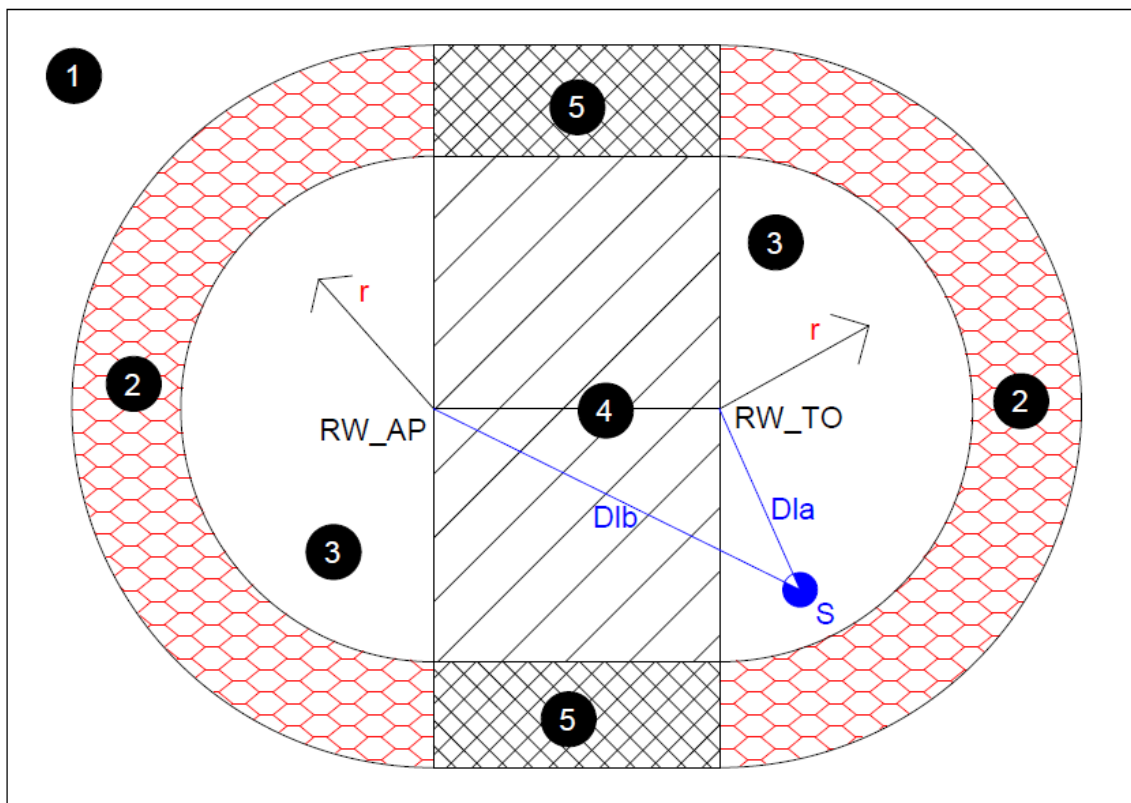
    fprintf(fout, "0\nENDSEC\n0\nEOF");

    fclose(fout);
}

```

ΠΑΡΑΡΤΗΜΑ Β

ΕΛΕΓΧΟΣ ΣΕ ΑΜΦΙΘΕΑΤΡΙΚΕΣ ΕΠΙΦΑΝΕΙΕΣ



ΣΧΗΜΑ Β.1

Κατά τις οδηγίες του ICAO, θεωρήθηκε ικανοποιητική προσέγγιση η λήψη εσωτερικής οριζόντιας επιφάνειας με ένα σημείο αναφοράς. Εναλλακτικά η λήψη εσωτερικής οριζόντιας επιφάνειας με δύο σημεία αναφοράς, τα άκρα του διαδρόμου, παρότι δεν αλλάζει τις μεθόδους υπολογισμού περίληψης σημείου σε επιφάνεια και επιτρεπτού υψομέτρου, μεταβάλλει αυτήν και τις σχετιζόμενες επιφάνειες (κωνική και εξωτερική οριζόντια) από κυκλικές σε «αμφιθεατρικές». Έτσι οι απλές ιδιότητες γεωμετρικών τόπων που χρησιμοποιήσαμε ως τώρα δεν αρκούν για να επιλύσουν το πρόβλημα και παρότι δεν εισάγονται νέες παράμετροι (οι R_{in} , R_{out} , RW_AP , RW_TO συνεχίζουν να αποτελούν επαρκή «περιγραφή») απαιτείται ο χωρισμός των επιφανειών σε περισσότερα τμήματα. Μια περιγραφή της «εξελιγμένης» αυτής παραμετροποίησης είναι η ακόλουθη:

ΠΕΡΙΟΧΗ	ΕΠΙΦΑΝΕΙΑ	ΕΛΕΓΧΟΣ ΠΕΡΙΛΗΨΗΣ
1	Εξωτερική οριζόντια	Αφαιρετικός
2	Κωνική	Σε τμήμα δακτυλίου
3	Εσωτερική οριζόντια	Σε κυκλικό τομέα
4	Εσωτερική οριζόντια	Σε πολύγωνο
5	Κωνική	Σε πολύγωνο

Έστω ότι προς έρευνα είναι το σημείο S. Η πορεία υπολογισμού (η μέθοδος βάση της οποίας σχεδιάζεται σχετικός αλγόριθμος) έχει ως εξής:

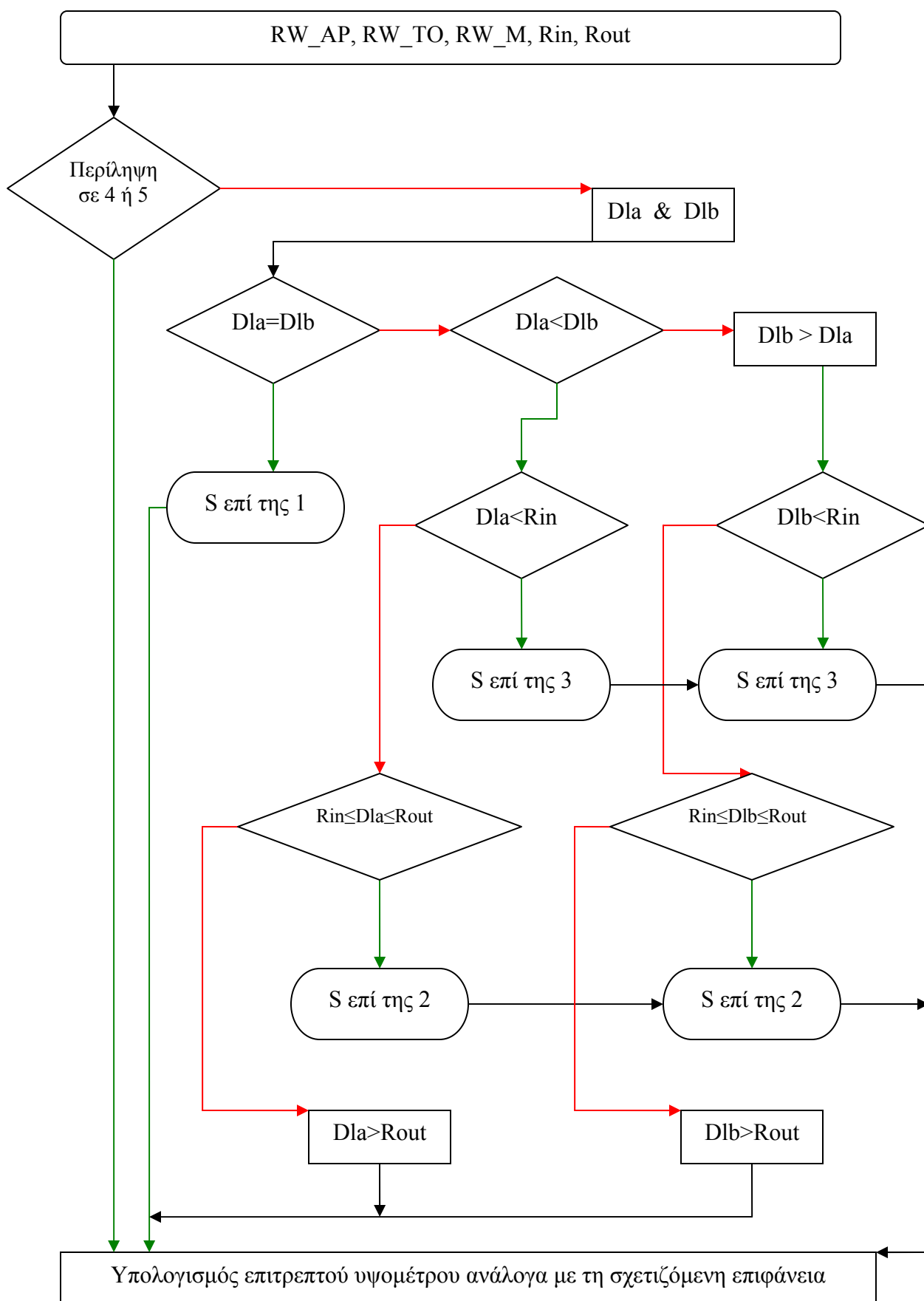
- i. Έλεγχος περίληψης (κατά τα γνωστά) στα πολύγωνα 4 και 5. Αν έχουμε θετικό αποτέλεσμα σταματάμε τον έλεγχο περίληψης και προχωράμε σε υπολογισμό επιτρεπτού υψομέτρου ανάλογα με την επιφάνεια.
- ii. Υπολογισμός αποστάσεων D1a & D1b από τα άκρα RW_TO, RW_AP αντίστοιχα. Σημείο αναφοράς των υπολογισμών θεωρείται το άκρο του διαδρόμου που δίνει τη μικρότερη επιβατική ακτίνα. Αν D1a = D1b τότε το σημείο κείται επί της μεσοκάθετης του άξονα του διαδρόμου οπότε όλοι οι έλεγχοι περίληψης θα έπρεπε είτε να είχαν σταματήσει στο προηγούμενο βήμα (άτοπο εφόσον φτάσαμε ως εδώ) είτε το S να ανήκει στην 1 (μόνο δεκτό αποτέλεσμα).
- iii. Έστω ότι D1a < D1b, οπότε το S βρίσκεται στην περιοχή του RW_TO. Εργαζόμαστε ως εξής (και αντίστοιχα για D1a > D1b): Εφόσον έχει ήδη ελεγχθεί η περίληψη στις 4 & 5, δεν απαιτείται ο έλεγχος της γωνίας της επιβατικής ακτίνας του σημείου S, επομένως διακρίνουμε τις περιπτώσεις:

$$D1a < R_{in} \rightarrow S \in 3$$

$$R_{in} \leq D1a \leq R_{out} \rightarrow S \in 2$$

$$R_{out} < D1a \rightarrow S \in 1$$
- iv. Υπολογισμός επιτρεπτού υψομέτρου, όπως ορίζεται από τις απαιτήσεις της επιφάνειας στην οποία περιλαμβάνεται το S.

Η μέθοδος εργασίας αυτή, σε λίγα σχετικά βήματα, δίνει αποτέλεσμα επί της περίληψης χωρίς να αντιμετωπίζουμε ευθέως το πρόβλημα τομέα κυκλικού δακτυλίου (περιοχές 2) ή σύνθετης εξωτερικής επιφάνειας (περιοχή 1). Έτσι αποφεύγεται η ενασχόληση με τις γωνίες επιβατικής ακτίνας του S, ενώ η περίληψη στην περιοχή 1 υποτίθεται αληθής όταν είναι άτοπη κάθε άλλη περίπτωση. Η σχετική πορεία εργασίας εικονίζεται στο ακόλουθο διάγραμμα ροής και αντίστοιχος αλγόριθμος μπορεί σχεδιαστεί κατά τα πρότυπα όσων έχουν υλοποιηθεί μέχρι τώρα. Σημειώνεται ότι προκύπτουν μικρές αλλαγές και στην διαδικασία υπολογισμού επιτρεπτού υψομέτρου, ωστόσο ο χωρισμός σε επιφάνειες 1,2,3,4,5 δίνει δυνατότητα διακριτού προγραμματισμού (ανά περίπτωση), εξαιτίας της απλότητας (και της ομοιότητας με τα προηγούμενα) του οποίου θεωρείται πως δεν πρέπει να γίνει ειδική μνεία.



ΠΑΡΑΡΤΗΜΑ Γ

ΠΙΝΑΚΕΣ, ΣΧΗΜΑΤΑ & ΕΙΚΟΝΕΣ

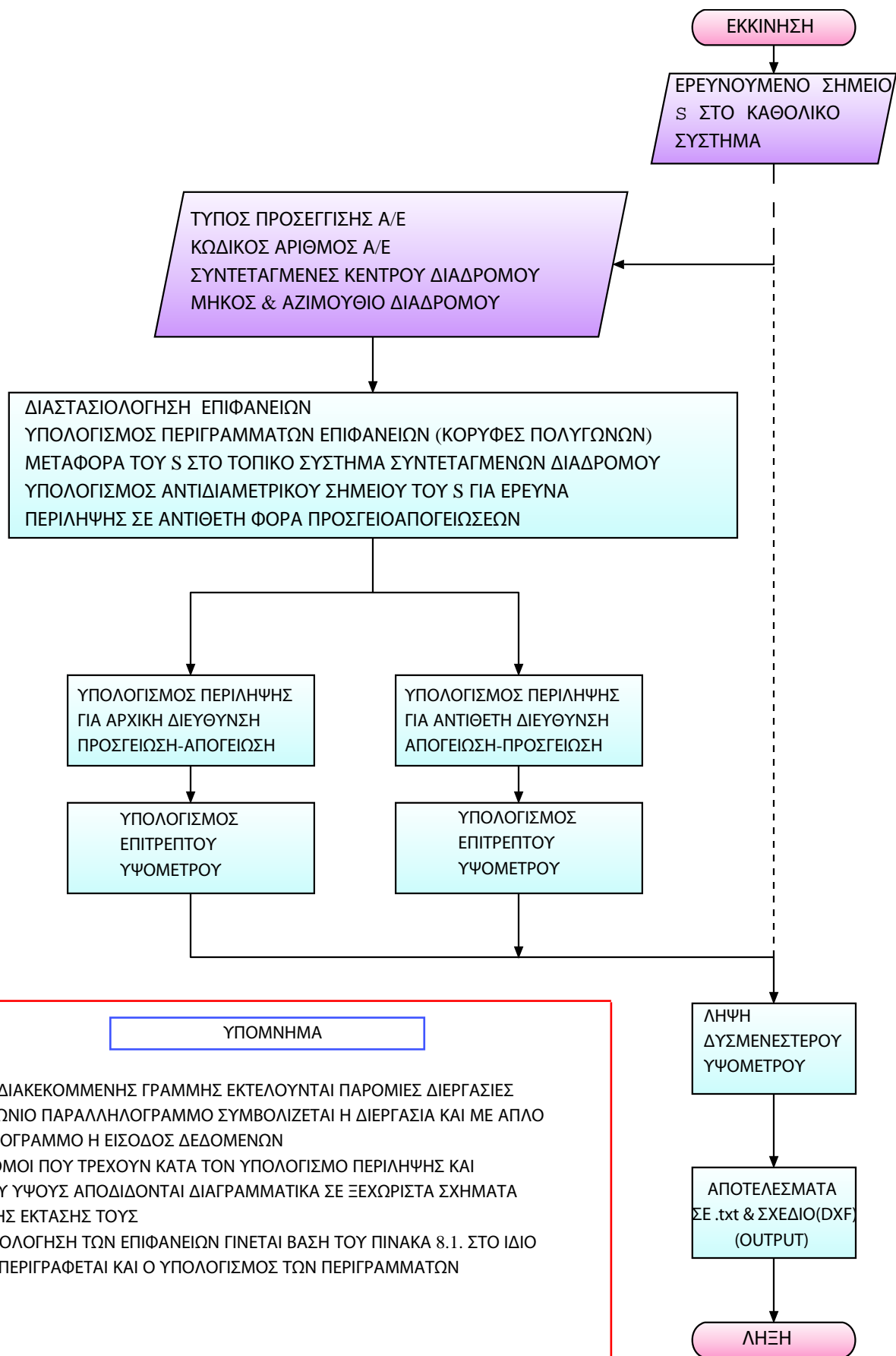
ΣΧΗΜΑ	ΠΕΡΙΓΡΑΦΗ
3.1	Επιφάνειες περιορισμού εμποδίων – Χρωματική διάκριση περιοχών σε κάτοψη
3.2	Επιφάνειες περιορισμού εμποδίων – Κάτοψη και τομές
3.3	Επιφάνειες περιορισμού εμποδίων – Προοπτικό σχήμα
3.4	Κάτοψη εσωτερικής οριζόντιας επιφάνειας με δύο σημεία αναφοράς
3.5	Εσωτερική μεταβατική επιφάνεια, εσωτερική επιφάνεια προσέγγισης και επιφάνεια ματαιούμενης προσγείωσης
4.1	Ανεμολόγιο
5.1	Δηλούμενες αποστάσεις
5.2	Μηκοτομή διαδρόμου (RUNWAY)
6.1	Συστήματα συντεταγμένων
7.1	Περιστροφή συστήματος συντεταγμένων
7.2	Μετατόπιση συστήματος συντεταγμένων
7.3	Προσανατολισμός τριών σημείων
7.4	Έλεγχος τομής βάση διαφορικής γεωμετρίας
7.5	Έλεγχος τομής βάση υπολογιστικής γεωμετρίας
7.6	Περίληψη σε κυκλική επιφάνεια
7.7	Κυρτό και μη κυρτό πολύγωνο
7.8	Απλό και μη απλό πολύγωνο
7.9	Περίληψη σε πολύγωνο
7.10	Περιπτώσεις ελέγχου περίληψης σε πολύγωνο
8.1	Άξονας διαδρόμου (προγραμματιστική απεικόνιση)
8.2	Ζώνη ασφαλείας (προγραμματιστική απεικόνιση)
8.3	Κωνική επιφάνεια (προγραμματιστική απεικόνιση)
8.4	Εσωτερική & εξωτερική οριζόντια επιφάνεια (προγραμματιστική απεικόνιση)
8.5	Επιφάνεια προσέγγισης (προγραμματιστική απεικόνιση)
8.6	Επιφάνεια απογείωσης (προγραμματιστική απεικόνιση)
8.7	Μεταβατική επιφάνεια (προγραμματιστική απεικόνιση)
10.1	Πορεία εργασίας στην περίπτωση ύπαρξης κλίσης σε διάδρομο

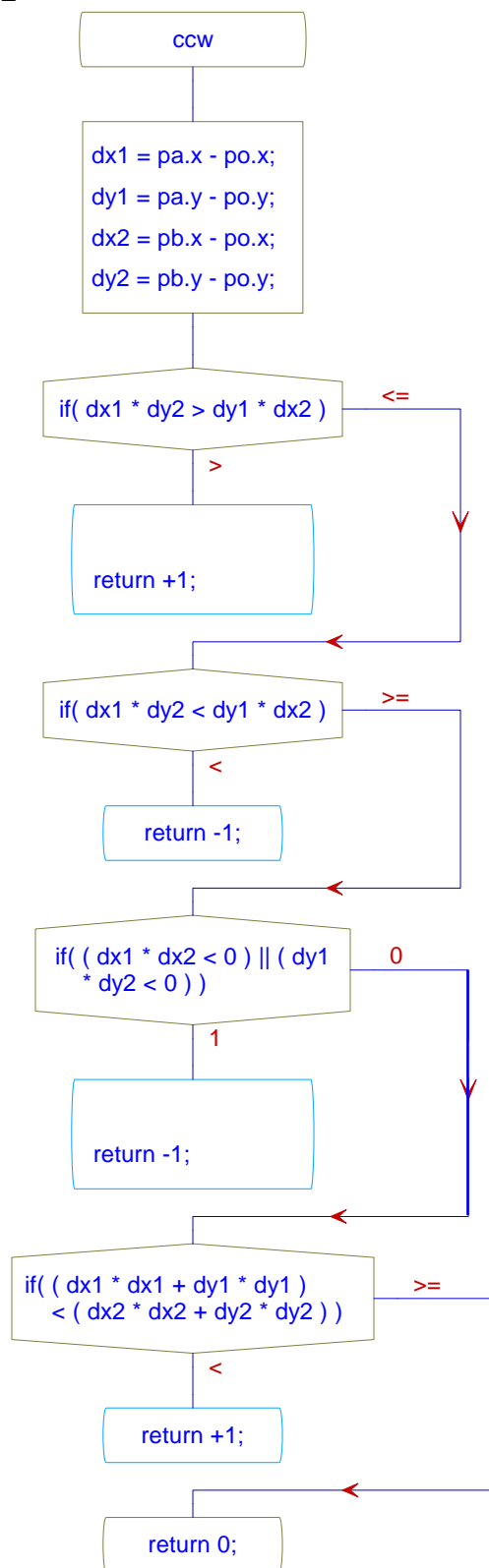
ΠΙΝΑΚΑΣ	ΠΕΡΙΓΡΑΦΗ
2.1	Χαρακτηριστικά α/φ που επηρεάζουν τον κωδικό αναφοράς α/ε
2.2	Κωδικός αναφοράς α/ε σχετιζόμενος με την κατηγορία προσέγγισης και σχεδίασης α/φ
3.1	Διάδρομοι προσγείωσης
3.2	Διάδρομοι απογείωσης
5.1	Ελάχιστο πλάτος διαδρόμου
5.2	Περιορισμοί μηκοτομής διαδρόμου
5.3	Ελάχιστες απαιτήσεις πλάτους ζώνης ασφαλείας
8.1	Παραμετροποίηση επιφανειών περιορισμού εμποδίων

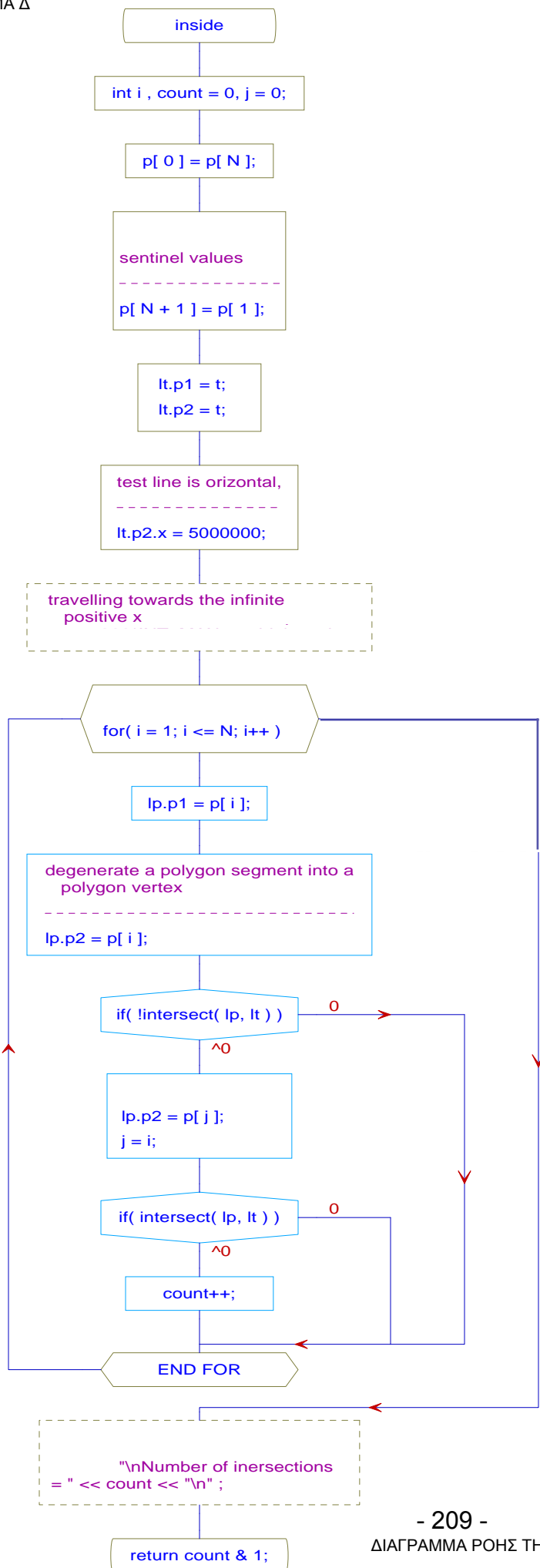
ΕΙΚΟΝΑ	ΠΕΡΙΓΡΑΦΗ
9.1	Το περιβάλλον της Visual C++ 2008 – Οθόνη δημιουργίας project
9.2	Το περιβάλλον της Visual C++ 2008 – Οθόνη σχεδίασης GUI
9.3	Πρόγραμμα rPORT – Οθόνη εισαγωγής δεδομένων
9.4	Πρόγραμμα rPORT – Οθόνη αποτελεσμάτων διαστασιολόγησης (Ζώνη ασφαλείας και μεταβατική επιφάνεια)
9.5	Πρόγραμμα rPORT – Οθόνη αποτελεσμάτων διαστασιολόγησης (οριζόντιες επιφάνειες και κωνική)
9.6	Πρόγραμμα rPORT – Οθόνη αποτελεσμάτων διαστασιολόγησης (επιφάνεια προσέγγισης)
9.7	Πρόγραμμα rPORT – Οθόνη αποτελεσμάτων διαστασιολόγησης (επιφάνεια απογείωσης)
9.8	Πρόγραμμα rPORT – Οθόνη αποτελεσμάτων έρευνας περίληψης και επιτρεπτού υψομέτρου
9.9	Παραγόμενη έξοδος των επιφανειών σε AutoCAD
10.1	Όγκος περικλειόμενος από δύο κυρτές επιφάνειες – Χρήση για υπολογισμό χωματισμών

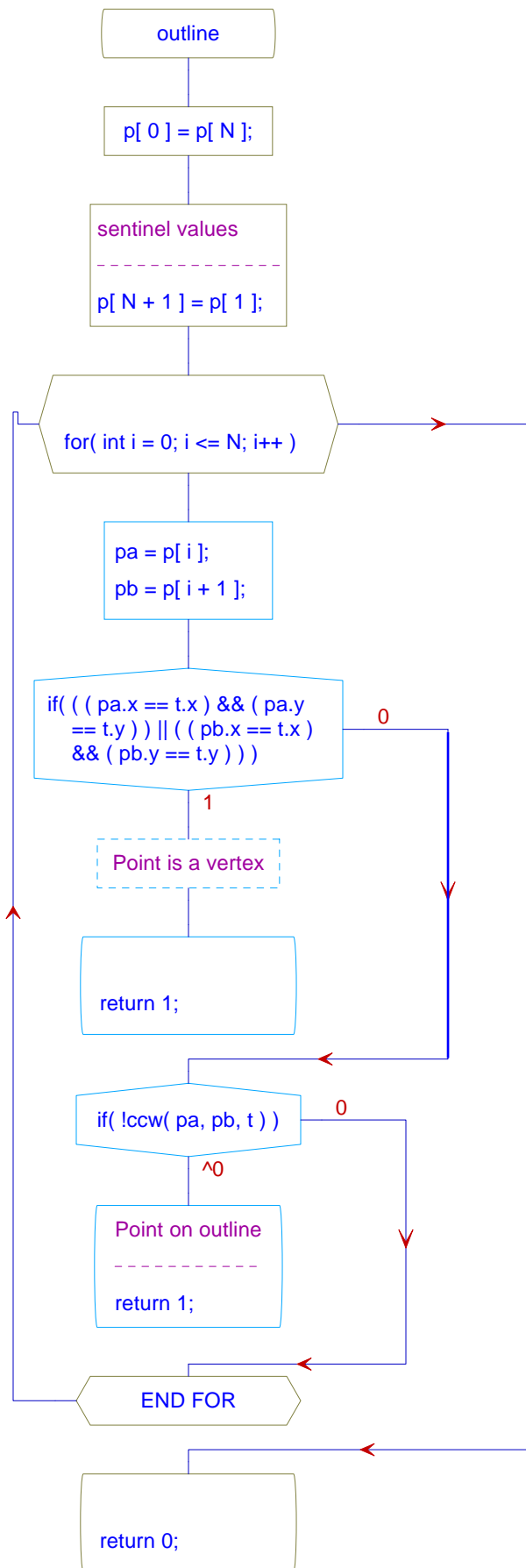
ΠΑΡΑΡΤΗΜΑ Δ

ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ

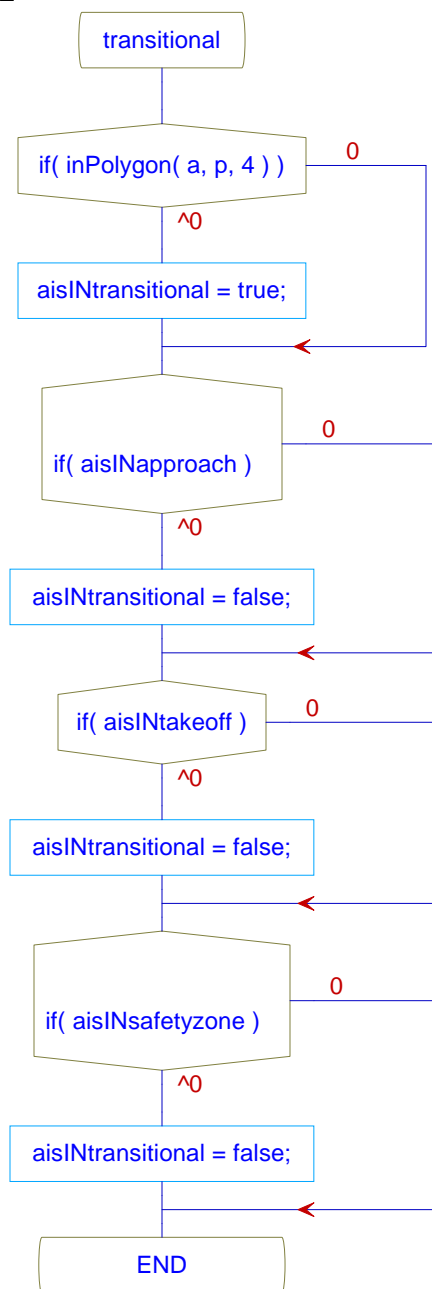


ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΤΟΥ ΚΑΤΗΓΟΡΗΜΑΤΟΣ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ (ΣΥΝΑΡΤΗΣΗ `ccw`)

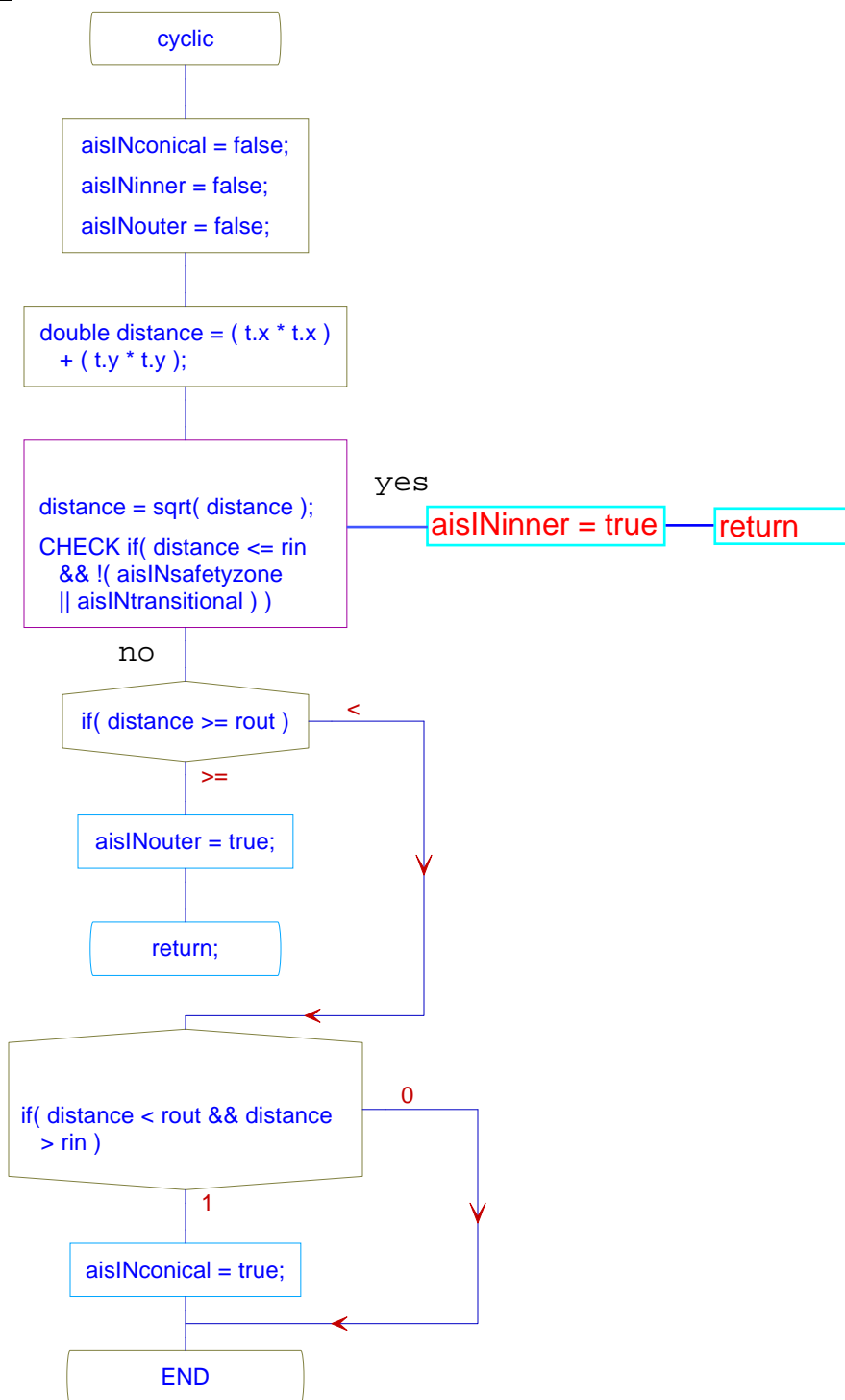


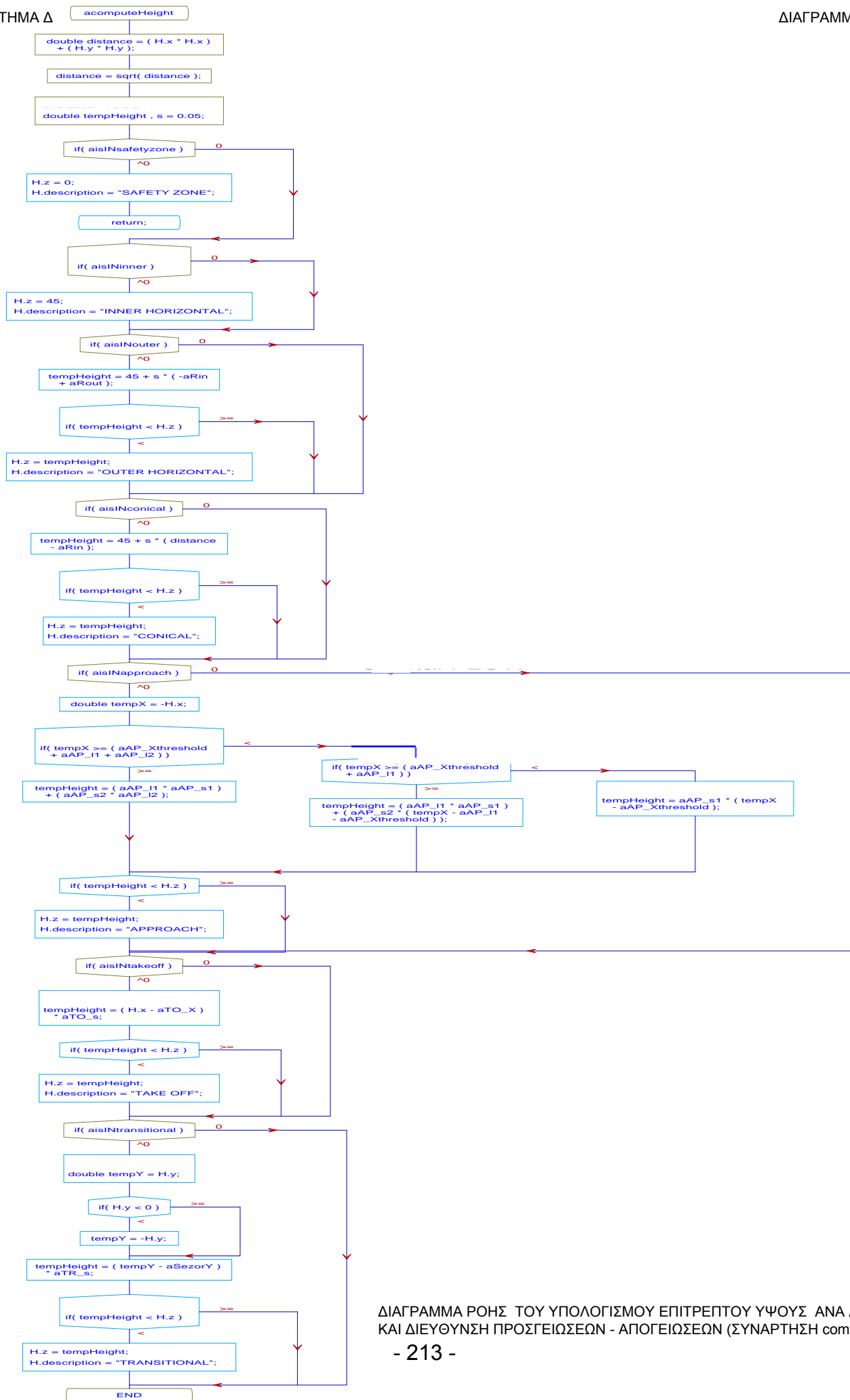


ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΤΗΣ ΕΡΕΥΝΑΣ ΠΕΡΙΛΗΨΗΣ ΣΕ ΠΕΡΙΓΡΑΦΜΑ ΠΟΛΥΓΩΝΟΥ (ΣΥΝΑΡΤΗΣΗ outline)

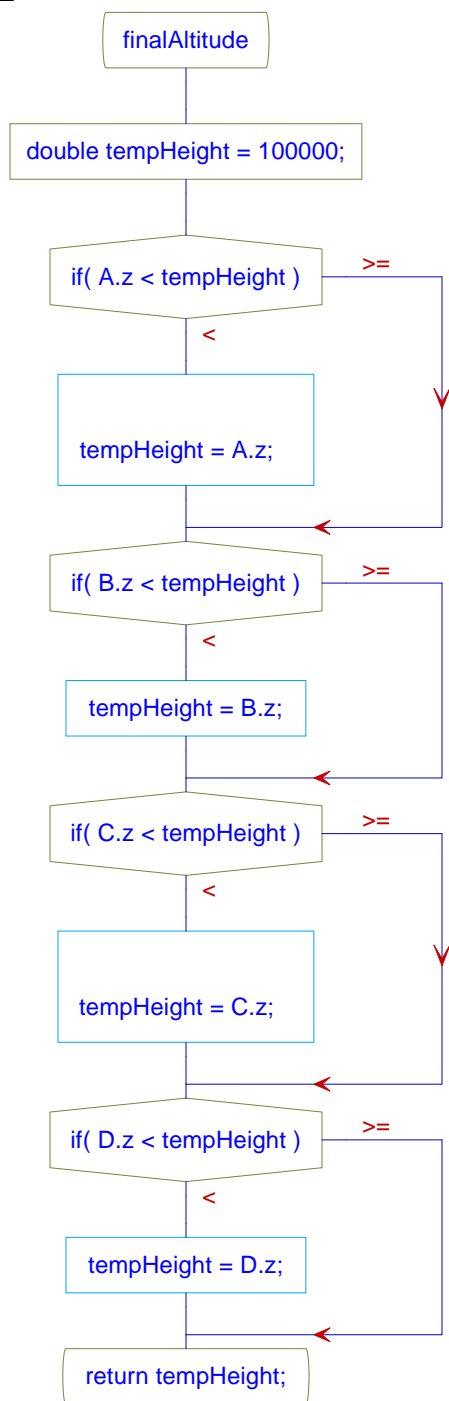


ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΤΗΣ ΕΡΕΥΝΑΣ ΠΕΡΙΛΗΨΗΣ ΣΤΗ ΜΕΤΑΒΑΤΙΚΗ ΕΠΙΦΑΝΕΙΑ (ΣΥΝΑΡΤΗΣΗ transitional)

ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΤΗΣ ΕΡΕΥΝΑΣ ΠΕΡΙΛΗΨΗΣ ΣΕ ΚΥΚΛΙΚΕΣ ΕΠΙΦΑΝΕΙΕΣ (ΣΥΝΑΡΤΗΣΗ `cyclic`)



ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΤΟΥ ΥΠΟΛΟΓΙΣΜΟΥ ΕΠΙΤΡΕΠΤΟΥ ΥΨΟΥΣ ΑΝΑ ΔΙΑΔΡΟΜΟ ΚΑΙ ΔΙΕΥΘΥΝΣΗ ΠΡΟΣΓΕΙΩΣΕΩΝ - ΑΠΟΓΕΙΩΣΕΩΝ (ΣΥΝΑΡΤΗΣΗ `computeHeight`)

ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΤΟΥ ΥΠΟΛΟΓΙΣΜΟΥ ΣΥΝΟΛΙΚΟΥ ΕΠΙΤΡΕΠΤΟΥ ΥΨΟΥΣ (ΣΥΝΑΡΤΗΣΗ `finalAltitude`)